Available online at www.sciencedirect.com
SCIENCE DIRECT®
European Journal of Operational Research 160 (2005) 501–514
EUROPEAN JOURNAL OF OPERATIONAL RESEARCH
www.elsevier.com/locate/dsw

Computing, Artificial Intelligence and Information Technology

Neural network forecasting for seasonal and trend time series

G. Peter Zhang [a,*], Min Qi [b]

[a] *Department of Management, J. Mack Robinson College of Business, Georgia State University, 35 Broad Street, NW, Atlanta, GA 30303, USA*
[b] *Department of Economics, College of Business Administration, Kent State University, Kent, OH 44242, USA*

Received 19 October 2001; accepted 8 August 2003
Available online 18 November 2003

Abstract

Neural networks have been widely used as a promising method for time series forecasting. However, limited empirical studies on seasonal time series forecasting with neural networks yield mixed results. While some find that neural networks are able to model seasonality directly and prior deseasonalization is not necessary, others conclude just the opposite. In this paper, we investigate the issue of how to effectively model time series with both seasonal and trend patterns. In particular, we study the effectiveness of data preprocessing, including deseasonalization and detrending, on neural network modeling and forecasting performance. Both simulation and real data are examined and results are compared to those obtained from the Box–Jenkins seasonal autoregressive integrated moving average models. We find that neural networks are not able to capture seasonal or trend variations effectively with the unpreprocessed raw data and either detrending or deseasonalization can dramatically reduce forecasting errors. Moreover, a combined detrending and deseasonalization is found to be the most effective data preprocessing approach.

Keywords: Neural networks; Box–Jenkins method; Seasonality; Time series; Forecasting

1. Introduction

Many business and economic time series exhibit seasonal and trend variations. Seasonality is a periodic and recurrent pattern caused by factors such as weather, holidays, repeating promotions, as well as the behavior of economic agents (Hylleberg, 1992). Although seasonal variations are perhaps the most significant component in a seasonal time series, a stochastic trend is often accompanied with the seasonal variations and can have a significant impact on various forecasting methods. A time series with trend is considered to be nonstationary and often needs to be made stationary before most modeling and forecasting processes take place. Accurate forecasting of seasonal and trend time series is very important for effective decisions in retail, marketing, production, inventory control, personnel, and many other business sectors (Makridakis and Wheelwright, 1987). Thus, how to model and forecast seasonal and trend time series has long been a major research topic that has significant practical implications.

* Corresponding author. Tel.: +1-404-651-4065; fax: +1-404-651-3498.
E-mail address: gpzhang@gsu.edu (G.P. Zhang).

0377-2217/$ - see front matter
doi:10.1016/j.ejor.2003.08.037

Traditional approaches to modeling seasonal time series are to remove the seasonal variations using certain seasonal adjustment method and then the models are scaled back using the estimated seasonal effects for forecasting purposes. The classic decomposition method, for example, decomposes seasonal time series into trend, seasonal, cyclical and irregular components. The seasonal influence is estimated and removed from the data first before other components are estimated. Seasonal autoregressive integrated moving average (ARIMA) models also require that the data be seasonally differenced to achieve stationarity condition (Box and Jenkins, 1976). This practice of seasonal adjustment is due to the belief that seasonal fluctuations can dominate the remaining variations in a time series, causing difficulty in effectively measuring other time series components. This is also the reason that macroeconomic statistics such as GDP and unemployment are often published as seasonally adjusted series for further analysis. Results from the well-known M-competition (Makridakis et al., 1982) suggest that data deseasonalization is an effective approach to seasonal time series modeling (Gardner and McKenzie, 1989).

However, seasonal adjustment of data before further analysis is not without controversy (Bell and Hillmer, 1984; Franses, 1996; Hylleberg, 1992). Ghysels et al. (1996) suggested that seasonal adjustment might lead to undesirable nonlinear properties in univariate time series. Ittig (1997) also questioned the traditional method for generating seasonal indexes and proposed a nonlinear method to estimate the seasonal factors. More importantly, some empirical studies find that seasonal fluctuations are not always constant over time and at least in some time series, seasonal components and nonseasonal components are not independent, and thus not separable (Hylleberg, 1994). The difficulty in distinguishing seasonal from nonseasonal fluctuations leads into the recent development of seasonal unit root models and periodic models that take explicit consideration of seasonal variations (Franses, 1996).

On the other hand, it is often not clear how to best model the trend pattern in a time series. In the popular Box–Jenkins approach to time series modeling, differencing is used to achieve stationarity in the mean. However, Pierce (1977) and Nelson and Plosser (1982) argued that differencing is not always an appropriate way to handle trend, and linear detrending may be more appropriate. Depending on the nature of the nonstationarity, a time series may be modeled in different ways. For example, a linear or polynomial time trend model can be used if the time series has a deterministic trend (so-called trend stationary series). On the other hand, if a time series exhibits a stochastic trend (so-called difference stationary series, or unit root process), the random walk model and its variations may be more appropriate. In practice, however, it is often difficult to determine whether a given series is trend stationary or difference stationary. Although statistical tests such as the Dickey–Fuller test and the Phillips–Perron test are available, they suffer from very low power in distinguishing between a unit root and a near unit root process. In addition, the testing procedure can be confounded by the presence of the deterministic regressors (i.e., the intercept and deterministic trend) which may further reduce the power of the test.

All aforementioned seasonal and trend time series models are parametric in nature. The limitation of these models is that the model form has to be prespecified without knowing the true underlying data generating process. In addition, an essentially linear relationship assumed in these models limits their capability to model complex nonlinear problems commonly encountered in reality. This may also be the cause of the mixed findings reported in the literature regarding the merits of seasonal adjustment.

The recent up-surging research activities in artificial neural networks (ANNs) as well as their numerous successful forecasting applications suggest that they can also be an important candidate for seasonal and trend time series forecasting. Being a flexible modeling tool, neural networks can, in principle, model any type of relationship in the data with high accuracy. With neural networks, no specific assumptions need to be made about the model and the underlying relationship is determined solely through data mining. This data driven approach is one of the most important advantages of neural networks in solving many complex real world forecasting problems. In addition, although ANNs are inherently nonlinear models, they are capable of modeling linear processes as well (Zhang, 1998).

