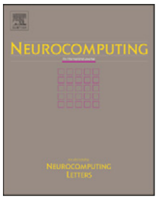
Neurocomputing 73 (2010) 1923–1936

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Feature selection for time series prediction – A combined filter and wrapper approach for neural networks

Sven F. Crone, Nikolaos Kourentzes *

Lancaster University Management School, Department of Management Science, Centre for Forecasting, Bailrigg campus, Lancaster LA1 4YX, UK

ARTICLE INFO

Available online 12 March 2010

Keywords:
Time series prediction
Forecasting
Artificial neural networks
Automatic model specification
Feature selection
Input variable selection

ABSTRACT

Modelling artificial neural networks for accurate time series prediction poses multiple challenges, in particular specifying the network architecture in accordance with the underlying structure of the time series. The data generating processes may exhibit a variety of stochastic or deterministic time series patterns of single or multiple seasonality, trends and cycles, overlaid with pulses, level shifts and structural breaks, all depending on the discrete time frequency in which it is observed. For heterogeneous datasets of time series, such as the 2008 ESTSP competition, a universal methodology is required for automatic network specification across varying data patterns and time frequencies. We propose a fully data driven forecasting methodology that combines filter and wrapper approaches for feature selection, including automatic feature evaluation, construction and transformation. The methodology identifies time series patterns, creates and transforms explanatory variables and specifies multilayer perceptrons for heterogeneous sets of time series without expert intervention. Examples of the valid and reliable performance in comparison to established benchmark methods are shown for a set of synthetic time series and for the ESTSP'08 competition dataset, where the proposed methodology obtained second place.

1. Introduction

Artificial neural networks (NN) have found increasing consideration in forecasting research and practice, leading to over 5000 academic publications indexed by ISI [1]. However, despite their proven theoretical capabilities of non-parametric, data driven universal approximation of any linear or nonlinear function [2], NN have not been able to confirm their potential against established statistical methods, such as ARIMA or Exponential Smoothing [3] in objective, empirical competitions on large sets of time series. The resulting gap between the theoretical capabilities, empirical accuracy and robustness in automatic applications of NNs has led to increased research activities to explore the empirical accuracy of NNs under different data conditions in a number of forecasting competitions (see e.g. the NN3, NN5, and the annual ESTSP competitions). Evidence from the algorithms employed in prior competitions has shown a myriad of unique approaches to specify NNs for time series prediction. A possible explanation is given by the many degrees of freedom offered by NN architectures, which must be chosen in the modelling process in interaction with the underlying data: from the selection of the information processing within each node (i.e. specifying input and activation functions), the selection of input, hidden and output nodes, structure and recurrencies of the connection-weights to combine nodes in adequate network topologies, to learning algorithms and parameters and choices made in preceding stages of data sampling and pre-processing. Consequently, the valid and reliable specification of NNs for a given time series is often considered as much an art as science, limiting the automation of NN modelling and implementation.

Previous research indicates that the automatic identification of the most relevant input variables to approximate an unknown data generating process, i.e. feature selection on time series data, poses one of the key challenges in automatic model specification of NNs [1,4]. The importance of input variable and lag selection is evident, as the input vector needs to capture all characteristics of complex time series, including the components of deterministic or stochastic trends, cycles and seasonality, interacting in a linear or nonlinear model with pulses, level shifts, structural breaks and different distributions of noise. Furthermore, the amount and complexity of time series patterns varies with the time series domain and sampling frequency of the data, from low frequency data recoded in quarterly or monthly intervals to high frequency time series of weekly, daily or intraday data. As empirical datasets often contain multiple time series with distinct properties and components, they require individual identification, specification and prediction. Despite recent interest in modelling NNs for time series with seasonal and trend components [5], these normally assume a given and known seasonal form for a set of synthetic time series. In contrast, of the 3003 time series of the M3-competition [3] each monthly time series contained different forms of monthly, quarterly or no seasonality, different forms of

* Corresponding author. Tel.: +44 1524593464.
E-mail address: n.kourentzes@lancaster.ac.uk (N. Kourentzes).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.01.017

trend and frequent outliers, level shifts, etc. that require individual identification and hence an automated methodology. A number of methodologies have been proposed for feature evaluation of NNs, including filter-based approaches employing statistical tests such as stepwise regression, autocorrelation or spectral analysis, in addition to wrappers employing a stepwise search of feasible model candidates using the increasingly available computational power. However, no single methodology has been proven to perform well consistently across varying data conditions [6], given their individual shortcomings. Linear statistical tests fail in identifying nonlinear interdependencies, and bias the results of nonlinear NNs, often provide ambiguous results for multiple seasonalities and are prone to overspecification on large datasets and high frequency data. Similarly, wrapper based approaches often prove inefficient, as they reach the limits of available computational power with the growing number of possible feature combinations. In the absence of valid and reliable evaluations, there currently exists no consensus on what methodology should be applied under which data conditions [7], in particular for unknown time series frequency and multiple overlying seasonality [8].

While some time series components have been successfully addressed by feature selection methodologies, e.g. identifying only the most relevant lagged realisations of the dependent variable, in feature evaluation others may require feature construction of explanatory dummy variables with adequate time-delays, depending on the stochastic or deterministic behaviour of each component. For multivariate modelling the specification of correct contemporaneous or lagged realisations of the dependent variable, and/or multiple explanatory variables provides an even bigger challenge [9]. These challenges determine the desirable properties of a necessary methodology of feature selection to specify the input vector of NNs: fully automatic (a) feature evaluation of unknown time series components of level, trend and seasonality of arbitrary length, magnitude or type, (b) feature construction to capture deterministic and/or stochastic time series patterns through explanatory variables, (c) feature transformation for adequate pre-processing of chosen input variables, and (d) network architecture selection. The resulting methodology should be able to approximate any unknown data generating process for each time series without the need of domain knowledge or expert intervention. To address this challenge, we propose a fully automatic methodology to specify multilayer perceptrons (MLP), founded on best practices of filters and wrappers from statistics and computational intelligence. The methodology is centred around an iterative neural filter, which combines a simple graphical tool of analysing the Euclidian distance in seasonal year-on-year-plots frequently employed by forecasting practitioners with an iterative specification of an MLP as a non-parametric filter for automatic feature evaluation and time series identification. In addition, we propose a series of subsequent wrappers for feature construction of explanatory dummy time series, feature transformation in the form of time series differencing, and to determine the MLP architecture.

The paper is organized as follows. First, we briefly introduce NNs in the context of time series forecasting to derive the particular importance of input vector specification and discuss challenges in conventional methodologies for feature selection on low and high frequency data. Sections 3 and 4 introduce the proposed methodology: Section 3 specifies the iterative neural filter for feature evaluation, which is embedded in a series of wrappers for feature construction and transformation specified in Section 4. Section 5 provides details on the submission to the ESTSP'08 competition in specifying the experimental design, models used and preliminary results obtained. Finally, we provide conclusions and future work in Section 6.

2. Modelling neural networks for forecasting

2.1. Time series prediction with multilayer perceptrons

Forecasting with NNs requires the specification of a hetero-associative NN architecture in order to approximate and extrapolate the underlying data generating process. The NN architecture determines the relationship $\hat{y}=f(X,Y)$ between a vector of past time series information of independent X and/or dependent Y variables and future predicted values of a dependent variable $\hat{y}$. Due to the many degrees of freedom in specifying NNs in time series forecasting, we present a brief introduction; a general discussion is given in [10,11].

In model specification, the variables (measured at discrete time intervals) included in the input vector determine the model form in accordance with statistical forecasting models. Including only n lagged realisations of the dependent variable y_{t-n} in the input vector, $\hat{y}_{t+1}=f(y_t, y_{t-1}, \ldots, y_{t-n+1})$, constructs a NN for time series forecasting. For models using only m explanatory variables x_m of metric or nominal scale, the NN is constructed for causal forecasting, estimating a functional relationship of the form $\hat{y}=f(x_1, x_2, \ldots, x_m)$. By combining contemporaneous and lagged realisations of the independent variables $x_{m,t-n}$ and lagged dependent variables y_{t-n} more general models of dynamic regression, autoregressive (AR) transfer functions and intervention models are constructed. To extend beyond the autoregressive models of feedforward architectures, recurrent architectures allow the inclusion of moving average components (MA) of past model errors in analogy to the ARIMA Methodology [12], enabling a large class of nonlinear dynamic regression models to be constructed using NNs [13]. Forecasting time series with NN conventionally employs a feedforward topology of the established multilayer perceptron (MLP) in analogy to an nonlinear autoregressive model of order p, NAR(p) [1,14], to which we will also limit our analysis. In time series prediction with MLPs, for a point in time t a h-step ahead forecast $\hat{y}_{t+h}$ is computed using $n=p$ lagged observations $y_t, y_{t-1}, \ldots, y_{t-n+1}$ from n preceding points in time $t, t-1, t-2, \ldots, t-n+1$, with $n=I$ denoting the number of input units of the MLP. The functional form of a single layered MLP with a single output node for is

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

with $Y=[y_t, y_{t-1}, \ldots, y_{t-I+1}]$ the vector of the lagged observations of the time series providing the network inputs. The network parameters are denoted as weights $w=(\beta,\gamma)$, $\beta=[\beta_1, \beta_2, \ldots, \beta_H]$ and $\gamma=[\gamma_{11}, \gamma_{12}, \ldots, \gamma_{21}, \ldots, \gamma_{hI}]$ for the output and the hidden layer, respectively, with β_0 and γ_{0i} denoting the biases of each node. I and H specify the number of input and hidden units in the network and $g(\cdot)$ is a nonlinear transfer function [15], conventionally using the sigmoid logistic or hyperbolic tangent functions [1]. Consequently, each hidden node h computes a $NAR(p)$ model on the $p=I$ input nodes, which are combined to $\hat{y}$ by a weighted sum of a single output node (although multiple outputs are feasible). A MLP architecture is displayed in Fig. 1.

