

# Frequency independent automatic input variable selection for neural networks for forecasting

Nikolaos Kourentzes and Sven F. Crone

**Abstract**—Key issue in time series forecasting with Neural Networks (NN) is the selection of the relevant input variables, which is often the result of data exploration by human experts, leading to dataset specific solutions and limiting forecasting automation. This becomes even more important in heterogeneous datasets, where each time series requires special modeling and can exhibit a different variety of stochastic and deterministic components of different unknown frequencies. Fully automated forecasting with NNs requires a methodology that can address these issues in an entirely data driven approach. This paper proposes a fully automated input selection methodology based on a novel iterative NN filter that automatically identifies for each time series the seasonal frequencies, if such are present, the dynamic structure of the time series, distinguishing between stochastic and deterministic components, ultimately producing a parsimonious set of input variables. The robustness and performance of the algorithm are evaluated against established time series forecasting methods.

## I. INTRODUCTION

OVER the last decades there has been an increase in the research of Artificial Neural Networks (NN) to forecasting problems. Both in theoretical and empirical works, NNs have shown evidence of good performance, in many cases outperforming established benchmarks [1]. NNs are non-parametric data driven models that in theory can approximate any linear and nonlinear data generating processes [2]. However, in practice, despite their theoretical capabilities, NNs have not been able to demonstrate robust performance and consistently good forecasting accuracy against established statistical forecasting methods, like exponential smoothing and ARIMA based models [3]. Furthermore, in the forecasting literature NNs have been shown to produce inconsistent results, resulting in limited practical applications, especially in the business forecasting and supply chain contexts [4, 5]. This can be attributed to the modeling complexity of such models [1]. There are numerous parameters that the human expert has to fine tune in order to produce accurate forecasts. Setting these parameters has proven to be challenging, resulting in several alternative model specification methodologies and NN variants [1, 6]. Although no specification methodology has been widely accepted [7, 8], it is well established that the most important determinant of forecasting accuracy for NNs

is the selection of the input variables [9, 10]. The importance of input variable and lag selection is evident, as the input vector needs to capture all characteristics of complex time series, including 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. Moreover, there is strong evidence that the ability of NNs to forecast trended and seasonal time series is connected with the correct identification and coding of the relevant inputs [11, 12], although it is still debatable how this should be done [13, 14]. As a result, the valid and reliable specification of NNs for time series forecasting is often considered as much an art as science, limiting automatic large-scale implementations in forecasting problems.

Automatic forecasts of large numbers of univariate time series are often needed in business and other contexts. Automatic model specification for forecasting requires a systematic time series analysis procedure that can scale-up [15, 16]. Time series should be analyzed for their components, identifying the presence of trend, cycles and seasonal patterns, while filtering irregularities, such as outliers, level shifts, etc [17]. Despite the existence of several semi- and fully-automatic approaches for statistical methods these normally assume a given and known underlying frequency in the dataset, which is typically reflected in the observed seasonality of the time series and is assumed to remain constant across time series of the same dataset. These are significant shortcomings of these approaches, since the human element is still required and heterogeneous dataset that include time series of varying frequencies cannot be forecasted automatically. In practice, it is not uncommon to need both short- and long-term forecasts from low- and high- frequency time series for strategic and operational decision support, as is the case for example in production and inventory decision making [18, 19]. Therefore, fully automated forecasting systems that are able to address these issues, even with well established statistical forecasting methods, do not exist.

Furthermore, many forecasting methods are not readily applicable to a wide range of frequencies, especially when multiple overlaying periodicities appear, as it is common in high frequency data [20]. Typically, in particular for high frequency applications, the functional form of statistical forecasting methods has to be altered to address the specific properties of the frequency of the time series at hand [21].