Because neural networks are universal function approximators, it is natural to use them to directly model seasonal and trend variations. Gorr (1994) pointed out that neural networks should be able to simultaneously detect both the nonlinear trend and the seasonality in the data. Sharda and Patil (1992) examined 88 seasonal time series from the M-competition and found that neural networks can model seasonality effectively and pre-deseasonalizing the data is not necessary. Franses and Draisma (1997) found that neural networks could also detect possible changing seasonal patterns. Other studies that reported encouraging results for direct seasonal and trend forecasting include Kang (1991), Tang and Fishwick (1993), Nam and Schaefer (1995), and Williams (1997).

Farway and Chatfield (1995), however, showed mixed results with different neural network structures and forecasting horizons. Although only one time series—the well-known airline data from Box and Jenkins (1976)—is used in their study, the instability of neural network models in directly forecasting seasonal time series is obvious. Kolarik and Rudorfer (1994) reported the similar problem. Based on a study of 68 time series from the M-competition, Nelson et al. (1999) found that neural networks trained on deseasonalized data forecast significantly better than those trained on seasonally nonadjusted data. Hansen and Nelson (2003) find that the combination of transformation, feature extraction, and neural networks through stacked generalization gives more accurate forecasts than classical decomposition or ARIMA models.

The purpose of this paper is to examine the issue of how to use neural networks more effectively in modeling and forecasting a seasonal time series with a trend component. Specifically the research question we try to address is whether neural networks are able to directly model different components of a seasonal and trend time series or whether data preprocessing is necessary or beneficial. Instead of focusing solely on the seasonal component alone as in previous studies (e.g., Nelson et al., 1999), we take a systematic approach on the data preprocessing issue to study the relevance of detrending and deseasonalization. Using both simulated and real time series, we evaluate the effect of different data preprocessing strategies on neural network forecasting performance. The performance of neural network is judged against that of the well-established seasonal ARIMA model.

The rest of the paper is outlined as follows. In the next section, we review the issues of seasonal adjustment as well as Box–Jenkins and neural network approaches to seasonal time series modeling. The third section gives the research design and the data description. The fourth section presents the empirical findings. Finally we present the discussions of the findings and the conclusion in the fifth section.

2. Modeling seasonal time series with trend

Modeling seasonal and trend time series has been one of the main research endeavors for decades. In the early 1920s, the decomposition model along with seasonal adjustment was the major research focus due to Persons (1919, 1923) work on decomposing a seasonal time series. Different seasonal adjustment methods have been proposed and the most significant and popular one is the X-11 method developed by the Bureau of the Census in 1950s and 1960s (Shiskin et al., 1967) which has evolved into the current X-12-ARIMA program (Findley et al., 1996). Because of the ad hoc nature of the seasonal adjustment methods, several model-based procedures have been developed. Among them, the work by Box and Jenkins (1976) on the seasonal ARIMA model has had a major impact on the practical applications to seasonal time series modeling. This model has performed well in many real world applications and is still one of the most widely used seasonal forecasting methods. More recently, neural networks have been widely used as a powerful alternative to traditional time series modeling (Zhang et al., 1998). While their ability to model complex functional patterns in the data has been tested, their capability for modeling seasonal time series is not systematically investigated. In this section, we first review classical decomposition method with an emphasis on seasonal adjustment because it is the most critical part of the decomposition model. The Box–Jenkins and neural network approaches to seasonal time series modeling are then discussed.

2.1. Decomposition model with seasonal adjustment

Earlier methods for seasonal time series forecasting rely heavily on seasonal adjustment of the data. Several reasons for doing seasonal adjustment are summarized by Bell and Hillmer (1984). First, it can help estimate the trend and make short-term forecasting more efficient. Second, it can reveal the relationships between the time series under study and other economic series, external events, or policy variables, which may be masked by seasonal variations. Third, it can aid direct comparison among series values from month to month.

The key assumption inherent in many different seasonal adjustment methods is that seasonality can be separated from other components of the time series. That is, the seasonal time series y_t can be decomposed into a seasonal component S_t and a nonseasonal component NS_t:

$$y_t = S_t \cdot \mathrm{NS}_t. \tag{1}$$

The nonseasonal component can be further decomposed into trend, cycle, and irregular components if necessary. Model (1) is the classical multiplicative form of the decomposition model which is appropriate for many time series with increasing seasonal variations. If the seasonal variation is relatively constant with the trend, then we can use the additive decomposition model that decomposes the series into a sum of seasonal and other nonseasonal components.

Seasonal adjustment is a process of estimating the seasonal component called seasonal factors and then dividing for multiplicative models or subtracting for additive models the original series by seasonal factors. The ratio to moving average is the most widely used method for seasonal component estimation. This method uses a sequence of linear centered moving averages to smooth the seasonal variation and then the seasonal effect can be estimated by ratios of original series to the moving averages.

The Census X-11 (or the most recent version X-12) method is an important seasonal adjustment procedure used by many statistical and governmental institutions worldwide. Although X-11 is essentially a decomposition method and the basic smoothing method used in X-11 is the same as the simple ratio to moving average method, X-11 is much more complex and versatile. It can handle a variety of situations commonly occurred in business and economic time series including the trading day effect, holiday effect, and extreme observations. Other important features of X-11 include the options of a variety of moving averages for estimating evolving trend and seasonal components, using refined asymmetric moving averages toward the ends of a time series, and many diagnostic statistics to assess the appropriateness of the seasonal adjustment.

The X-11 method involves a number of tasks in sequence. First, the trading day and certain holiday effects are adjusted. Then, the initial or preliminary estimates of the seasonal factor and then trend-cycle component are made. Extreme values are also removed or adjusted. This is followed by a refinement of the preliminary seasonal factor and the trend-cycle component. Finally, diagnostic statistics are generated to aid evaluation of the model adequacy. The technical detail of the X-11 method can be found in, for example, Shiskin et al. (1967) and Hylleberg (1986).