The task of the NN is to model the underlying generator of the data during training (parameterisation), so that a valid forecast is made when the trained NN is subsequently presented with a new, previously unseen input vector (generalisation) [16]. For parameterisation, data is presented to the MLP as a randomised set of input vectors of fixed length I formed as a sliding, overlapping window over the time series observations. The weights are adjusted by minimising the differences between network output and actuals measured by an objective function (predominantly the sum of squared errors) across all input vectors, whereby

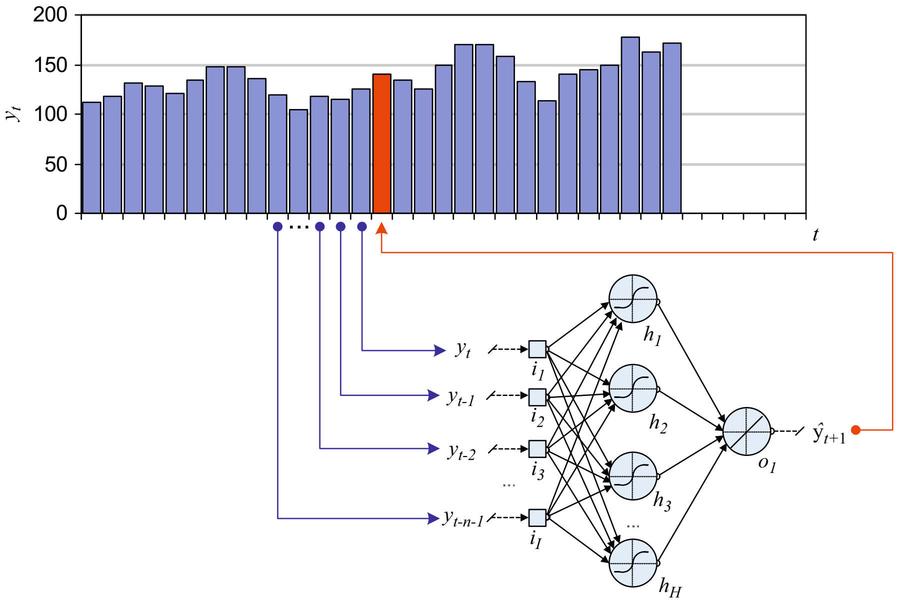


Fig. 1. Autoregressive MLP for time series forecasting.

the learning algorithm only serves to minimise the objective function given the input and output patterns for a given network architecture. Consequently, the specification of the network architecture in general, as determined through the network topology (i.e. the size and structure of the input layer I, the size H of one or more hidden layers, the number of output nodes o_j), the signal processing within nodes (i.e. the choice of activation functions $g(\bullet)$), and the information processing between nodes (i.e. the connectivity of the weights w with or without feedback and the activation strategy), and the input vector in particular, determines the fundamental capability of the MLP to capture, approximate and extrapolate the time series components from the data generating processes.

To specify these meta-parameters for forecasting, the majority of publications to date employ a variety of trial-and-error approaches and simple heuristic rules. However, only limited empirical evidence exists that the proposed heuristics resolve the problem of architecture specification [17–19], but rather result in inconsistent best practices that harm the reliability of their forecasts on different data [1,6], rendering most heuristics of limited value. To better guide the specification of NN for forecasting, a number of methodologies have been proposed in the form of either filters or wrappers [20]. In contrast to heuristic rules, methodologies provide a coherent and consistent procedural structure to modelling NNs depending on the underlying data conditions, and allow replication. Methodologies have been developed both for modeling generic data [18,21–25] or for specific data properties including financial data [26,27], telecommunication data [18], etc. (for an introductory discussion see [1]). However, to date no methodology has been universally accepted to guide the architecture specification of MLPs for time series prediction. As prior research has identified the specification of the input vector as being crucial to achieving valid and reliable results, methodologies for feature selection are discussed in more detail.

2.2. Challenges in feature selection for time series data

Feature selection aims at identifying the most relevant input variables within a dataset [28]. It improves the performance of the predictors by eliminating irrelevant inputs (and hence noise), achieves data reduction for accelerated training and increased computational efficiency [29], and often facilitates a better understanding of the underlying process that generated the data. In order to present features in the most suitable (often parsimonious) format, feature selection is comprised of feature evaluation, feature construction and feature transformation. For time series data, feature evaluation aims at detecting those input variables and dynamic lags that capture the regular time series components of level, trend and/or (single or multiple overlying) seasonality, while remaining adaptive to change of stochastic components and robust against outliers and noise. Feature construction considers the creation of new features from the input variables, e.g. through principal component or factor analysis, or in the form of exogenous dummy variables to explicitly model time series components. Feature transformation in time series aims at adequate pre-processing of features in order to facilitate better modelling, e.g. by differencing to remove trends or seasonality. As time series of similar frequency and domain may exhibit different patterns, the development of an automatic, data driven methodology for feature evaluation, construction and transformation is desirable that does not require input from human experts.

In feature evaluation a variety of methodologies exist, which may be categorised as either wrappers or filters [20]. Filters make use of designated methods for feature evaluation, analysing the properties of the data in order to limit the search space of possible meta-parameters, e.g. in the form of autocorrelation analysis, spectral analysis or stepwise regression originating from linear statistics. While filters are thus independent of a particular predictive algorithm, wrappers use the underlying algorithm to compute forecasts for feature subsets, often employing a grid-search or an exhaustive evaluation of meta-parameters, and assess the resulting forecasting accuracy to identify suitable meta-parameters. As both methodologies exhibit unique properties and different shortcomings, we explore further these in order to overcome their limitations.

Wrappers are often recognized as a superior alternative for feature evaluation in supervised learning problems, as they take the properties and biases of the inductive algorithm into consideration when forecasting the dataset in question, and have proven more popular in the computational intelligence and machine learning domain (see e.g. [30,31]). However, the application of wrappers is limited by the available computational power. While they provide an effective solution for many meta-parameters of MLP architectures, the degrees of freedom in feature evaluation from time series data depend on time series length and frequency. As an autoregressive seasonality may impact only a single lag (e.g. y_{t-12} and y_{t-24} but not y_{t-11}), the use of a fixed or flexible grid size provides no reliable solution, but requires an exhaustive enumeration. However, the search space to identify a single annual seasonality in a monthly time series requires the analysis of two or better three (to identify possible MA(q)-processes) full seasons and hence $2^{24}-1$ or $2^{36}-1$ input vector candidates of lagged variables. For weekly or daily time series of higher frequency the search space is increased to $2^{156}-1$ or $2^{1095}-1$ combinations, respectively, with further increases on intraday data. As this regularly exceeds the available computational power, wrappers are not employed for feature evaluation on high-frequency data and provide no universal methodology for time series with unknown frequencies and components. (However, wrappers with different grid sizes are routinely employed to identify other architectural components with less degrees of freedom, e.g. an adequate number of hidden units [1].)

In comparison, filters that identify only the relevant time series structure have proven more efficient in feature evaluation, and are regularly employed in statistics and econometrics. Based upon the popular Box–Jenkins methodology of linear statistics [32], the time series structure including seasonality is frequently identified as a mixture of AR- and MA-components, effectively filtering non-significant features. The specification of a parsimonious input vector requires a stepwise analysis of the patterns in the plotted autocorrelation function (ACF) and partial autocorrelation function (PACF) to identify statistically significant components of the dependent variable. Although the visual ACF/PACF analysis is itself not automated, it is feasible to formalise heuristic rules that allow an automatic algorithmic implementation (see e.g. the benchmark software Autobox [33] or ForecastPro [34]). Per se, the assumption of linearity (as of most filters based on linear theory) allows no identification of nonlinear interdependencies [35], which introduces a fundamental mismatch that may substantially bias the application of a nonlinear MLP towards linear components. In the absence of feasible alternatives, linear filters are none the less employed in identifying significant lags for NN forecasting, e.g. following Lachtermacher and Fuller [36], without careful consideration of known limitations. Early studies limited their analysis to PACF-analysis in order to identify AR-lags for MLPs [37], omitting the identification of linear MA-components. On data with multiple seasonality, the interpretation of ACF and PACF often provides ambiguous and misleading information on the individual components (e.g. on weekly data, a seasonality $s_1=13$ of week in the quarter will interact with the magnitude of an annual weekly seasonality of $s_2=52$ as it represents a multiple of the quarterly cycle, causing it to inflate or diminish depending on the sign of the shorter autocorrelation cycle). Furthermore, ACF and PACF-analysis fails to identify parsimonious lag structures for large datasets such as high frequency time series, as demonstrated in Fig. 2. As the confidence intervals are related to the sample size [38], an increase in time series length results in tighter confidence bounds (see Fig. 2a). With a growing sample size the individual autocorrelations of a constant magnitude become statistically significant, eventually causing the confidence intervals to become so tight that nearly every unrelated lag becomes significant (an effect shared by statistical significance tests employed in all variants of stepwise regression [39]), increasing the length of the input vector dramatically (see Fig. 2b). As a result, the methodologies based upon statistical test would construct non-parsimonious models that depend not on the structure of the data generating process, but merely the sample size. Consequently, ACF/PACF analysis yields no solution for nonlinear components or high-frequency data.

Another popular linear filter suggested in literature for feature evaluation of MLPs uses statistical tests in the form of stepwise regression (SR) [40–42]. The approach employs conventional regression to identify the significant AR-lags of the dependent variable and uses them as inputs for the MLP, with straightforward extensions of this approach to multivariate modelling [41], albeit only of stationary time series. However, the identified lags are typically serially correlated and lead to problems of multicollinearity, an effect even more pronounced on time series with higher frequency where the serial autocorrelation has longer memory. Also, it shares the challenge of increasing significance of its stepwise tests on large sample sizes and high frequency data with ACF and PACF analysis. The result is that the stepwise identified models are not guaranteed to include the true significant lags, which may potentially lead to selection of ill defined inputs.

An alternative filter approach to feature evaluation, spectral analysis (SA) is concerned with the exploration of the cyclical patterns in the data. It decomposes complex time series into a few underlying sine and cosine functions of particular wavelengths, thus providing information on the structure of single or multiple seasonalities [43]. Time series frequencies of high power are identified as an indication of a strong periodicity, which are then recoded as lags to allow a direct construction of input vectors for MLPs to extrapolate the periodicities. SA is mathematically equivalent to autocorrelation analysis [44], yet without information on the potential MA structure. Consequently, SA can be employed in analogy to the Box–Jenkins methodology to identify periodicities and AR-lags from time series, but shares its shortcomings in the assumption of linearity and the sensitivity to the sample size of the datasets. In contrast to the statistical tests of Box–Jenkins, SA requires the setting of a threshold depending on the dataset properties in order to facilitate the identification of the seasonalities. Setting this threshold automatically, so that the

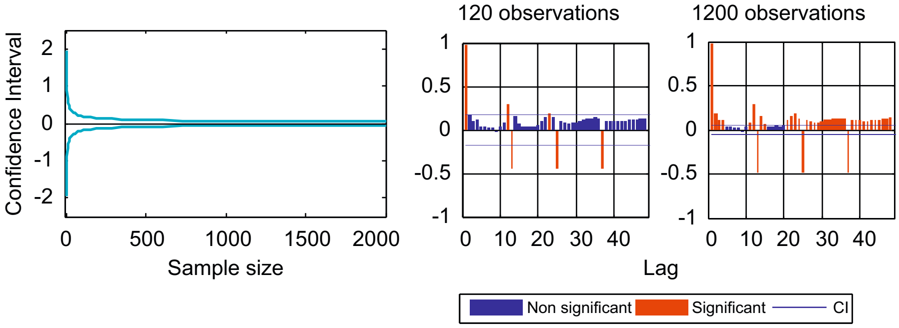


Fig. 2. Effect of sample size on confidence intervals (a) and PACF plots of a short and a long sample of an artificial time series (b).

algorithm is capable of dealing with datasets of arbitrary periodicity that contain both low and high frequency time series patterns, presents a further challenge.