Manuscript received January 7, 2010. Nikolaos Kourentzes (corresponding author) and Sven F. Crone are with the department of Management Science at Lancaster University Management School, Lancaster, LA1 4YX, United Kingdom. (phone: +44.1524.5-92991; fax: +44.1524.844885; e-mail: {n.kourentzes; s.crone}@lancaster.ac.uk).

This makes it impossible to forecast datasets of unknown frequencies and challenging to produce fully automated forecasting systems. However, NNs are flexible models that have been shown to perform well in forecasting a large variety of different frequencies, requiring only minimal changes to their input vector [22, 23].

Based on this attribute of NNs, we have proposed an input variable selection methodology for forecasting with NNs that is applicable to datasets of varying seasonality, enabling the construction of fully automated forecasting systems. The proposed methodology combines a novel Iterative Neural-Filter (INF), which is based on multilayer perceptrons, with statistical methods, in order to produce a set of input variables for automated forecasting with NNs. The INF can automatically identify frequencies that are present in a time series and an initial set of input variables that are fed to statistical methods that are able to robustly analyze the dynamic structure and nature of time series, once the relevant frequencies are known; thus, performing automatically feature evaluation and time series identification. The proposed methodology is able to capture the structure of time series, and to automatically identify the periodicity of the time series, even in the presence of multiple overlaying seasonalities. It is fully data driven and should be able to capture any unknown data generating process without the need for domain knowledge and human intervention.

We have applied a prototype of the INF methodology successfully in forecasting the ESTSP 08' (European Symposium on Time Series Prediction) dataset that includes time series of varying seasonality [22], demonstrating that fully data driven NN modeling is possible. That prototype had several limitations and in this paper we present a complete input selection methodology based on the INF. Forecasting accuracy is improved substantially, by identifying and distinguishing between deterministic and stochastic elements of the time series, coding the inputs accordingly and more economically; improving performance and computational efficiency by capturing the time series structure in parsimonious input vectors with minimal information redundancy. A key advantage of this methodology is that it avoids using wrappers. Consequently, it readily scales-up for large-scale applications. The novel input variable selection methodology is independent of time series frequency, allowing for automated forecasting with NNs in heterogeneous datasets, thus addressing both the problems of input variable selection for forecasting with NNs and automated forecasting for datasets of varying frequency.

We provide evidence of good performance using a Monte Carlo simulation of synthetic time series of varying frequencies, exhibiting both stochastic and deterministic seasonalities and varying levels of Gaussian noise, replicating real business time series and real monthly data. We find consistent performance of the methodology across different frequencies, levels of noise and nature of

seasonality. Furthermore, the performance of the proposed methodology is benchmarked against exponential smoothing, a well established statistical method [3, 24], and found to be superior.

In section II, we introduce NNs in the context of time series forecasting and presents the proposed INF methodology. Section III describes the experimental setup of the forecasting accuracy evaluation simulations and section IV provides the results, followed by concluding remarks and future work in section V.

## II. METHODS

### A. Neural Networks for Time Series Forecasting

NNs have been successfully applied in both univariate and multivariate time series forecasting. The most widely employed architecture is the common multilayer perceptron (MLP). These well researched regarding their properties and have been shown to be able to generalize any linear or nonlinear functional relationship to any degree of accuracy without any prior assumptions about the underlying data generating process [1, 25]. In this study we will use MLPs to produce the forecasts, though it is possible to use other types of NNs in conjunction with the proposed INF input variable selection methodology.

In forecasting feed-forward architectures of MLPs are used to model nonlinear autoregressive NAR( $p$ )-processes or NARX( $p$ )-processes using external variables to code exogenous events as intervention variables. Given a time series  $y$ , at a point in time  $t$ , a one-step ahead forecast  $\hat{y}_{t+1}$  is computed using  $p=I$  observations  $y_t, y_{t-1}, \dots, y_{t-I+1}$  from  $I$  preceding points in time  $t, t-1, t-2, \dots, t-I+1$ , with  $I$  denoting the number of input units of the NN. The functional forms is