2.2. Seasonal ARIMA model

The seasonal ARIMA model belongs to a family of flexible linear time series models that can be used to model many different types of seasonal as well as nonseasonal time series. In the most popular multiplicative form, the seasonal ARIMA model can be expressed as

$$\begin{aligned}\phi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D y_t \\ = \theta_q(B)\Theta_Q(B^s)\varepsilon_t,\end{aligned} \tag{2}$$

with

$$\phi_p(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p,$$

$$\Phi_P(B) = 1 - \Phi_s B^s - \Phi_{2s} B^{2s} - \cdots - \Phi_{Ps} B^{Ps},$$

$$\theta_q(B) = 1 - \theta_1 B - \theta_2 B^2 - \cdots - \theta_q B^q,$$

$$\Theta_Q(B) = 1 - \Theta_s B^s - \Theta_{2s} B^{2s} - \cdots - \Theta_{Qs} B^{Qs},$$

where s is the seasonal length ($s = 4$ for quarterly data and $s = 12$ for monthly data), B is the back shift operator defined by $B^k y_t = y_{t-k}$ and ε_t is a sequence of white noises with zero mean and constant variance. $(1 - B)^d$ and $(1 - B^s)^D$ are the nonseasonal and seasonal differencing operators, respectively. Model (2) is often referred to as the $\text{ARIMA}(p, d, q)(P, D, Q)_s$ model.

Seasonal ARIMA model building requires the specification of differencing orders (d, D) and the orders of both nonseasonal and seasonal autoregressive (AR) and MA operators (p, q, P, Q) as well as the estimation of model parameters in the AR and MA operator polynomials. This can be done with separate processes for nonseasonal and seasonal elements. It is important that the data be properly preprocessed before the estimation process takes place, as the ARIMA model requires that the time series be stationary. As most seasonal time series exhibit increasing trend and/or seasonal variations, both seasonal and nonseasonal differencing are often used to stabilize the time series. If the time series has unstable seasonal variations, the predifferencing Box–Cox transformation can be performed before further analysis.

Box and Jenkins (1976) proposed a set of effective model building strategies for seasonal ARIMA based on the unique autocorrelation structures in a time series. In addition to the regular autocorrelations among the time series observations, high correlations can also be found among observations from the same season, which is the main property of the seasonal time series used in the model identification process. The Box–Jenkins approach to the model building is through an iterative process of model identification, parameter estimation, and diagnostic checking. In the model identification stage, the sample autocorrelation properties are examined to help specify one or more tentatively appropriate models. This is equivalent to specifying the model orders for both nonseasonal part (p, d, q) and seasonal part (P, D, Q). Then the model parameters for the tentative models are estimated via maximum likelihood or least squares method. Finally, all estimated tentative models are undergone statistical tests for how well the fitted model conforms to the data. Diagnostic checking information can also be used to suggest how the model should be modified if the model does not fit well. The most parsimonious model among all adequate ones is usually chosen as the final model for forecasting.

2.3. Neural network model

Neural networks are a class of flexible nonlinear models that can discover patterns adaptively from the data. Theoretically, it has been shown that given an appropriate number of nonlinear processing units, neural networks can learn from experience and estimate any complex functional relationship with high accuracy. Empirically, numerous successful applications have established their role for pattern recognition and forecasting.

Although many types of neural network models have been proposed, the most popular one for time series forecasting is the feedforward network model. Fig. 1 shows a typical three-layer feedforward model used for forecasting purposes. The input nodes are the previous lagged observations while the output provides the forecast for the future value. Hidden nodes with appropriate nonlinear transfer functions are used to process the

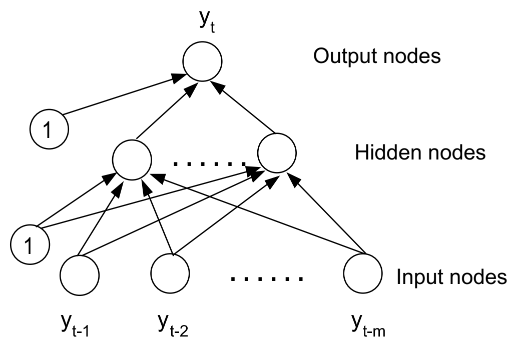

Fig. 1. A three-layer feedforward neural network.

information received by the input nodes. The model can be written as

$$y_t = \alpha_0 + \sum_{j=1}^{n} \alpha_j f\left(\sum_{i=1}^{m} \beta_{ij} y_{t-i} + \beta_{0j} \right) + \varepsilon_t, \qquad (3)$$

where m is the number of input nodes, n is the number of hidden nodes, f is a sigmoid transfer function such as the logistic: $f(x) = \frac{1}{1+\exp(-x)}$ · $\{\alpha_j, j = 0, 1, \ldots, n\}$ is a vector of weights from the hidden to output nodes and $\{\beta_{ij}, i = 0, 1, \ldots, m; j = 1, 2, \ldots, n\}$ are weights from the input to hidden nodes. α_0 and β_{0j} are weights of arcs leading from the bias terms which have values always equal to 1. Note that Eq. (3) indicates a linear transfer function is employed in the output node as desired for forecasting problems.

Functionally, the neural network expressed in (3) is equivalent to a nonlinear AR model. This simple structure of the network model has been shown to be capable of approximating arbitrary function (Cybenko, 1989; Hornik et al., 1989, 1990). However, few practical guidelines exist for building a neural network model for a time series. In particular, the specification of the neural model architecture in terms of the number of input and hidden nodes is not an easy task. Experiments with different architectures are often performed to identify an appropriate model. The parameters of the model are the arc weights that are typically estimated via the least squares method through a nonlinear optimization routine. Because of the potential overfitting problem, the model adequacy is often checked with a holdout sample that is not used in the model building process. In fact, to build an appropriate neural network model, the available data is often divided into three portions. The first training part is used for model training, i.e. parameter estimation, while the second validation part is for model selection. The last test sample is then used for true forecasting evaluation.