Beyond feature evaluation, an aspect equally neglected by existing filter and wrapper approaches to date is the possibility of feature construction. Methodologies rarely distinguish between stochastic and deterministic seasonalities in model specification, which may require different treatment. To capture deterministic trend or seasonal components it is advisable to create additional features in the form of integer or dummy variables [45], rather than merely select lagged inputs. The conventional approach to model deterministic seasonal patterns is to use $S-1$ binary dummy variables for each time period t, where S is the seasonal length. However, for high frequency data this creates very long input vectors through the use of $S-1$ additional time series. Alternatively, one may consider a set of sine and cosine dummy variables, which have been shown to capture deterministic seasonal elements of the time series well [46]. However, the ex ante identification of stochastic or deterministic seasonality is not supported by ACF/PACF, SR nor SA, requiring subsequent manual modelling choices and limiting the automatic use of these filters for feature creation.

Furthermore, no consensus exists on approaches for automatic feature transformation. While most statistical filters require stationary time series to identify seasonal features, i.e. removal of trend and level shifts, dissimilar prerequisites exist for NN which are in theory capable of approximating any time series structure [5,47]. Linear statistical tests exist in order to identify non-stationary time series, such as the augmented Dickey–Fuller (ADF) test, similarly limited by their assumption of linearity. Furthermore, no consent exists whether a time series with identified (stochastic) seasonality should be deseasonalised first to enhance the accuracy of NN predictions [4,5,48] or seasonality be incorporated as AR- and MA-components in the NN structure [49–52]. This problem becomes especially pronounced for datasets with time series of unknown frequencies and potentially overlying seasonality, such as the datasets provided for the ESTSP'08 competition.

As a result of the shortcomings of existing filter and wrapper approaches, there currently exists no consensus on how to identify linear and nonlinear time series features across different time series frequencies, nor their treatment through feature creation nor feature transformation [8,53]. Consequently we propose a methodology for feature selection that reflects prior shortcomings and provides a non-parametric methodology for the identification of a single and/or multiple repetitive, stochastic or deterministic seasonal components of unknown length, magnitude and type in order to facilitate fully automatic MLP modelling. To combine the advantages of filters and wrappers, we will develop a novel filter for feature evaluation, combined with successive wrapper approaches for feature construction and transformation.

3. Automatic feature evaluation for time series data

3.1. Seasonal identification using an iterative neural filter

In order to identify time series features and to capture them in the input vector of a NN, we propose a non-parametric, iterative filter based on the combination of Euclidean distance estimation and MLPs. The methodology is motivated from the iterative Box–Jenkins methodology [44], and the use of simple seasonal (year-on-year) plots which forecasting practitioners frequently employ to visually identify single and multiple seasonality in times series. Although the visual analysis frequently fails to reveal complex seasonal interactions of autoregressive and moving average components, multiple overlying and interacting seasonality of different cycle lengths and nonlinear patterns, the identification can be aided by a stepwise process of model refinement and re-identification from the residuals, allowing the effective use of simple visualisations.

Finding the seasonal structure of a time series is equivalent to identifying the correct seasonality s of the input vector of a MLP that allows it to capture the seasonal information as lagged variables. Any given time series Y of length N can be split in $n=N/s$ vectors of varying seasonal length s, where $s=[1, 2, \ldots, N/2]$. For $s=1$ the maximum number of N vectors is created, each containing a single observation y_t; for $s=N/2$ only two vectors are constructed, each containing $N/2$ observations, $[y_t, y_{t-1}, \ldots, y_{t-N/2+1}]$ and $[y_{t-N/2}, y_{t-N/2-1}, \ldots, y_{t-N+1}]$. For each value of $s \neq S$ the vectors will exhibit some non-correlated pattern as a fraction or multitude of the seasonality S. When s matches the actual underlying seasonal length S, all vectors will exhibit a similar seasonal pattern with deviations only due to noise, decomposing the total variance into that caused by seasonality and by noise. Hence the input vector of length s that minimises the distance between all N/s vectors identifies a potential seasonality. We measure the distance between these vectors using the Euclidean distance. For the two-dimensional case, for two vectors $P=[p_1, p_2, \ldots, p_s]$ and $Q=[q_1, q_2, \ldots, q_s]$ the Euclidian distance is defined as

$$d(P,Q)_s = \sqrt{\sum_{i=1}^{s}(p_i-q_i)^2}. \tag{2}$$

Distances are calculated as the sum of all n pair-wise distances of equal length s. For $n \geq 2$ all combinations are considered as pair-wise distances; consider three vectors $P=[p_1, p_2, \ldots, p_s]$, $Q=[q_1, q_2, \ldots, q_s]$ and $R=[r_1, r_2, \ldots, r_s]$; for $s=1$ the distance of three pairs is measured by (p_1,q_1), (p_1, r_1) and (q_1, r_1). *For n* vectors of length s there are $n\,(n-1)/2$ pair-wise combinations. In order to compare distances across different s the Euclidian distance is subsequently divided by the number of pair-wise distances to estimate an average distance for a given s independent of the number of vectors n or their length s in the original time series. The input vector length s that minimises the distance indicates a potential seasonal length; note that for $s=1$ the time series would exhibit no regular seasonality. As an example, let us consider a synthetic time series for $t=[1, 2, \ldots, 100]$ constructed as a sine wave with a periodicity of $s=12$, with $Y(t)=\sin((2\pi t)/12)$ and no noise. Following the method described above we split the time series for different s, e.g. 5, 12, 19 and 24 as shown in Fig. 3.

All seasonalities $s \neq 12$ result in a distance $d_p>0$, interpreting seasonality as noise; for $s=12$ a zero distance d_p is measured, identifying the periodicity of the time series. However, for times series without noise the seasonal distance s would exhibit an identical mean distance for all multiples of s, $d(P,Q)_s=d(P,Q)_{ns}$. In order to accurately distinguish the shortest underlying seasonality from its multiples, and to achieve a parsimonious input vector, we penalise the mean distance of longer vectors s using a penalty factor τ proportional to the log of s:

$$d_p(P,Q)_s = \log(d(p,Q)_s+1) - \tau\log(s), \tag{3}$$

The penalty τ controls the sensitivity of the method and is empirically determined to penalise a growing seasonal length as less vectors become available to estimate the distances (we employ $\tau=0.15$ in all experiments). The minimum of penalised distances $d_p(P,Q)_s$ identifies the shortest seasonality of the time series.

In order to identify and account for multiple overlying seasonalities, that may co-exist independently or interact with

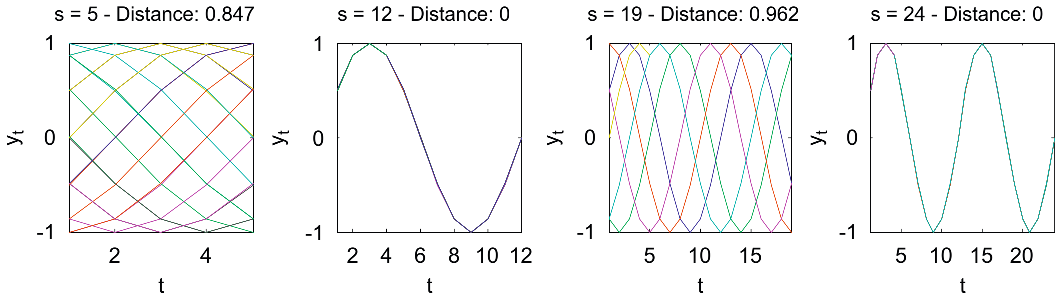


Fig. 3. Plots of vector Y for different s=[5, 12, 19, 24] with average Euclidean distance for each s.

other seasonal frequencies, the identified seasonality s needs to be iteratively filtered from the time series in order to identify less dominant patterns. We propose an iterative neural filter (INF), capable of removing any type of nonlinear seasonality in the presence of other seasonalities, trends and irregularities in the time series. (Note that the term filter is not used the sense of feature selection methodologies, i.e. wrappers vs. filters, but in the sense of filtering out noise – or components – from a time series signal.) In order not to bias the predictive modelling of the algorithm, the filter should exhibit similar functional capabilities to the algorithm of the NN employed for forecasting. We will use a MLP to estimate the INF, utilising the capability of universal approximation [2,54] but employing a distinct topology dissimilar to those used for forecasting. The inputs of the MLP do not consist of lagged realisations of the dependent variable y_t, but of contemporaneous explanatory variables that encode deterministic time series patterns. Two inputs $x_{s,1}$ and $x_{s,2}$ encode seasonality using an explanatory variable that is created uses Sin(t) and Cos(t) for an explicit representation of the point in time within an identified seasonality of length s (see e.g. [45,46]) with

$$x_{s,1}(t) = \sin\left(\frac{2\pi t}{S}\right), \text{ and } x_{s,2}(t) = \cos\left(\frac{2\pi t}{S}\right). \quad (4)$$

In contrast to the $s-1$ binary dummies conventionally used in regression to encode deterministic seasonality, the explanatory variables $x_{s,1}$ and $x_{s,2}$ code a deterministic seasonality as sine–cosine pairs for each s, as this substantially decreases the size of the input vector for long and multiple seasonalities. In addition two explanatory variables z_1 and z_2 are created that provide an explicit representation of the point in time t within the time series (which is lost in creating disjoint input vectors for feedforward NNs) by encoding the linear distance from the beginning and end of the time series N, with $t=[1, 2, \ldots, N]$, and

$$z_1(t) = t \text{ and } z_2(t) = N - t + 1. \quad (5)$$

These variables facilitate a representation of structural changes of the level of the time series, i.e. different forms of trend or level shifts, which may interact with the periodic seasonal signals. Both variable pairs $x_{s,i}$ and z_j aid the MLP in identifying (interacting) trend and seasonality simultaneously, in contrast to prior transformation and subsequent modeling, effectively enabling the MLP to capture and model non-stationary time series.

The MLP architecture itself is kept consistent (also in all subsequent experiments) for reasons of simplicity and to facilitate replication. Preliminary experiments across trials of single and multiple seasonality, different magnitudes and different noise levels indicated the need for a comparatively large number of hidden nodes to capture complex periodic signals, regardless of the number of input nodes. We chose a constant topology of 16 hidden nodes arranged in a single hidden layer with hyperbolic tangent activation functions, and a single linear output node y_t for output. As the objective of this MLP is not to forecast or generalize on unseen data, but only to approximate in order to filter out structure, no test set is used during training, withholding merely S observations for validation purposes and using all remaining N–S observations of the time series for training. All contemporaneous inputs are linearly scaled between $[-1, 1]$. The weights of the MLP are randomly initialized once for each iteration of estimating the INF. The network is subsequently trained using a standard backpropagation algorithm: input patterns of the deterministic variables $x_{s,1}$, $x_{s,2}$, z_1, and z_2 for a point in time t are shown to the network, which learns the mapping of these inputs to the target output of the actual time series observation y_t by minimizing a squared error loss function. The result is a heteroassociative nonlinear filter that approximates only the deterministic time series patterns of level, trend and seasonality of pre-specified length which are provided as inputs, but no other patterns. Note that different architectures and training algorithm were tried but yielded similar results; however, data and domain-specific architectures may yield even more robust results and more parsimonious models. Also, the MLP maybe initialised several times to provide more robust filter results given the stochastic nature of MLP training.