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

where  $Y = [y_t \dots y_{t-I+1}]$  is the vector of the lagged observations (inputs) of the time series. The network weights are  $w = (\beta, \gamma)$ ,  $\beta = [\beta_1, \beta_2 \dots \beta_H]$  and  $\gamma = [\gamma_{11}, \gamma_{12} \dots \gamma_{HI}]$  for the output and the hidden layer respectively. The  $\beta_0$  and  $\gamma_{0i}$  are the biases of each neuron.  $I$  and  $H$  are the number of input and hidden units in the network and  $g(\cdot)$  is a non-linear transfer function [26], which is usually either the sigmoid tangent or the hyperbolic tangent function [1]. MLPs offer extensive degrees of freedom in modeling for prediction tasks. The modeler must choose the appropriate data and its pre-processing, the NN architecture, the signal processing within nodes, the training algorithm and the cost function. For a detailed discussion of these issues and the ability of NNs to forecast time series, the reader is referred to [1]. The specification of the input vector has been identified as being particularly crucial to achieving valid and reliable forecasts and examined in more detail in the following section.

### B. Automatic input variable selection using the INF

The identification of input variables and variable lags aims to capture the relevant components of the data generating process in a parsimonious form. In time series modeling, this involves the identification of the underlying time series components of trend and seasonality and capturing their behavior in lags of the dependent variable. Several methodologies have been suggested for input variables selection for NNs in the forecasting literature, some of which are better suited than others to heterogeneous datasets [6]. However, all methodologies assume that the underlying time series frequency is known.

Fully automatic input variable selection, which assumes no domain knowledge of the forecasting problem at hand, requires a data driven methodology to identify the frequencies that exist in the time series. Once the frequency of the time series is known then possible trend and seasonality have to be identified and coded in the input vector. There has been considerable debate on how to best perform trend and seasonal modeling for time series forecasting [11, 13, 14, 27, 28]. It has been shown that correctly identifying and coding these components as deterministic or stochastic has a significant impact on forecasting accuracy [12, 29, 30]. Furthermore, the dynamic lag structure of the time series needs to be identified. In this study we propose a framework that address all these issues in a fully data driven methodology.

First we will define the proposed INF, which allows us to perform automatic frequency identification and initial feature selection. Any given time series  $Y$  of length  $N$  can be split in  $N/s$  vectors, where  $s = [1, 2 \dots N/2]$ . Each vector exhibits some distinct pattern, which is different for each value of  $s$ . Furthermore, for different value of  $s$  the time series is split in a different number of vectors, with the extremes of  $s = 1$  that the time series is split in  $N$  vectors, each containing a single observation and  $s = N/2$  that the time series is split in 2 vectors each containing  $N/2$  observations. When  $s$  is the same as  $S$ , the true seasonal length of the time series, all the vector will exhibit the same pattern with some deviations due to noise; hence the distance between them will be minimised. We can measure the distance between these vectors using the Euclidean distance. Euclidean distance measures the distance between two points. For the one-dimensional case, for two points  $P = p$  and  $Q = q$  the Euclidian distance is defined as

$$f(P, Q) = \sqrt{(p - q)^2}. \quad (2)$$

For the case of one-dimensional vectors with equal length  $n$ , the distance is the sum of the pairwise distances of the  $n$  pairs. If the vectors are more than two, then for the pairwise distance calculation all the possible combinations must be considered. To illustrate this, for three vectors  $\mathbf{P} = [p_1, p_2 \dots p_n]$ ,  $\mathbf{Q} = [q_1, q_2 \dots q_n]$  and  $\mathbf{R} = [r_1, r_2 \dots r_n]$  for  $t = 1$  there are three different pairs that the distance is measured:  $(p_1,$

$q_1)$ ,  $(p_1, r_1)$  and  $(q_1, r_1)$ . Generalising, for  $n$  number of vectors there are  $n(n-1)/2$  combinations. Dividing the sum of distance of all the pairs by the number of pairs provides us with an average distance per pair for each given  $s$ . This measurement is independent of the number of vectors, or their sizes, in which we divide the original time series. This way we can compare distances calculated for any  $s$  directly. The  $s$  that minimises the distance will indicate a possible seasonal length; note that if  $s = 1$  then there is no seasonality.