In a seasonal time series with seasonal autocorrelations, observations separated by multiples of seasonal period s are often correlated. For example, with monthly time series, observations that 12 months away are likely correlated. That is, a given observation y_t is related to the past and future observations $(y_{t-12}, y_{t+12}, y_{t-24}, y_{t+24}, \ldots)$. Therefore it is critical to include these seasonal observations in neural network model as possible input variables if directly modeling seasonality is desirable. It is important to note that although theoretically, many seasonal lagged observations may be included in the model, in practice, the number of seasonal observations need to be considered in model (3) can be fairly small, as empirical studies often suggest that the seasonal AR order is 1 or at most 2 in seasonal ARIMA models (Box and Jenkins, 1976; Pankratz, 1983). In modeling a monthly airline passenger series, Farway and Chatfield (1995) considered several neural network specifications with only one seasonal lag of 12 observations, y_{t-12}.

3. Research methodology

This research aims to provide some empirical evidence on the effectiveness of neural networks on modeling and forecasting seasonal and trend time series. The major research questions we investigate are:

- Are neural networks able to directly model seasonal and trend variations and produce satisfactory forecasts?
- Is data preprocessing helpful for neural network model to generate better forecasts? If so, what data preprocessing strategy is most useful?
- What is the comparative advantage of neural networks to the traditional Box–Jenkins ARIMA model in seasonal time series forecasting?

To address these and other related questions, we conduct a simulation study with data generated from a known seasonal and trend process. In addition, 10 real data sets are used to examine the issues and capabilities of neural network model with regard to the above research questions.

3.1. Data

Both simulated and real monthly data are considered in this study. The reason to use monthly time series is that they are the most commonly encountered data in business and economics as evidenced by the X-11 seasonal adjustment program that is almost exclusively applied to monthly data. Monthly time series are also more difficult to predict because they have more seasons to deal with than other types of seasonal data such as quarterly time series. Nonetheless, the same methodology can be applied to other seasonal time series.

Simulated series are generated according to the following simple multiplicative model:

$$y_t = T_t \mathrm{SI}_t + E_t, \tag{4}$$

where $T_t = 100 + 0.6t$ is the linear trend, SI_t is the seasonal index which is given in Table 1, and E_t is the error term following a normal distribution, $N(0, \sigma^2)$. To see the effect of the noise level on the forecasting accuracy of various models, we use the three levels of the error term variance, $\sigma^2 = 1$, 25, 100.

A total of 228 points are generated according to model (4) for each of the three noise levels. The first 216 observations are used for model selection and parameter estimation and the last 12 points are reserved as the test set for forecasting evaluation and comparison.

Our real time series include five US retail sales (from US Census Bureau), four industrial production series (from Federal Reserve Board), as well as US total new privately owned housing units started (from the Census Bureau). All data series end in December 2001, and are not seasonally adjusted. The five retail series are retail department stores, bookstores, clothing stores, furniture stores, hardware stores, all starting from January 1992. The four industrial production series are consumer goods (starting January 1970), durable goods (starting January 1947), fuels (starting January 1954), and total industrial production (starting January 1947). These aggregate time series are chosen because they demonstrate the trend and seasonal patterns that are investigated in the present study. Note that the sample sizes of these data sets vary from the smallest of 120 for retail sales series to the largest of 660 for the durable goods and total industrial production. Fig. 2 plots three representative time series of retail department, consumer goods, and durable goods. It is clear that all three series have an upward trend together with seasonal variations although the patterns are quite different. Department store sales seem to have the most regular though increasing seasonal and trend variations. In the consumer goods and durable goods series, we can see several discontinuities in the overall trend with durable goods demonstrating more dramatic changes in both trend and seasonal components. In each case, we reserve the last 12 months' data as the holdout sample for forecasting evaluation and the rest of the data (in-sample) are used for model selection and estimation.

3.2. Research design

Three types of data preprocessing are applied to the original time series: (1) trend adjustment (detrending) only, (2) seasonal adjustment (deseasonalization) only, and (3) both detrending and deseasonalization. Detrending is performed by fitting a linear time trend to the data and then subtracting the estimated trend from the series. Deseasonalization is done through the most recent X-12-ARIMA seasonal adjustment procedure. Therefore, for each data set, we have four time series representing the original (O), detrended (DT), deseasonalized (DS), and both detrended and deseasonalized (DSDT) upon which neural networks are built.

On the other hand, data preprocessing is a standard requirement and is built in the Box–Jenkins methodology for ARIMA modeling and hence seasonal ARIMA models are built using only the original data. Through the iterative model building process of identification, estimation, and

Table 1
Seasonal indexes used for simulated monthly data

Month	January	February	March	April	May	June	July	August	September	October	November	December
SI	0.75	0.80	0.82	0.90	0.94	0.92	0.91	0.99	0.95	1.02	1.20	1.80