The network output o_t, which expresses the regular structure of the time series as captured by the MLP, is subsequently subtracted from the original time series y_t, effectively creating a filtered time series from which the dominant pattern of the periodic signal has been removed. Following this, the process is repeated in order to stepwise identify and eliminate further remaining periodicities. In successive iterations, seasonal components of different length s to the one identified before are added as additional pairs of sine–cosine inputs $x_{s,1}$ and $x_{s,2}$ to allow a simultaneous filtering of multiple seasonalities. The process is repeated until the most prominent period identified is $s=1$, which implies that no further seasonality is present in the time series, as illustrated in Fig. 4.

The seasonal identification follows the established tradition of iterative modelling in the ARIMA context, and is equivalent to a stepwise decomposition of variance into structural components of seasonality starting with the most dominant pattern. The iterative nature of the proposed algorithm offers the advantage that should a seasonality s not be fully filtered by the MLP, it may be identified again in the following iteration until it is fully removed. Using pair-wise comparisons enables the identification not only of deterministic seasonality, but also stochastic seasonality and seasonality with structural breaks, as the pair-wise comparisons also identify similarity within disjoint parts of the time series where a homogeneous similarity across the whole series cannot be found. The INF also offers an advantage regarding its interpretability, as the filter is based upon the simple distances of vectors within a seasonal diagram, a heuristic established with practitioners that may be visualised to allow an analysis of the identified seasonal structure. In addition, the periodicity of the identified seasonalities allows inference of the frequency in which

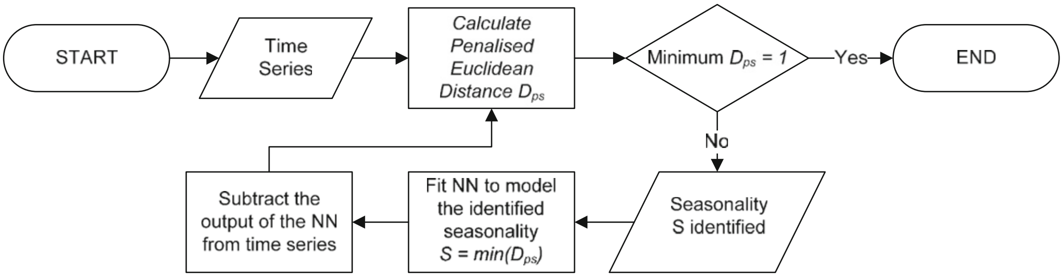

START
Time Series
Calculate Penalised Euclidean Distance D_{ps}
Minimum D_{ps} = 1
Yes
END
No
Seasonality S identified
Fit NN to model the identified seasonality $S = min(D_{ps})$
Subtract the output of the NN from time series

Fig. 4. Flow chart of the iterative filter.

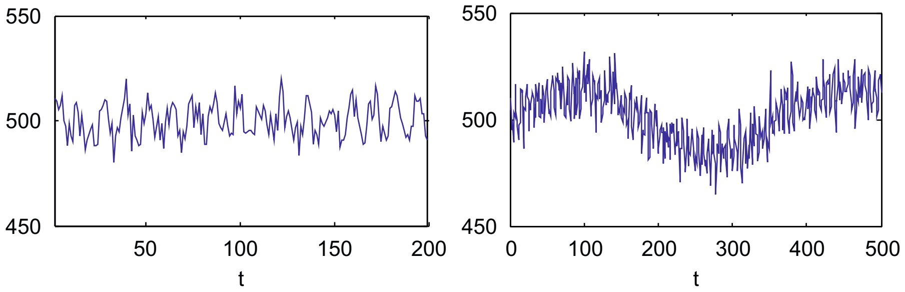


Fig. 5. The synthetic time series A and B (only first 500 observations).

the time series had been measured. Consequently, the identified seasonalities allow a general, non-parametric insight into the structure of the seasonal components of the time series, which may also be used for other algorithms of computational intelligence and linear statistics.

3.2. Experimental illustration of the iterative neural filter

Conventionally, the INF algorithm processes the information fully automatically without graphical output or user intervention, which may impair the understanding of its process. To illustrate its iterative functionality, we visualise the intermediate output after each step of the INF for two synthetic time series, plotted in Fig. 5.

The time series are constructed using a constant level and no trend, with different components of single and multiple overlying seasonality (see Appendix A for the equation and parameters). Time series A contains 200 observations with a single seasonality $S_1=12$ (representative of monthly data), time series B contains 1500 observations with double seasonality $S_1=7$ and $S_2=365$ (representative of daily data). We provide the intermediate graphical output after each step of the INF in (a) estimating the Euclidian Distance for each $s=(1, 2, \ldots, N/S)$ to identify the minimum penalised distance d_{ps} (indicated in the graph by a cross), which is used to specify the seasonality s for the input variables $x_{s,1}$, $x_{s,2}$, z_1 and z_2 of the MLP, (b) the output of the MLP using only the identified input variables for s to match the output of the actual time series, and (c) the corresponding residuals of the MLP output. The plots for time series A are shown in Fig. 6; the plots with additional iterations for double seasonality of series B are shown in Fig. 7.

The analysis of seasonal distances on the original time series A (Fig. 6.1a) identifies a minimum mean Euclidian distance for $s=12$. Consequently the MLP is fitted with four inputs to include two deterministic seasonal variables $x_{s,i}$ that encode a sine and cosine of length $S=12$ and the two time indicators z_j, each in a single input node; after parameterisation, the network output depicted in (Fig. 6.1b) shows clear seasonality of the same amplitude and frequency as the original time series in Fig. 5a, resulting in stationary and uncorrelated residuals after deducting the network output from the time series observations (Fig. 6.1c). The residuals contain no additional seasonal information, which is verified by running the seasonal identification of the INF again to determine the minimum Euclidian distance for $s=1$ in the 2nd iteration (Fig. 6.2a). The methodology therefore identifies only the correct seasonality from the time series, stopping after a single iteration.

For time series B the plot of the penalised seasonal distances (Fig. 7.1a) identifies a first minimum of $d_{ps}=364$, which is subsequently used to fit a first set of sine and cosine variables $x_{s,i}$ with $S=364$ plus two time indicators z_j, to the MLP. The network output of the seasonal pattern is shown in Fig. 7.1b and the residuals after subtracting the MLP output from the original series in Fig. 7.1c. (Note that only the first 50 observations are plotted to limit visual clutter and allow identification of the remaining systematic pattern in the residuals.) A repetitive, seasonal pattern of shorter time series frequency is apparent in the residuals (Fig. 7.2a), initiating a second iteration of the process on the residuals. The penalised Euclidean distance identifies a second seasonal frequency of 7; thus the NN inputs are updated to include a second pair of sine–cosines with periodicity $S=7$. Following network retraining on the original time series we compute network output, residuals and identify an optimal distance of $d_{ps}=1$, which signifies the absence of further seasonality in the time series, aborting the iterative search algorithm. The MLP successfully captures both overlying seasonalities of $s_1=364$ and $s_2=7$ using 6 deterministic inputs, identifying the dominant seasonality (that explains most of the variation in the Euclidian distance) first, followed by the less dominant one. The so identified time series components of seasonality of time series A and B, explicitly captured by the explanatory time series $x_{12,1}$, $x_{12,2}$, $z_{1,}$ z_2 and $x_{7,1}$, $x_{7,2}$, $x_{364,1}$, $x_{364,2}$, $z_{1,}$ z_2, respectively, are later fed to the MLP for the actual prediction.

3.3. Accuracy and robustness of the iterative neural filter

To demonstrate the accuracy of the proposed filter under different data conditions, and to compare its accuracy with that of

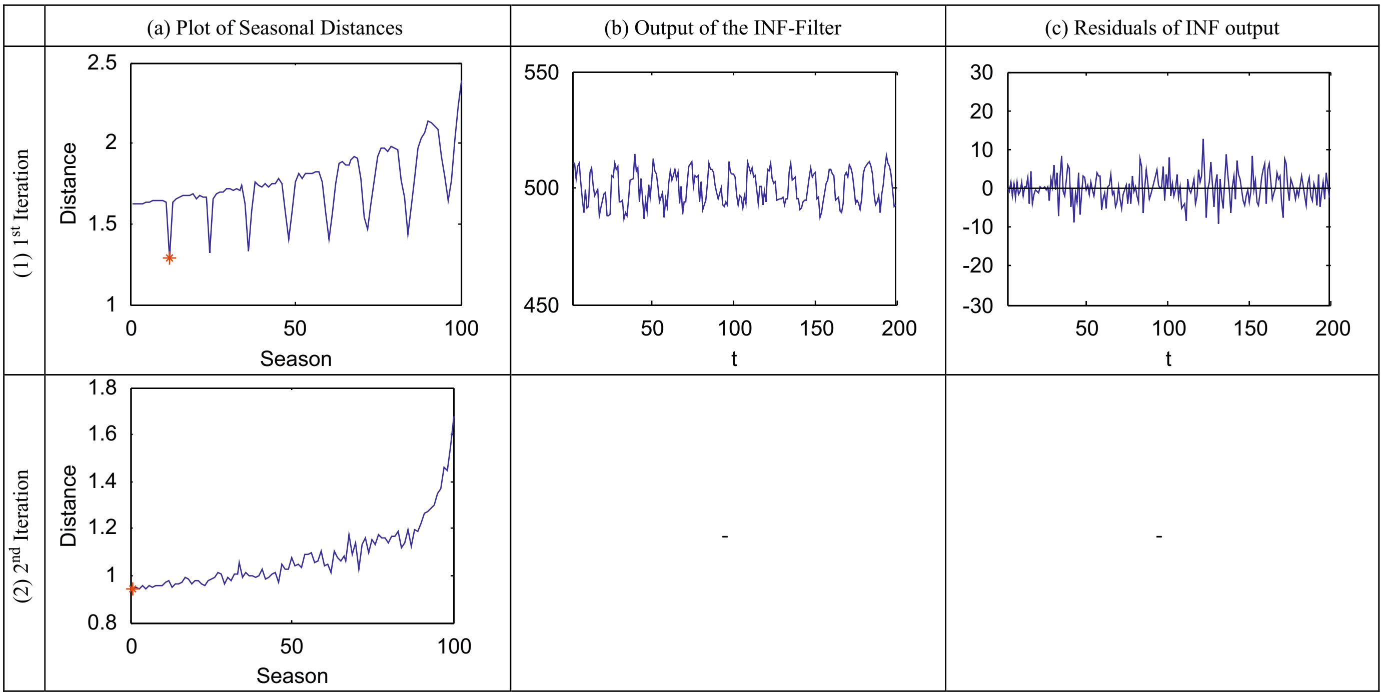


Fig. 6. Iterative outputs of (a) seasonal distances, (b) MLP filter and (c) MLP residuals for time series A.

established benchmark filter techniques for feature selection, we conduct a simulation experiment on an extended set of synthetic time series. The time series are designed as a balanced sample with different data conditions: all series have an identical constant level, using different components of single and multiple overlying seasonality, seasonality of different magnitude and different noise levels, thereby extending the data used in the illustration above. Two of the synthetic time series A.1 (denoted as A in Section 3.2) and A.2 are constructed to mimic the properties of monthly data, both using 200 observations with a single seasonality $S_1=12$ with different noise levels. In addition, two synthetic time series B.1 (denoted as B in Section 3.2) and B.2 are constructed representative of daily data, both using 1500 observations with double seasonality $S_1=7$ and $S_2=365$, but with different noise levels. Furthermore, one synthetic time series C.1 is created with 200 observations without seasonality in order to evaluate the algorithms' sensitivity to the absence of patterns. All equations and parameters used to construct the time series are provided in Appendix A for replication.