If no noise is present in the time series then  $S$  and all multiples of it will exhibit the same minimum distance, therefore making it hard to distinguish the correct  $S$  from its multiples. In order to overcome this problem, a penalty is applied to longer  $s$ . The penalised mean distance  $D_{ps}$  is expressed as

$$D_{ps}(s) = \log(D_s + 1) - \tau \log(s), \quad (3)$$

where  $D_s$  is the mean distance for a given  $s$  and  $\tau$  is a scaling factor. This penalty ensures that no multiples of  $S$  will be identified instead of the true  $S$ , as they will always be penalised by  $\tau \log(\alpha S)$ , with  $\alpha$  being the number that  $S$  is multiplied. Penalty  $\tau$  controls the sensitivity of  $D_{ps}$  to noise (we have empirically found  $\tau = 0.15$  to be adequate for most datasets). The minimum penalised distance identifies the shortest seasonality of the time series. In order to account for multiple independent or interacting seasonal frequencies, the prominent seasonality that minimises  $D_{ps}$ , has to be filtered from the time series, thus allowing the identification of less dominant frequencies.

To do this, we propose an iterative neural filter (INF), capable of removing any type of non-linear seasonality in the presence of other seasonalities, trends and irregularities in the time series. This filter takes advantage of the properties of NNs, which have been shown to be able to approximate any linear or nonlinear patterns [1], model any seasonality in a time series regardless of the frequency by altering only the inputs of the network [22, 23] and model seasonal signals very robustly and parsimonious [12]. This allows us to avoid a strict functional form of the filter; hence be applicable to more cases. We use a special MLP to implement the INF that uses as inputs only contemporaneous explanatory variables that encode time series patterns. The objective of this MLP is not to forecast, but only to capture the observed seasonalities in the time series. Therefore, no test set is used during training, withholding  $S$  observations for validation purposes and using all remaining  $N-S$  observations of the time series for training. Note that  $S$  is estimated in the previous phase and reflects where  $s = D_s$ . Initially the INF has four inputs:

$$\psi_1(t) = \sin\left(\frac{2\pi t}{S}\right), \quad (4)$$

$$\psi_2(t) = \cos\left(\frac{2\pi t}{S}\right), \quad (5)$$

$$\psi_3(t) = t, \quad (6)$$

$$\psi_4(t) = N - t + 1. \quad (7)$$

For all (4) - (7)  $t = 1 \dots N$ . The first two inputs (4) and (5) code signals of any given period  $N$ . In contrast to the  $s-1$  binary dummies conventionally used in regression to encode deterministic seasonality, the explanatory variables  $\psi_1$  and  $\psi_2$  code seasonality as a single pair of sine-cosine waves, letting the NN do the detailed mapping, thus substantially reducing the size of the input vector, without any loss in accuracy [12]. This becomes especially useful for modeling very long and multiple seasonalities. The remaining inputs (6) and (7) have a dual purpose. They provide an explicit representation of the point in time  $t$  within the time series facilitating 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. Furthermore, they are used to model changes in the magnitude of the periodic signal over time helping the MLP to capture multiplicative forms of periodic signals. All contemporaneous inputs are linearly scaled between  $[-1, 1]$ .