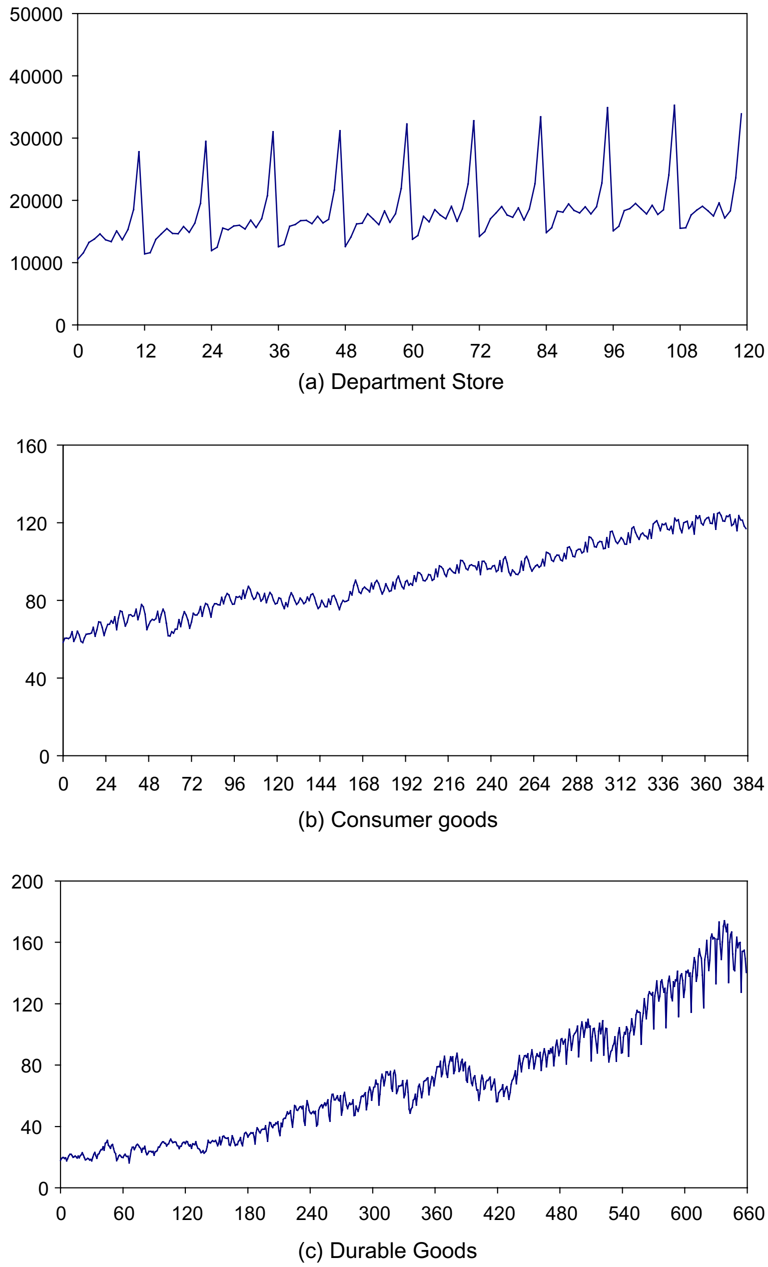

Fig. 2. Three representative seasonal time series: (a) department store, (b) consumer goods, (c) durable goods.

diagnostic checking, the final selected ARIMA model based on the in-sample data is believed to be the best for the testing sample (Fildes and Makridakis, 1995). In this study, we use Forecast Pro to conduct automatic ARIMA model building and evaluation. The capability of Forecast Pro is documented in Goodrich (2000).

To determine the best neural network structure for each time series, an experiment is conducted with the basic cross validation method. The available in-sample data are divided into a training part and a validation part. The validation part is the last 12-month observations while the training sample consists of all previous observations. The training sample is used to estimate the parameters for any specific model architecture. The validation set is then used to select the best model among all models considered. Because of the nature of the autocorrelation in a time series, the number of input nodes or the lagged observations used in the neural networks is often a more important factor than the number of hidden nodes (Zhang, 1998). In this study, the number of hidden nodes varies from 2 to 14 with an increment of 2. For the original and detrended data where the seasonal variations are present, we consider 10 various lag numbers: 1–4, 12–14, 24, 25 and 36. The lags of 12, 24, and 36 are included due to the fact that for monthly seasonal data, observations 12, 24, and 36 months apart are usually highly correlated. To model any seasonal time series, it is necessary to incorporate such seasonal autocorrelation structure. For the deseasonalized data, the number of input nodes simply varies from 1 to 4. All together, we consider 28 architectures in the experimentation for the seasonally adjusted (or deseasonalized) series and 70 models for the seasonal series.

The neural network model used in this study is the standard three-layer feedforward network shown in Fig. 1. Since the one-step-ahead forecasting is considered, only one output node is employed. The transfer function for hidden nodes is the logistic function and for the output node, the identity function. Bias terms are used in both hidden and output layers. The fast Levenberg and Marquardt algorithm provided by the MATLAB neural network toolbox is employed in training. To avoid getting stuck in local minima, we adopt the common practice of multiple starts in neural network training. In particular, based on the training subsample we train each neural network five times using five different sets of initial random weights. This results in five neural network models, among which we choose the best-fit one to report its training, validation and testing results. To increase the chance of getting the global minimum, adaptive learning parameter is also used in Levenberg and Marquardt algorithm. The initial value is set to 0.01. The learning rate is increased by a factor of 10 until the change above results in a reduced performance value. The change is then made to the network and the learning rate is decreased by a factor of 0.1. The maximum training epochs are 1000.

For preprocessed data including detrending and deseasonalization, we convert the outputs from the neural networks back to their original scales. Therefore, the model fitting and forecasting performance can be directly compared. The performance of in-sample fit and out-of-sample forecast is judged by three commonly used error measures. They are the root mean squared error (RMSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE). The use of these measures represents different angles to evaluate forecasting models. The first two are absolute performance measures while the last one is a relative measure.

4. Empirical findings

Table 2 summarizes the neural network modeling and forecasting results for the simulation data. Recall that for neural network modeling, all of the available data are partitioned into three parts, a training sample, a validation sample, and a testing set. The model estimation and selection is performed using the first two samples with the final model selected based on the validation sample result. Table 2 reports performance measures across training, validation, and testing samples along with three noise levels. Several observations can be made from Table 2. First, neural networks are not able to model and forecast well directly with the original seasonal time series. For example, at the lowest noise level (Noise level 1), the RMSE, MAE, and MAPE of the training sample are 0.92, 0.72, and 0.49, respectively, for the deseasonalized series. Yet, for the unpreprocessed original data, they are 45.31, 28.36, and 18.18, respectively. It is obvious that compared with detrended and/or deseasonalized series, neural networks trained with the original data perform definitely the worst not only in the in-sample training and validation sets, but also in the out-of-sample testing sample, at all three noise levels and by all three error measures.

Table 2
Simulation result for neural networks

Noise level	Data type[a]	Training			Validation			Testing		
		RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
1	O	45.31	28.36	18.18	65.52	30.31	10.09	71.98	34.54	10.98
	DT	1.17	0.91	0.59	1.34	1.16	0.53	1.70	1.48	0.65
	DS	0.92	0.72	0.49	1.05	0.81	0.38	1.67	1.34	0.62
	DSDT	0.78	0.59	0.39	0.78	0.73	0.35	1.33	1.08	0.49
2	O	42.49	24.3	15.25	57.89	29.13	9.84	62.03	28.94	9.02
	DT	14.08	10.81	6.73	18.54	14.37	5.97	20.61	15.29	5.90
	DS	7.09	5.65	3.73	8.56	6.26	2.89	6.59	5.48	2.36
	DSDT	4.27	3.26	2.15	5.14	4.56	2.16	3.61	2.64	1.09
3	O	47.91	32.53	20.59	59.24	27.81	8.91	69.94	38.15	12.57
	DT	33.36	23.62	13.56	37.22	19.83	7.41	44.07	27.32	10.75
	DS	12.32	8.98	5.80	20.34	18.35	8.02	29.61	26.37	10.72
	DSDT	8.72	6.53	4.23	8.55	6.77	3.35	8.56	6.72	3.07

[a] O = original, DT = detrended, DS = deseasonalized, DSDT = deseasonalized and detrended.