To evaluate the efficiency and robustness of the proposed INF algorithm for automatic feature evaluation we compare its precision in identifying only the correct seasonality with that of three established statistical filtering methods (discussed in Section 2.2): spectral analysis (SA) using periodograms derived from fast Fourier transforms, the analysis of autocorrelation functions (ACF) and of partial autocorrelation functions (PACF). Table 1 summarises the seasonalities identified by the proposed INF, SA, ACF and PACF analysis. Due to space constraints, for SA only the largest periodicities identified from the periodograms, and for ACF/PACF only the strongest correlations are presented in the table; in addition the table lists the total number of significant variables identified by the algorithm to show the resulting length of the input vector.

The results of SA, ACF and PACF in Table 1 confirm the theoretical shortcomings of these algorithms discussed in Section 2.3. Due to sample size restrictions, the SA does not approximate the correct frequencies, in many cases omitting the periodicity and introducing several artefacts. While the true seasonality of $s=12$ for A.1 and A.2 is identified, all periodograms identify additional, spurious seasonalities, and even identify periodicities in C.1 that are purely due to randomness (created using a particular distribution). On series with double seasonality, SA identifies the high frequency of 7 correctly, but also a false low frequency seasonality of 375 plus additional, non-existent ones. Outputs such as these require the interpretation of a human expert, limiting the ability to automate the process of feature selection. These shortcomings will result in misspecified, non-parsimonious MLP input vectors, introducing randomness from non-existing periodicities to the model that may impair learning. Similarly, ACF and PACF equally fail to identify the true underlying seasonality even on low frequency series where values of $s/2$ and around s are identified, a common problem for sinusoidal seasonal patterns. For longer time series of high frequency data, almost all lags become statistically significant (at a 5% significance level) as a result of the tight confidence bounds (caused by the large sample sizes). This creates long, non-parsimonious input vectors of over 600 input nodes, and voids any interpretation of the results.

In contrast to the benchmark methods, the INF identifies all seasonal patterns accurately from the time series, without expert guidance or user intervention, and models them parsimoniously using only few input nodes. Therefore the INF is capable of an automatic identification of only the correct frequencies of the seasonalities present in the time series, without any 'false positives' of identifying irrelevant periodicities (see Table 1). For series A.1 and A.2 the single seasonality of $s=12$ is accurately identified in a single iteration of the process, while for time series B.1 and B.2 with multiple seasonality the dominant seasonality explaining the most variation in the Euclidian distance is identified first, followed by the less dominant one. Note that for time series B.1 and B.2 a seasonality of 364 was identified, instead of the 'true' seasonality of 365. This is due to the effect of aliasing – the interaction of low and the high frequency periodicities in the time series (i.e. caused by 52 multiples of a seasonality of 7, the higher frequency). While it is feasible to correct for this effect,

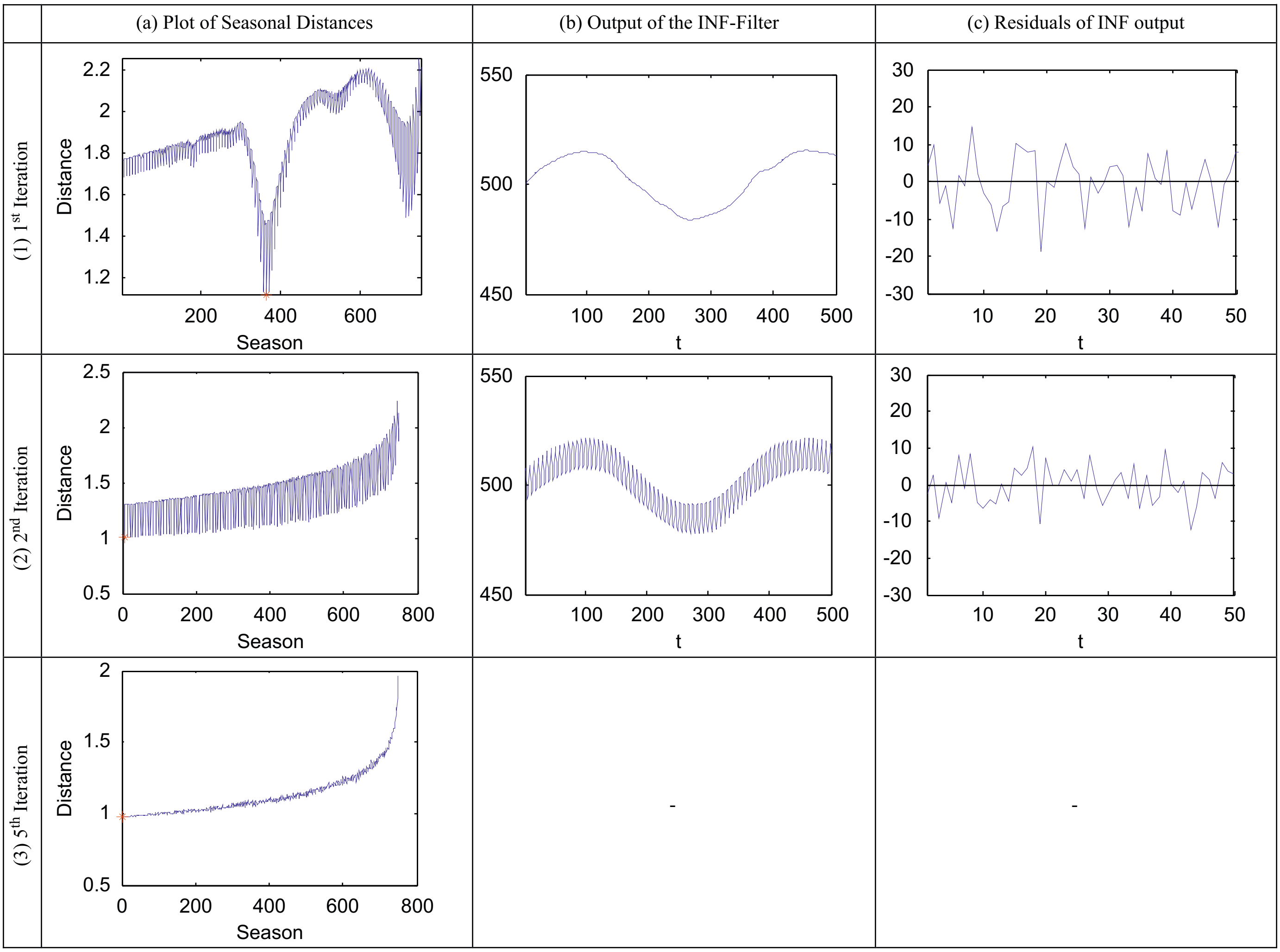


Fig. 7. Iterative outputs of (a) seasonal distances, (b) MLP filter and (c) MLP residuals for time series B.

Table 1
Identified seasonalities from synthetic time series by algorithm (in order of significance).

Series	True s_i	INF		SA		ACF		PACF	
		# Vars	Final lags	# Vars	Top 5 lags	# Vars	Top 5 lags	# Vars	Top 5 lags
A.1	12	4	12	8	11.8, 4.4, 2.4, 4.1, 4	18	1, 7, 24, 6, 12	6	1, 7, 4, 5, 23
A.2	12	4	12	9	11.8, 22.2, 200.0, 14.3, 10.5	20	6, 18, 12, 1, 13	6	1, 6, 13, 5, 3
B.1	7, 365	6	364, 7	38	375.0, 7.0, 214.3, 2.4, 2.2	673	7, 14, 21, 8, 1	36	1, 8, 5, 2, 30
B.2	7, 365	6	364, 7	41	7.0, 375.0, 7.1, 250.0, 2.3	661	14, 7, 21, 28, 35	37	1, 4, 5, 8, 6
C.1	–	0	–	6	2.0, 2.6, 11.8, 3.8, 2.7	2	8, 17	0	–

prior experimentation verified that applying the misidentified frequency of 364 provides a better fit and lower out-of-sample forecast errors than the true seasonality of 365, regardless of the forecasting algorithm (a comprehensive analysis of aliasing is beyond the scope of this paper). For time series C.1 without seasonality no periodicity is identified. We therefore conclude that the algorithm overcomes the limitations of common statistical approaches for feature evaluation by parsimoniously capturing only relevant seasonality through deterministic explanatory variables. The time series features of trend and seasonality identified by the INF that are capable of extracting all seasonal information from the time series are explicitly captured by the explanatory time series $x_{s,I}$ and z_j, respectively. The same input vector is subsequently used in training of the MLPs for prediction, providing a coherent methodology of capturing and extrapolating arbitrary periodicities in a fully automatic way.

4. Automatic feature construction and transformation for time series data

Although the INF accurately evaluates and identifies all relevant features form the time series, it cannot overcome the challenge of distinguishing between stochastic or deterministic seasonality, which is required for an adequate feature construction of dummy variables. Similarly, the INF provides no insight into the most suitable pre-processing of the time series through feature transformation, such as single differencing for non-stationary time series or seasonal differencing for seasonal ones. In order to enable the MLP to also capture non-seasonal AR- and MA-processes, and stochastic or mixed form seasonality, we need to consider additional input vector candidates to account for additional autoregressive lags in feature evaluation, feature construction of explanatory variables for deterministic patterns and feature transformation through seasonal and trend differencing for stochastic patterns. Only a limited amount of options exist for feature construction and transformation. Therefore we propose a simple wrapper around the proposed INF to create additional input vector candidates, combining the effectiveness of wrappers in low dimensional problems of feature construction and transformation with the efficiency of filters for high dimensional problems of feature evaluation.