Once the MLP is trained, the network output, which expresses the regular structure of the time series as captured by the MLP, is subtracted from the original time series  $Y$ , 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 the following iterations, seasonal components of different length  $s$  to the one identified before are added as additional pairs of sine-cosine inputs. The process is repeated until the most prominent period, as identified by  $D_{ps}$  is  $s = 1$ , which implies that no further seasonality is present in the time series. At this point the inputs  $\psi_3$  and  $\psi_4$  are dropped and all the remaining identified pairs of sine-cosine waves (that encode the frequencies that are present in the time series) are used as preliminary inputs in the phase of the proposed methodology. Figure 1 illustrates the steps of INF. Although the MLP in the heart of INF does not have to follow a specific architecture, we empirically found that a common single hidden layer MLP with hyperbolic tangent activation function<sup>1</sup> and a linear output node performs well under most circumstances. Regarding the training algorithm and setting of the NN, there are is no special consideration and standard training is used. The reader can refer to [1] for more details about MLP training. Note that the network should be initialized several times to make it robust to the stochastic nature of its training.

Once the number of seasonal patterns and their respective frequencies are identified then it is necessary to determine whether they are deterministic or stochastic, as this evidently requires different coding of the inputs for forecasting with NNs [12]. Obviously, if no seasonalities are identified this step is skipped. To do this a simple heuristic based on

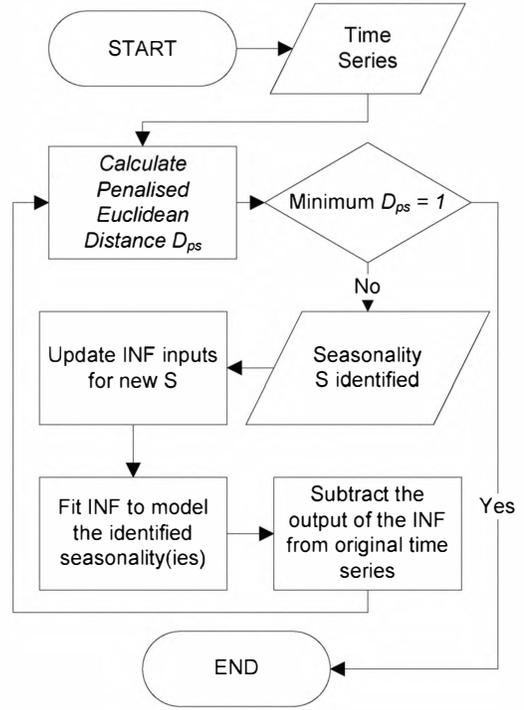


Fig. 1. Flow chart of the proposed INF methodology. The outputs are the identified frequencies and an initial set of input variables expressed through pairs of sines-cosines.

regression models is employed. Initially, two competing models formulations, a deterministic and a stochastic one, are constructed. For time series  $Y$ , given  $j = 1, \dots, N_s$  different identified seasonalities  $S_j$  of length  $i$ , the deterministic and the stochastic models respectively are:

$$\hat{Y}^D_t = a + M + \sum_{j=1}^{N_s} \sum_{i=1}^{S_j} b_{ji} d_{jit} + \varepsilon_t, \quad (8)$$

$$\hat{Y}^S_t = a + \sum_{j=1}^{N_s} b_j Y_{t-S_j} + \varepsilon_t. \quad (9)$$

Both deterministic  $\hat{Y}^D$  and stochastic  $\hat{Y}^S$  models include a constant  $a$ . For the deterministic model a centered moving average  $M$  of length equal to  $\max(S_j)$  is used to aid in the estimation of the coefficients  $b_{ji}$  for the seasonal dummies  $d_{jit}$ . Note that (8) can be simplified by replacing  $M$ ,  $b_{ji}$  and  $d_{jit}$  with  $N_s$  vectors of seasonal indices, one for each seasonality. The stochastic formulation is a dynamic regression, where the inputs are  $N_s$  seasonal lags of the time series  $Y$ , one for each identified frequency, multiplied by the respective coefficients  $b_j$ . The models are fitted to the original time series  $Y$  by minimizing the square of residuals  $e$  and the Akaike Information Criterion (AIC) of both models is calculated. The model that performs best indicates whether the periodic components are deterministic or stochastic. Note that this simple test essentially follows the ideas of formal statistical tests that are used in distinguishing between deterministic and stochastic seasonality, like the Canova-

<sup>1</sup> We used 16 hidden nodes for all experiments. The performance of the filter proved to be very robust to the number of nodes.