Second, data preprocessing has significant impact on neural network learning and generalization ability. As can be seen from Table 2, at all three noise levels and across the training, validation, and testing samples, the detrended and the deseasonalized series have dramatically smaller RMSE, MAE, and MAPE compared to the original series. Therefore, both detrending and deseasonalization are critical for neural networks to learn the underlying pattern and forecast seasonal and trend time series. Furthermore, at all three noise levels and across the training, validation, and testing samples, the DSDT series have the smallest errors based on all three measures. This indicates that applying detrending and deseasonalization simultaneously is the most effective data processing approach in modeling and forecasting seasonal time series, although either data detrending or deseasonalization alone can dramatically reduce model fitting and forecast errors.

Finally, as the noise in the time series increases, neural network performance generally gets worse. With a few exceptions with the original data, neural networks perform worse judged by three measures of RMSE, MAE, and MAPE when the noise level becomes larger. For example, on the training side with deseasonalized and detrended data, the RMSE, MAE, and MAPE at the lowest noise level 1 are 0.78, 0.59 and 0.39, respectively. These measures increase to 4.27, 3.26 and 2.15 at noise level 2 and 8.72, 6.53, and 4.23 at the highest level 3. The same pattern can be recognized in the validation and testing results.

The out-of-sample forecasting comparison between neural networks and the benchmark seasonal ARIMA models is presented in Table 3. It is

Table 3
Out-of-sample comparison between neural networks (DSDT) and seasonal ARIMA models

	Noise 1			Noise 2			Noise 3		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
ANN	1.33	1.08	0.49	3.61	2.64	1.09	8.56	6.72	3.07
ARIMA	1.91	1.54	0.66	4.56	3.85	1.76	11.58	9.83	4.22
Difference[a]	−0.58	−0.46	−0.17	−0.95	−1.21	−0.67	−3.02	−3.11	−1.15

[a] Difference = ANN − ARIMA.

clear that neural networks forecast better than the ARIMA models in all situations. Although both neural networks and ARIMA models perform worse as data become noisier, neural networks seem to be more robust as the differences between the two models are getting larger.

Table 4 gives the overall result for the 10 real seasonal time series. Across all 10 series, we find that neural networks are not capable of modeling original seasonal data well. The in-sample and out-of-sample performance with the unpreprocessed data is much worse than that with the preprocessed series. In addition, neural network modeling and forecasting ability is quite sensitive to the type of data preprocessing performed for the seasonal time series. Although both detrending and deseasonalization help in improving neural network modeling and forecasting performance, deseasonalization seems to be a more important factor in out-of-sample forecasting as it yields lower testing error measures than detrending in all 10 time series except clothing stores and hardware stores sales. The most effective data preprocessing form from both model fitting and forecasting perspectives is again the combined detrending and deseasonalization, reinforcing the findings obtained with the simulation data.

Compared to the seasonal ARIMA models, neural networks trained with DSDT data consistently forecast better than ARIMA models in all time series across all performance measures. It is important to note that although neural networks with both detrending and deseasonalization preprocessing outperform ARIMA models, they do not always beat ARIMA if the data are not appropriately preprocessed. In fact, for all but two series (consumer goods and total industrial production), neural networks built with original data perform worse than ARIMA models. This further suggests the importance of appropriate data preprocessing for neural network modeling and forecasting.

To understand how neural networks model the seasonal time series, the final neural network models for the three representative series with regard to various data types are given in Table 5. In Table 5, the column labeled "lag" gives the maximal lagged term that is included in a model. For example, the "lag" of the original department store sales is 36, which means that the neural network model for this original series has the following 10 inputs:

$$y_{t-1},\ y_{t-2},\ y_{t-3},\ y_{t-4},\ y_{t-12},\ y_{t-13},\ y_{t-14},\ y_{t-24},\ y_{t-25},\ \text{and}\ \ y_{t-36}.$$

From Table 5, it is obvious that there is a great amount of variations in the structure of the best neural network model in terms of the lag number and the number of hidden nodes for different data types in modeling different series. On one hand, we find that for original and detrended data which still have the seasonal components, the best neural network model for all series contains at least 12 lags that correspond to a seasonal period of 12 for monthly series. For all the deseasonalized series (no matter whether detrended or not), much fewer lags have been selected. Our findings thus suggest that neural networks have the capability to correctly identify the seasonal autocorrelation structure for data containing seasonal variations. On the other hand, we notice relatively large discrepancies in the number of hidden nodes across the four data types for each retail series and across the three time series within the same data type. This is not surprising given the different statistical properties of the raw and preprocessed data, the different numbers of input variables, and the uncertainties in neural network training.

5. Conclusions

Seasonal and trend variations are the two most commonly encountered phenomena in many sectors of business and economics. How to best model and forecast these variations is an important task in planning and other related decision-making activities. In this paper, we examine the capability of artificial neural networks in modeling and forecasting seasonal and trend time series. The issue of how to effectively model seasonal and trend time series is investigated with both simulated and real data.