Each time series is first explored using the proposed INF algorithm (see Section 3) to identify any cyclical and seasonal data frequencies. The identified time series components provide the first set of input vector candidates by including the specified sine and cosine-explanatory variables $x_{s,l}$ and z_j for each of the identified seasonalities $s_1, s_2, \ldots, s_n$ into the input vector, encoding deterministic seasonality and trend [55]. In order to capture any stochastic seasonality, the input vector can be further extended to integrate time lagged realisations of the dependent variable in the form of nonlinear $\mathrm{AR}(P)_s$ terms, as suggested in previous studies [56]. Beyond the time series components of trend and season, the time series may also exhibit additional auto- and/or cross-correlated structures of non-seasonal length, $n \neq s_i$, which must equally be captured in the input vector. We therefore employ the conventional linear parametric approach of stepwise regression (at a 5% significance level) to include the most significant seasonal and/or non-seasonal lags, creating additional input vector candidate models. The stepwise regression is primed with the inputs identified by the INF to account for deterministic seasonality and instationarity.

No consensus exists on whether a time series with identified trend should be detrended, and whether a seasonal time series should be deseasonalised first to facilitate NN learning [4,5,48]. Moreover, valid identification of significant autoregressive lags through a stepwise regression requires stationarity of the data, which is explicitly violated in trended or seasonal time series patterns. Yet, a prior removal of trend and/or seasonality may impact on the lags identified as significant, effectively altering the input vector. In the absence of best practices, we consider the original, detrended and (for each of the n identified seasonalities s_n) deseasonalised time series as candidates for a further wrapper of feature transformation. To detect integrated time series we employ the well established Augmented Dickey–Fuller (ADF) test for non-stationarity (i.e. trend or structural breaks through level shifts). For non-stationary time series we create candidates of (a) the original time series $I(0)$ and (b) the detrended time series using (first or second order) differencing $I(d)$. For seasonal time series we consider candidates of (a) the original time series $I(0)$ and (c) deseasonalised time series using seasonal differences $I(d)s_i$, and for trend-seasonal time series all combinations (a), (b), (c) plus (d) candidates of detrended and deseasonalised time series $I(0)$ $I(d)s_i$. Note that the first or seasonal differenced time series are used only for the stationary identification of the input vector through stepwise regression – the NNs are trained only on the original, undifferenced time series containing all patterns.

The input vector candidates identified through feature selection serve as a starting point for additional model specification of the remaining MLP network architecture. Following the identification of all possible feature candidates, a set of MLP architectures is conventionally specified via wrappers to evaluate hidden nodes, layers, and activations functions for each input vector, and trained to predict the unseen data. From all combined candidate models the best ones are selected to form an ensemble for the final prediction, as will be described in the consecutive sections.

5. Empirical evaluation on the ESTSP'08 competition

5.1. Exploratory data analysis and feature selection

To verify the performance of the proposed iterative neural filter we submitted predictions to the forecasting competition of the 2008 European Symposium on Time Series Prediction (ESTSP'08). The ESTSP'08 competition provided three time series of different length and structure (displayed in Fig. 8), without any

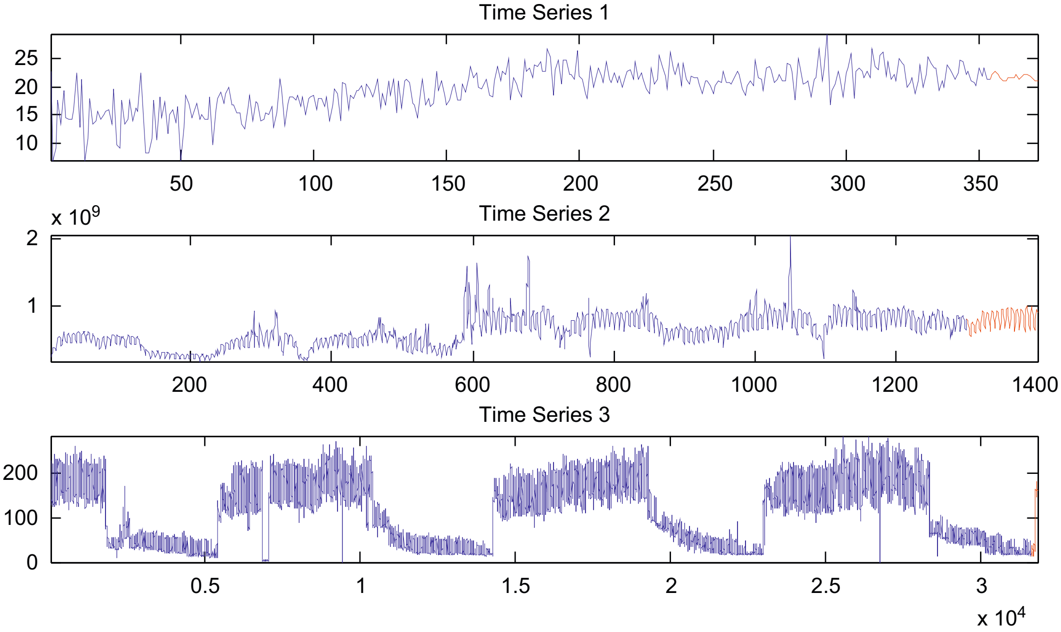


Fig. 8. The time series and forecasts of the ESTSP'08 forecasting competition.

domain information on the origin, structure, components or frequency of the time series. The objective was to predict all three time series as accurately as possible for multiple steps ahead into the future, measured on true ex ante forecasts from withheld data using the average *MSE*.

The first time series of the competition, plotted in Fig. 8a, is comprised of 354 observations, and was provided with two additional explanatory time series of equal length as potential inputs; as no future inputs for the explanatory series were provided these would require separate predictions to serve as contemporaneous predictors. No domain knowledge on the time series itself was provided. The objective is to forecast the next 18 values. Applying the INF using the penalised Euclidean distance, a single seasonality of 12 observations was identified, indicating a monthly time series of 29.5 years of data with a month in the year seasonality. The ADF test indicates the absence of trend, suggesting two options to model the input vector regarding pre-processing of the time series: identifying significant lags using stepwise regression on the original time series (1.a) and identifying lags after taking a 12th order seasonal difference to remove possible stochastic seasonality (1.b).

The second time series of the competition (Fig. 8b) contains 1300 observations without explanatory time series. The objective is to forecast the next 100 values. The ADF test identifies an instationary time series caused by a structural break in the form of a single level shift, visible in Fig. 8b (for an automatic identification of level shifts, see [57]). As a consequence no first order differencing of the data to compensate for trend-instationarity is required; we introduce a binary dummy variable to code the identified level shift. Using the INF we identify seasonalities of 7 and 364 observations, indicating a time series of daily observations with day in the week seasonality and day in the year seasonality. The significant lags are identified on the original time series (2.a), applying a 7th order differencing (2.b), a 364th order differencing (2.c) and both differences (2.d) in order to identify possible additional significant lags as input vectors candidates.

The third time series of the competition (Fig. 8c) contains 31,614 observations; the objective is to predict the 200 next values. The ADF test identifies no unit root, i.e. no indication of a trend. The INF identifies three overlaying seasonalities of 24, 168 and 8760 observations, indicating an hourly time series with hour of the day, day of the week, and day in the year seasonalities. This provides several alternatives to input vectors specification by applying different levels of seasonal differencing, including the original series (3.a), the 24th (3.b), the 168th (3.c), and the 8760th differenced series (3.d), plus four combinations of the seasonal differences.

5.2. Architecture specification and training

The input vector candidates of feature selection are embedded in a further wrapper to specify different candidate MLP architectures for each of the identified feature candidates, resulting in a combined filter-wrapper approach. Due to the possible interaction of the time series structure, the input vector and its encoding with the number of hidden nodes, we evaluate different MLP architectures for every input vector candidate. Following the input vector specification, we construct a set of MLPs using a stepwise wrapper that evaluates a range of hidden nodes n_{HI}=[2, 4, 6, 8, 10, 12, 14] for a single hidden layer, and the Hyperbolic Tangent (TanH) and the Logistic (Log) activation functions f_{act}=[TanH, Log] for each input vector candidate for consideration in model selection. All other parameters remain unaltered, applying a single output node with the identity function for an iterative multiple step-ahead trace forecast, and a conventional feedforward topology of a MLP.

Each MLP is trained using backpropagation with momentum for 1000 epochs or until an early stopping criterion is satisfied. (Alternative learning algorithms including Levenberg–Marquard were also evaluated, without any significant differences in accuracy.) For the early stopping criterion the *MSE* is evaluated every epoch, and training is halted if the *MSE* does not improve by 1% in 100 epochs. The initial learning rate is set to $\eta=0.5$, applying a cooling factor $\Delta\eta=0.01$ to reduce the learning rate in each epoch; the momentum term is kept constant at $\varphi=0.4$. All data is pre-processed using linear scaling into the interval of $[-0.6, 0.6]$ to allow for headroom on non-stationary time series, and presented to the MLP using random sampling without replacement. In order to avoid local minima during the training and to provide an adequate error distribution using sufficient results, each MLP candidate is initialised 40 times with random starting weights in the interval of $[-0.6, 0.6]$. To facilitate training and a simulated out-of-sample evaluation prior to the submission to the competition, and without knowledge of the true test-data, each of the time series was split in three sequential data subsets for single fold cross-validation, using 60% for training, 20% for validation and 20% for testing, respectively.

5.3. Model selection

Given the large number of alternative input vector candidates and architecture candidates created for each time series, the selection of a single MLP, which promises accurate and robust out-of-sample performance on unseen data is highly challenging. The limited evidence of NN in time series prediction, and, in particular, their low consistency and robustness of performance across homogeneous datasets [6] can in part be contributed to suboptimal model selection using a single-fold cross-validation, as has been discussed in detail in forecasting literature [58,59]. Best-practice approaches, such as *k*-fold or leave-one-out-(*LOO*)-subsampling regularly employed in machine learning, face particular challenges in forecasting due to the serial dependency of observations in a time series. To obtain more accurate results given the single fold evaluation, within each data subset of length *n* we employ a rolling forecast origin evaluation, e.g. estimating the average performance metrics of each model candidate across *m* trace forecasts, instead of a single fixed origin as is common practice. This scheme overcomes the shortcomings of fixed origin evaluation, like its dependence of the randomness contained in the particular forecasting origin and the limited sample size of errors [60]. In rolling origin evaluation, the number of time origins depends on the maximum length of the input vector *l*, and the data subset length *n*, with $m=n-l$. For a forecast horizon *h* we produce $h(h+1)/2$ different forecasts, instead of only *h* under the fixed origin evaluation. Therefore, we obtain more forecast errors that allow a more accurate estimation of the forecasting error distributions and hence model selection.

To further limit the impact of overfitting to a data subset or origin in model selection, we consider an ensemble of diverse candidates to generate average predictions. In addition to substantial evidence that ensembles of simple methods perform well in classification tasks, similar findings have been confirmed for time series prediction, e.g. at the M3 competition, where a simple average of all competing methods performed better than each of the competing methods itself [3]. Consequently we rank all parameterised MLP candidates for each time series, select the 10 models of the highest rank and average their forecasts $\hat{Y}_{t+h}$ for each future horizon *h*. The ESTSP'08 competition assesses the accuracy of the models using a normalised *MSE* for each time series averaged over all three series. In order to align the performance metric used in model development with the metric used for the final evaluation of the competition, we parameterise and select models using the *MSE* despite its well explored and documented shortcomings in empirical evaluations (e.g. the sensitivity to outliers, etc.; see [61,62] for a discussion).

The combined filter-wrapper approach of feature selection and architecture specification resulted in a large number of MLP candidate models, of which different candidates may be combined in the ensemble for the final prediction. Due to the large number of candidates a comprehensive overview of the individual candidate models and architectures actually employed in the ensembles is infeasible given the space constraints of this paper. However, we will briefly discuss some general findings in order to indicate potential learnings from our methodology: 80% of the top 10 candidate models (which were used to create the composite ensemble forecasts) for each of the time series use the sine/cosine variables identified by the INF to code seasonality of different lengths, implying that this methodology aids the model in capturing the complex overlying seasonal forms. For those candidates for which an input vector was identified using both the original and the differenced time series, both alternatives were always selected to be within the top 10 across all time series, implying that these approaches are complementary. An unexpected finding was that the univariate models for time series 1, excluding the two provided time series of explanatory variables, outperformed all multivariate models that used this information. This reduced the complexity of creating the final forecasts, as no predictions for the explanatory time series were required and no accumulation of the errors could arise from their inaccurate forecasts to impact the final forecasts. Consequently, all variables contained in the input vector were univariate lags or related to explanatory encodings of the time series' components.

Regarding network topology, no coherent structure of the number of hidden nodes could be identified for a particular time series. For time series 1 and 2 most candidates ranked highly applied a TanH-activation function, while candidates for time series 3 used the Log-activation function. With the increase in time series frequency on the ESTSP'08 data, and the resulting increase of observations per annual season, both the data volume and the length of the input vector to capture a full season increase proportionally. Consequently, high frequency time series resulted in longer input vectors. Across all input vectors of candidate models for monthly data, the methodology employs an average of 7 input nodes with the longest time lag identified as *t*-36. For daily and hourly time series, the methodology utilises an average of 30 and 354 input nodes, and a maximum lag in the input vector of *t*-392 and *t*-9072, respectively. Reflecting upon the large number of input and hidden nodes, a candidate model developed for the hourly time series would use 2478 parameters on average, in comparison to only 49 for the monthly time series. The implications of this for MLP training are substantial, considering the intricacy and time to solve such a complex optimisation problem in the light of a limited amount of training vectors. Furthermore, our experiments identified a positive correlation between the frequency of the time series and the size of the search space required to find suitable input lags (represented by the maximum time-lag in the input vector). Not only does the input vector for time series of higher frequency increase in size, the maximum time lag to be considered also moves further into the past. Most methodologies which identify the input vector based upon wrappers, grid search, exhaustive random search, genetic algorithms and other meta-heuristics using on computational force are bound to encounter constraints in providing valid and reliable results in a reasonable time frame. In contrast, the filter approach based upon the INF and iterative stepwise regression to identify the appropriate lags increased computational times only proportional to the increase in the search space, providing solid identification of the relevant time lags for forecasting in an acceptable time. Still computational time varied substantially, ranging from virtually instantaneous for the time series 1 and 2 to several days for time series 3 using a pair of dual core processors at 2.4 GHz with 10 GB of RAM.

5.4. Preliminary results of the ESTSP'08 submission

Prior to submission, the MLP ensemble forecast for each time series was compared with a series of statistical benchmark forecasting methods, in order to assess the potential gain in accuracy in comparison to the increased complexity of MLPs. We compared three different benchmark models for each time series: a random walk, $\hat{y}_{t+h}=y_t$, which assumes that the future forecasts $\hat{y}_{t+h}$ for all horizons h are equal to the present value of the time series y_t. In addition, we evaluated single and seasonal exponential smoothing models (EXSM), with the seasonal length set to match the longest seasonal cycle of each time series (as identified in Section 3.1) as this would include multiples of all shorter periodicities. Information on statistical benchmarks and their parameterization can be found in [38,63]. Table 2 contains the *MSE* errors of the NN ensembles and the statistical benchmarks for time series 1, 2 and 3, respectively. The method with the lowest *MSE* per data subset is indicated in bold italics.

The MLP ensemble, employing the INF for feature evaluation and wrappers for feature construction, feature transformation and architecture selection, outperforms the statistical benchmarks on all three time series, and across all data subsets of training, validation and test. The dominance of the MLP ensembles identifies significant gains in accuracy using our proposed methodology, and its robustness across data subsets. The same composite ensemble forecasts were subsequently submitted to the ESTSP competition (see Fig. 8).

Note that the true ESTSP'08 competition test set was withheld and has not been released to date, so that we cannot provide error measures or actuals for the true ex ante forecasts. However, in the overall ranking of the ESTPS'08 competition our approach ranked 2nd.

6. Conclusions

This paper proposes an initial methodology for automatic modelling of MLPs for time series with arbitrary time frequencies, seasonalities and trends, evaluated on synthetic time series and

Table 2
MSE of ESTSP'08 time series 1, 2 and 3.

Model	Time series 1			Time series 2			Time series 3		
	Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
Random walk	13.59	6.63	9.99	6.11^{E+16}	5.78^{E+16}	5.38^{E+16}	1997.54	1399.56	1530.28
Single EXSM	7.31	4.05	4.78	5.89^{E+16}	5.59^{E+16}	5.04^{E+16}	1770.96	1230.68	1310.10
Seasonal EXSM	5.95	3.81	5.38	1.11^{E+17}	3.40^{E+16}	4.31^{E+16}	1409.56	1120.93	1161.48
MLP ensemble	***4.33***	***3.32***	***4.76***	***2.50^{E+16}***	***1.63^{E+16}***	**1.01^{E+16}**	***353.65***	***430.33***	***971.07***

the true ex ante predictions of the ESTSP'08 competition. An iterative neural filter is proposed for feature evaluation to automatically identify the frequency of the time series, embedded in wrappers for feature construction, feature transformation and architecture selection. The principle of the methodology is to utilise the efficiency of filters for known and well specified properties, and to employ wrappers for additional modelling choices with limited search spaces, effectively combining the best practices of statistics with those of computational intelligence. As a result we construct a large set of competing candidate models of MLPs with different input vectors utilising varying temporal information on trends, stochastic and deterministic seasonality through autoregressive and/or dummy variables, respectively. In addition, wrappers are employed to create candidate models for a number of hidden layers, hidden nodes and activation functions. In order to omit the need for manual intervention we employ a composite ensemble forecast of the 10 best models selected on their sample performance for each time series. The proposed methodology, which is based on established tools and methods, avoids arbitrary modelling decisions and manages to overcome the challenges of modelling heterogeneous sets of time series with varying time frequency and time series patterns, a task where most methodologies developed to date fall short.

Using this automatic forecasting methodology for MLPs we took part in the 2008 ESTSP forecasting competition. We outperformed a set of benchmark methods and achieved a good ranking for each time series in comparison to state-of-the art algorithms from statistics and computational intelligence. Overall, our proposed methodology ranked 2nd, demonstrating that a fully automatic, purely data driven methodology that requires no expert human intervention to specify NNs is feasible and can perform well.

The novel feature evaluation algorithm of the iterative neural filter (INF), used to automatically identify the number and the period of the seasonalities that are present in a time series, is the nucleus to our methodology. In this study we discussed the underlying functionality and demonstrated its performance using a set of synthetic time series and the ESTSP'08 dataset. However it is necessary to run representative Monte Carlo simulations to access its robustness, sensitivity and power on different data conditions of time series length and patterns. The adequacy of aliasing should be carefully monitored for the proposed INF and explored for other forecasting algorithms, in particular for the case of non-sinusoidal, discontinuous seasonality. Furthermore, it may prove worthwhile to explore the potential of anti-aliasing techniques in order to further increase accuracy in forecasting applications. The proposed methodology aimed to overcome some of the challenges encountered automatic NN modelling, in particular towards high frequency data. Given the current computational resources, high frequency data remains extremely demanding and limits the amount of ad-hoc experimentation, in particular for wrappers and model ensembles. In comparison to monthly and daily data, an hourly time series contains 24 and 720 times more observations, respectively, within an identical time period. The resulting increase in sample size creates various unique challenges that are beyond the scope of this paper: increasing input vector length to capture full seasons, handling the increasing degrees of freedom in training, and the resulting computational time – both in model identification and parameterisation. However, it is on these high-frequency datasets that NN and other nonlinear algorithms of computational intelligence have demonstrated preeminent performance in recent empirical forecasting competitions such as the ESTP'08, and it is here that they may prove their worth against the established statistical benchmarks in future applications.

Table A1
Properties and parameters used to construct the synthetic time series.

Series	Periodicity	Sample	l	α_1	α_2	S_1	S_2	Σ
A.1	12	200	500	8.40	–	12	–	5.62
A.2	12	200	500	13.53	–	12	–	15.81
B.1	365, 7	1500	500	7.53	14.44	7	365	5.78
B.2	365, 7	1500	500	20.23	13.09	7	365	10.85
C.1	1	200	500	–	–	–	–	1.00

Appendix A

The synthetic time series to evaluate competing feature selection algorithms are constructed using

$$Y_t = l + \sum_{i=1}^{k} \alpha_i \sin\left(\frac{2\pi t}{S_i}\right) + e_t, \tag{6}$$

where l denotes the level of the time series, and k the number of sines with periodicity of S_i to model seasonality of amplitude α_i, which is randomly chosen from a random uniform distribution $U(1, 20)$ for each of the k sines. Each time series is overlaid with random noise e_t with Gaussian distribution $N(0, \sigma)$, using a standard deviation σ randomly picked from a uniform distribution $U(\Sigma(\alpha_i)/2k, \Sigma(\alpha_i)/k)$ in order to provide a high noise to signal ratio and to make the identification of seasonality sufficiently challenging. The parameters used for the construction of the STS and the identified periodicities of the seasonalities are summarised in Table A1 (Note that Time Series A.1 is equivalent to series A and B.1 to series B in Section 3.2.)