Our results clearly indicate that neural networks are not able to model seasonality directly. Prior data processing is necessary and critical to build an

Table 4
Results for real time series

Data	Data type[a]	Model	Training			Validation			Testing		
			RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Retail department	O	ANN	1314.33	1042.37	5.72	1138.36	916.06	4.54	2166.86	1570.66	8.73
	DT	ANN	1313.48	893.83	4.92	924.98	792.81	3.94	1785.77	1549.09	8.51
	DS	ANN	849.11	705.14	4.03	1519.41	1431.82	7.11	1127.02	1088.03	5.67
	DSDT	ANN	382.23	305.03	1.79	228.05	172.25	0.88	975.55	847.35	4.17
	O	ARIMA							1005.41	858.64	4.33
Consumer goods	O	ANN	1.05	0.84	0.96	1.00	0.76	0.62	1.57	1.34	1.12
	DT	ANN	0.93	0.72	0.81	1.01	0.81	0.67	1.48	1.16	0.96
	DS	ANN	0.88	0.67	0.78	0.75	0.69	0.57	0.80	0.64	0.53
	DSDT	ANN	0.76	0.57	0.66	0.81	0.70	0.57	0.68	0.55	0.46
	O	ARIMA							3.96	3.39	2.83
Durable	O	ANN	2.60	1.93	3.40	4.11	3.73	2.37	6.43	4.89	3.27
	DT	ANN	2.49	1.82	3.24	3.64	3.12	1.93	5.98	4.52	3.01
	DS	ANN	1.97	1.36	2.41	3.09	2.59	1.61	3.80	2.95	1.96
	DSDT	ANN	1.90	1.33	2.36	3.03	2.65	1.67	3.63	2.66	1.75
	O	ARIMA							5.61	4.47	2.98
Book store	O	ANN	159.93	124.68	11.56	215.65	208.68	16.40	328.62	305.61	21.44
	DT	ANN	76.45	55.56	5.27	60.12	48.10	3.60	170.49	105.96	6.42
	DS	ANN	50.28	32.82	3.33	60.96	52.53	4.28	132.87	85.06	5.77
	DSDT	ANN	37.39	23.14	2.19	39.69	28.98	2.14	88.74	43.23	2.70
	O	ARIMA							98.17	51.06	3.26
Clothing stores	O	ANN	712.54	578.97	6.98	1546.63	1440.64	14.46	1746.55	1662.36	16.57
	DT	ANN	499.70	365.06	4.50	722.17	602.92	6.00	1117.72	691.46	6.60
	DS	ANN	455.85	363.31	4.66	746.20	661.93	6.36	920.96	866.82	8.49
	DSDT	ANN	174.07	131.06	1.70	203.20	175.09	1.93	315.43	206.52	1.96
	O	ARIMA							519.60	405.53	3.98
Furniture store	O	ANN	269.82	211.52	6.56	434.42	404.42	9.48	479.62	391.92	8.97
	DT	ANN	132.11	101.62	3.08	107.84	91.44	2.18	226.68	193.12	4.66
	DS	ANN	67.08	52.60	1.68	108.39	68.76	1.63	135.89	109.46	2.66
	DSDT	ANN	52.35	38.22	1.20	140.93	112.56	2.67	99.45	76.23	1.86
	O	ARIMA							124.44	103.82	2.49
Hardware store	O	ANN	73.60	57.65	4.93	91.35	79.74	6.14	183.49	167.30	11.89
	DT	ANN	67.09	52.78	4.61	61.58	48.01	3.95	105.12	84.55	5.96
	DS	ANN	43.39	33.16	2.86	32.22	25.26	1.97	142.76	131.33	9.21
	DSDT	ANN	26.16	18.69	1.62	20.48	15.52	1.27	49.17	41.24	2.93
	O	ARIMA							100.71	89.40	6.25
Housing start	O	ANN	27.89	21.74	18.15	10.59	8.78	7.05	16.39	12.60	10.13
	DT	ANN	11.51	9.11	7.84	7.53	6.51	5.03	7.55	5.73	4.32
	DS	ANN	9.60	7.22	6.07	5.84	4.71	3.55	5.03	3.63	2.96
	DSDT	ANN	9.11	6.81	5.73	5.87	4.62	3.60	4.23	3.05	2.40
	O	ARIMA							7.88	6.63	5.18
Fuels	O	ANN	1.88	1.46	1.70	2.62	2.15	1.92	1.98	1.89	1.65
	DT	ANN	1.78	1.38	1.62	2.33	1.86	1.67	1.83	1.59	1.41
	DS	ANN	1.60	1.15	1.36	1.90	1.50	1.34	1.15	0.93	0.82
	DSDT	ANN	1.48	1.08	1.29	1.95	1.59	1.42	0.81	0.63	0.56
	O	ARIMA							1.62	1.34	1.16

Table 4 (*continued*)

Data	Data type[a]	Model	Training			Validation			Testing		
			RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
Iptotal	O	ANN	0.81	0.62	1.08	1.22	0.97	0.66	1.64	1.41	1.01
	DT	ANN	0.81	0.62	1.09	1.24	1.03	0.70	1.62	1.32	0.94
	DS	ANN	0.54	0.39	0.68	0.73	0.63	0.43	1.14	1.03	0.74
	DSDT	ANN	0.47	0.35	0.62	0.76	0.62	0.43	0.85	0.71	0.50
	O	ARIMA							8.94	8.05	5.79

[a] O = original, DT = detrended, DS = deseasonalized, DSDT = deseasonalized and detrended.

Table 5
Best neural network models for three retail series

Data type[a]	Department		Consumer		Durable	
	Lag	Hidden	Lag	Hidden	Lag	Hidden
O	36	12	25	10	14	10
DT	13	14	36	8	14	6
DS	1	4	3	10	3	10
DSDT	1	10	4	8	4	2

[a] O = original, DT = detrended, DS = deseasonalized, DSDT = deseasonalized and detrended.

adequate neural forecaster. In addition to seasonally adjusting the data, we find removing the trend component is also vital in significantly improving forecasting performance. Our finding on the effectiveness of seasonal adjustment confirms that of Nelson et al. (1999) who concluded that neural networks built with deseasonalized data could produce significantly more accurate forecasts than with non-deseasonalized data. Furthermore, we show that the most effective data preprocessing method for time series with both seasonality and trend is a joint approach of detrending and deseasonalization.

Our explanation to why the preprocessing helps is that seasonal and trend variations of the time series with seasonality and trend may account for the preponderance of its total variance. Models that ignore these seasonal or trend patterns will results in a high variance thus poor forecasting accuracy. The process of deseasonalizing and detrending removes these large seasonal and trend variations from the raw data, thus helps improving the modeling accuracy. At first glance, the inability of the feedforward neural network model to model a trend seems to be at odds with its universal approximation theory. The explanation is that a trend time series does not meet the conditions for universal approximation. The neural network model considered in Eq. (3) is bounded because the sigmoid transfer function is bounded. Therefore neural network forecasting of a process with a trend is doomed to fail.