References

[1] G.Q. Zhang, B.E. Patuwo, M.Y. Hu, Forecasting with artificial neural networks: the state of the art, International Journal of Forecasting 14 (1) (1998) 35–62.

[2] K. Hornik, Approximation capabilities of multilayer feedforward networks, Neural Networks 4 (2) (1991) 251–257.

[3] S. Makridakis, M. Hibon, The M3-competition: results, conclusions and implications, International Journal of Forecasting 16 (4) (2000) 451–476.

[4] T. Hill, M. O'Connor, W. Remus, Neural network models for time series forecasts, Management Science 42 (7) (1996) 1082–1092.

[5] G.P. Zhang, M. Qi, Neural network forecasting for seasonal and trend time series, European Journal of Operational Research 160 (2) (2005) 501–514.

[6] J.S. Armstrong, Findings from evidence-based forecasting: methods for reducing forecast error, International Journal of Forecasting 22 (3) (2006) 583–598.

[7] U. Anders, O. Korn, Model selection in neural networks, Neural Networks 12 (2) (1999) 309–323.

[8] J.W. Taylor, L.M. de Menezes, P.E. McSharry, A comparison of univariate methods for forecasting electricity demand up to a day ahead, International Journal of Forecasting 22 (1) (2006) 1–16.

[9] B. Curry, P.H. Morgan, Model selection in neural networks: some difficulties, European Journal of Operational Research 170 (2) (2006) 567–577.

[10] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 1995.

[11] S.S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd ed., Prentice-Hall, Upper Saddle River, NJ, 1999.

[12] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, et al., Time Series Analysis: Forecasting and Control, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ [u.a.], 1994.

[13] G. Dorffner, Neural networks for time series processing, Neural Network World 6 (4) (1996) 447–468.

[14] A. Lapedes, R. Farber, How neural nets work, in: D.Z. Anderson (Ed.), Neural Information Processing Systems, American Institute of Physics, New York, 1988, pp. 442–456.

[15] U. Anders, O. Korn, C. Schmitt, Improving the pricing of options: a neural network approach, Journal of Forecasting 17 (5–6) (1998) 369–388.

[16] G. Lachtermacher, J.D. Fuller, Backpropagation in time-series forecasting, Journal of Forecasting 14 (4) (1995) 381–393.

[17] D.W. Bunn, Non-traditional methods of forecasting, European Journal of Operational Research 92 (3) (1996) 528–536.

[18] K.P. Liao, R. Fildes, The accuracy of a procedural approach to specifying feedforward neural networks for forecasting, Computers & Operations Research 32 (8) (2005) 2151–2169.

[19] M. Adya, F. Collopy, How effective are neural networks at forecasting and prediction? A review and evaluation, Journal of Forecasting 17 (5–6) (1998) 481–495.

[20] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1–2) (1997) 273–324.
[21] S.D. Balkin, J.K. Ord, Automatic neural network modeling for univariate time series, International Journal of Forecasting 16 (4) (2000) 509–515.
[22] E. Liitiainen, A. Lendasse, Long-term prediction of time series using state-space models, Artificial Neural Networks – Icann 2006, Pt 2 4132 (2006) 181–190.
[23] Y. Miche, A. Sorjamaa, A. Lendasse, OP-ELM: theory, experiments and a toolbox, Artificial Neural Networks – Icann 2008, Pt I 5163 (2008) 145–154.
[24] A. Sorjamaa, J. Hao, N. Reyhani, et al., Methodology for long-term prediction of time series, Neurocomputing 70 (16–18) (2007) 2861–2869.
[25] A. Sorjamaa, Y. Miche, R. Weiss, et al., Long-term prediction of time series using NNE-based projection and OP-ELM, in: 2008 IEEE International Joint Conference on Neural Networks, vols. 1–8, 2008, pp. 2674–2680.
[26] A.P.N. Refenes, A.D. Zapranis, Neural model identification, variable selection and model adequacy, Journal of Forecasting 18 (5) (1999) 299–332.
[27] M.C. Medeiros, T. Terasvirta, G. Rech, Building neural network models for time series: a statistical approach, Journal of Forecasting 25 (1) (2006) 49–75.
[28] J.Y. Yang, S. Olafsson, Optimization-based feature selection with adaptive instance sampling, Computers & Operations Research 33 (11) (2006) 3088–3106.
[29] S. Piramuthu, Evaluating feature selection methods for learning in data mining applications, European Journal of Operational Research 156 (2) (2004) 483–494.
[30] S. Viaene, B. Baesens, D. Van den Poel, et al., Wrapped input selection using multilayer perceptrons for repeat-purchase modeling in direct marketing, International Journal of Intelligent Systems in Accounting, Finance & Management 10 (2) (2001) 115–126.
[31] Y.S. Kim, W.N. Street, G.J. Russell, et al., Customer targeting: a neural network approach guided by genetic algorithms, Management Science 51 (2005) 264.
[32] G.E.P. Box, G.M. Jenkins, Time Series Analysis: Forecasting and Control, Holden-Day, San Francisco, 1970.
[33] D. Reilly, The AUTOBOX system, International Journal of Forecasting 16 (4) (2000) 531–533.
[34] R.L. Goodrich, The Forecast Pro methodology, International Journal of Forecasting 16 (4) (2000) 533–535.
[35] G. Zhang, B.E. Patuwo, M.Y. Hu, Forecasting with artificial neural networks: the state of the art, International Journal of Forecasting 14 (1) (1998) 35–62.
[36] G. Lachtermacher, J.D. Fuller, Backpropagation in time-series forecasting, Journal of Forecasting 14 (4) (1995) 381.
[37] Z.Y. Tang, P.A. Fishwick, Feed-forward neural nets as models for time series forecasting, ORSA Journal on Computing 5 (4) (1993) 374–386.
[38] S. Makridakis, S.C. Wheelwright, R.J. Hyndman, Forecasting: Methods and Applications, 3rd ed., John Wiley & Sons Inc., New York, 1998, p. 642.
[39] D.J. Hand, Data mining: statistics and more? American Statistician 52 (2) (1998) 112–118.
[40] N.R. Swanson, H. White, Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models, International Journal of Forecasting 13 (4) (1997) 439–461.
[41] M. Qi, G.S. Maddala, Economic factors and the stock market: a new perspective, Journal of Forecasting 18 (3) (1999) 151–166.
[42] C.M. Dahl, S. Hylleberg, Flexible regression models and relative forecast performance, International Journal of Forecasting 20 (2) (2004) 201–217.
[43] S.M. Kay, S.L. Marple, Spectrum analysis – a modern perspective, Proceedings of the IEEE 69 (11) (1981) 1380–1419.
[44] G.E.P. Box, G.M. Jenkins, G.C. Reinsel, Time Series Analysis: Forecasting and Control, 3rd ed., Prentice-Hall Inc., New Jersey, 1994.
[45] E. Ghysels, D.R. Osborn, The Econometric Analysis of Seasonal Time Series, Cambridge University Press, Cambridge, 2001.
[46] S. Crone, N. Kourentzes, Forecasting seasonal time series with multilayer perceptrons – an empirical evaluation of input vector specifications for deterministic seasonality, pp. 232–238.
[47] R. Neuneier, H.-G. Zimmermann, How to train neural networks, in: G. Orr, K.-R. Müller (Eds.), Neural Networks: Tricks of the Trade, Springer, Berlin, New York, 1998, pp. 373–423.
[48] M. Nelson, T. Hill, W. Remus, et al., Time series forecasting using neural networks: should the data be deseasonalized first? Journal of Forecasting 18 (5) (1999) 359–367.
[49] L. Zhou, F. Collopy, M. Kennedy, The Problem of Neural Networks in Business Forecasting – An Attempt to Reproduc th Hill, O'Connor and Remus Study, Case Wetsern Reserve University, Cleveland, 2003.
[50] S.F. Crone, J. Guajardo, R. Weber, The impact of Data Preprocessing on Support Vector Regression and Artificial Neural Networks in Time Series Forecasting.
[51] S.F. Crone, J. Guajardo, R. Weber, A study on the ability of Support Vector Regression and Neural Networks to Forecast Basic Time Series Patterns.
[52] S.F. Crone, S. Lessmann, S. Pietsch, An empirical Evaluation of Support Vector Regression versus Artificial Neural Networks to Forecast basic Time Series Patterns.
[53] C.W.J. Granger, Extracting information from mega-panels and high-frequency data, Statistica Neerlandica 52 (3) (1998) 258–272.
[54] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Networks 2 (5) (1989) 359–366.
[55] S.F. Crone, 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.
[56] S. Heravi, D.R. Osborn, C.R. Birchenhall, Linear versus neural network forecasts for European industrial production series, International Journal of Forecasting 20 (3) (2004) 435–446.
[57] M. Adya, F. Collopy, J.S. Armstrong, et al., Automatic identification of time series features for rule-based forecasting, International Journal of Forecasting 17 (2) (2001) 143–157.
[58] R. Fildes, The evaluation of extrapolative forecasting methods, International Journal of Forecasting 8 (1) (1992) 81–98.
[59] L.J. Tashman, Out-of-sample tests of forecasting accuracy: an analysis and review, International Journal of Forecasting 16 (4) (2000) 437–450.
[60] L. Tashman, Out-of-sample tests of forecasting accuracy: an analysis and review, International Journal of Forecasting 16 (2000) 437–450.
[61] J.S. Armstrong, F. Collopy, Error measures for generalizing about forecasting methods: empirical comparisons, International Journal of Forecasting 8 (1) (1992) 69–80.
[62] R. Fildes, S. Makridakis, Forecasting and loss functions, International Journal of Forecasting 4 (4) (1988) 545–550.
[63] E.S. Gardner, Exponential smoothing: the state of the art – Part II, International Journal of Forecasting 22 (4) (2006) 637–666.

Sven F. Crone is an Assistant Professor of Management Science at Lancaster University Management School and the deputy director of the Lancaster Research Centre for Forecasting. He received his Diplom-Kaufmann (BBA and MBA equivalent) and Ph.D. from Hamburg University, Germany, with research fellowships in South Africa and the USA. His research focuses on forecasting, time series prediction and data mining in business applications, frequently employing methods from Computational Intelligence such as neural networks and support vector machines. His research has been published in the European Journal of Operational Research, Journal of Operational Research Society and International Journal of Forecasting. Sven is the competition chair of the IEEE CIS Data Mining Technical Committee and has organised the 2007 Neural Network Forecasting Competition (NN3) co-sponsored by the IIF, NSF and SAS, the 2008 NN5 and the current 2009 IEEE Grand Challenge on Time Series Prediction with Computational Intelligence.

Nikolaos Kourentzes is a post-doctorate research assistant in Management Science at Lancaster University Management School. He received his BBA from Athens University of Economics and Business (AUEB), an M.Sc. in Management Science and a Ph.D. from Lancaster University Management School. His research focuses on time series prediction with neural networks, with a particular emphasis on input vector specification and high frequency data.