Several practical implications can be drawn from the present study. First, just as traditional statistical models, neural networks are not able to simultaneously handle many different components of the data well and hence data preprocessing is beneficial. Second, with preprocessed data that do not contain long dynamic autocorrelation structures, parsimonious neural models can be constructed as evidenced by our empirical results. This is an important advantage from the practical perspective as parsimonious neural networks can ease not only the model building process but also the potential overfitting problem. Finally, neural networks with both detrending and deseasonalization are able to significantly outperform seasonal ARIMA models in out-of-sample forecasting. However, without appropriate data preprocessing, neural networks may yield much worse forecasting performance than ARIMA models.

References

Bell, W.R., Hillmer, S.C., 1984. Issues involved with the seasonal adjustment of economic time series. Journal of Business and Economic Statistics 2, 291–320.

Box, G.E.P., Jenkins, G.M., 1976. Time Series Analysis: Forecasting, and Control. Holden Day, San Francisco, CA.

Cybenko, G., 1989. Approximation by superpositions of a sigmoid function. Mathematics of Control Signals and Systems 2, 303–314.

Farway, J., Chatfield, C., 1995. Time series forecasting with neural networks: A comparative study using the airline data. Applied Statistics 47, 231–250.

Fildes, R., Makridakis, S., 1995. The impact of empirical accuracy studies on time series analysis and forecasting. International Statistical Review 63 (3), 289–308.

Findley, D.F., Monsell, B.C., Bell, W.R., Otto, M.C., Chen, B.C., 1996. New capabilities and methods of the X-12-ARIMA seasonal-adjustment program. Journal of Business and Economic Statistics 16 (2), 127–152.

Franses, P.H., 1996. Recent advances in modelling seasonality. Journal of Economic Survey 10 (3), 299–345.

Franses, P.H., Draisma, G., 1997. Recognizing changing seasonal patterns using artificial neural networks. Journal of Econometrics 81, 273–280.

Gardner, E., McKenzie, E., 1989. Seasonal exponential smoothing with damped trends. Management Science 35 (3), 372–376.

Ghysels, E., Granger, C.W.J., Siklos, P.L., 1996. Is seasonal adjustment a linear or nonlinear data filtering process? Journal of Business and Economics Statistics 14, 374–386.

Goodrich, R.L., 2000. The forecast pro methodology. International Journal of Forecasting 16, 533–535.

Gorr, W.L., 1994. Research prospective on neural network forecasting. International Journal of Forecasting 10, 1–4.

Hansen, J.V., Nelson, R.D., 2003. Forecasting and recombining time-series components by using neural networks. Journal of the Operational Research Society 54 (3), 307–317.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366.

Hornik, K., Stinchcombe, M., White, H., 1990. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural Networks 3, 551–560.

Hylleberg, S., 1986. Seasonality in Regression. Academic Press, Orlando, FA.

Hylleberg, S., 1992. General introduction. In: Hylleberg, S. (Ed.), Modelling Seasonality. Oxford University Press, Oxford, pp. 3–14.

Hylleberg, S., 1994. Modelling seasonal variation. In: Hargreaves, C.P. (Ed.), Nonstationary Time Series Analysis and Cointegration. Oxford University Press, Oxford, pp. 153–178.

Ittig, P.T., 1997. A seasonal index for business. Decision Sciences 28 (2), 335–355.

Kang, S., 1991. An investigation of the use of feedforward neural networks for forecasting. Ph.D. Dissertation, Kent State University, Kent, OH.

Kolarik, T., Rudorfer, G., 1994. Time series forecasting using neural networks. APL Quote Quad 25, 86–94.

Makridakis, S., Wheelwright, S.C., 1987. The Handbook of Forecasting: A Manager's Guide, second ed. Wiley, New York.

Makridakis, S., Anderson, A., Carbone, R., Fildes, R., Hibdon, M., Lewandowski, R., Newton, J., Parzen, E., Winkler, R., 1982. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. Journal of Forecasting 1, 111–153.

Nam, K., Schaefer, T., 1995. Forecasting international airline passenger traffic using neural networks. Logistics and Transportation Review 31 (3), 239–251.

Nelson, C.R., Plosser, C.I., 1982. Trends and random walks in macroeconomic time series: Some evidence and implications. Journal of Monetary Economics 10, 139–162.

Nelson, M., Hill, T., Remus, T., O'Connor, M., 1999. Time series forecasting using NNs: Should the data be deseasonalized first? Journal of Forecasting 18, 359–367.

Pankratz, A., 1983. Forecasting with Univariate Box–Jenkins Models: Concepts and Cases. John Wiley & Sons, New York.

Persons, W.M., 1919. Indices of business conditions. Review of Economics and Statistics 1, 5–107.

Persons, W.M., 1923. Correlation of time series. Journal of the American Statistical Association 18, 713–726.

Pierce, D.A., 1977. Relationships—and the lack of thereof—between economic time series, with special reference to money and interest rates. Journal of the American Statistical Association 72, 11–26.

Sharda, R., Patil, R.B., 1992. Connectionist approach to time series prediction: An empirical test. Journal of Intelligent Manufacturing 3, 317–323.

Shiskin, J., Young, A.H., Musgrave, J.C., 1967. The X-11 variant of the census method II seasonal adjustment program. Technical Paper No. 15, US Department of Commerce, Bureau of Economic Analysis.

Tang, Z., Fishwick, P.A., 1993. Feedforward neural nets as models for time series forecasting. ORSA Journal of Computing 5 (4), 374–385.

Williams, P.M., 1997. Modelling seasonality and trends in daily rainfall data. In: Touretzky, D.S., Mozer, M.C., Hasselmo, M.E. (Eds.), Advances in Neural Information Processing Systems, vol. 10, pp. 985–991.

Zhang, G., 1998. Linear and nonlinear time series forecasting with artificial neural networks. Ph.D. Dissertation, Kent State University, Kent, OH.

Zhang, G., Patuwo, B.E., Hu, M.Y., 1998. Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting 14, 35–62.