Hansen test [30, 31]. However, in this study such tests are not used because they have not been designed to cope with either multiple or long seasonalities.

The next step in the methodology is to identify the dynamic structure of the time series. In the forecasting literature there are several alternatives on how to select the relevant time lags of time series for univariate forecasting [7]. We have opted to avoid using wrapper methodologies to retain the scalability of the algorithm; therefore we consider only filter approaches. The most common are based on the analysis of the (partial-) autocorrelation function, ARIMA modeling analogies, stepwise regression and their nonlinear extensions. In an evaluation of these methodologies stepwise regression was found to be consistently the most robust and accurate on different time series patterns and frequencies [6]. Therefore, here we employ stepwise regression to identify the dynamic structure of the time series. However, we take advantage of the preliminary input identification done with the INF by priming the regression with these inputs. We define  $X_{INF}$  a matrix which has a columns the outputs of the INF as transformed in (8) or (9), depending which model showed better AIC. In the case of the deterministic model formulation, vectors of seasonal indices are preferred to dummies. This is done in order to limit the variables and speed up the calculation of the stepwise regression. Furthermore, we define the matrix  $X_{LAG}$  that contains the lags of time series  $Y$  for lags  $t-1$  to  $t-n$ , where  $n$  is the maximum lag considered and a constant column. Using these two matrices we run a stepwise regression model, forcing the variables described by the columns of  $X_{INF}$  to be initially included in the stepwise model. These variables are expected to be significant, as identified by the INF, and therefore substantially limit the number of stepwise iterations required until the primed regression converges. Furthermore, this formulation limits the inclusion of redundant information, as the primed variables do not allow collinear variables to enter the regression, essentially decreasing their  $F_{to\_enter}$  statistic. For additional info on stepwise regression see [32, 33].

The output of the previous step is a set of lagged variables (seasonal or not) and deterministic seasonal vectors, which were identified based on the frequencies outputted by the INF. These are consequently used as inputs to NN models to forecast time series of any unknown frequency and structure. Note, that a key advantage of using the INF and later NNs to produce the forecast is their ability to accurately represent seasonal information with very few variables, while at the same time being able to model additive or multiplicative, continuous or discontinuous seasonal patterns with the same inputs [6]. This greatly extends the flexibility of the proposed methodology to address unknown time series. An overview of the complete input variable selection methodology is illustrated in figure 2.

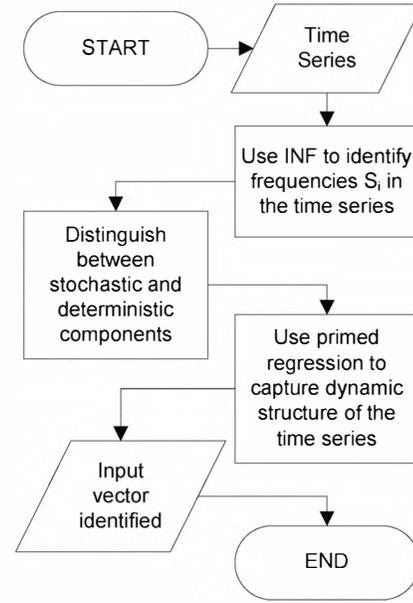


Fig. 2. Flow chart of the proposed automatic input variable selection methodology.

### III. EXPERIMENTAL DESIGN

#### A. Time Series

In order to evaluate the performance of the proposed methodology we construct a synthetic dataset that contains a mixture of stochastic and deterministic seasonality time series, with different frequencies and different levels of noise. Time series are constructed using two basic equations, one for the stochastic (10) and one for the deterministic case (11).

$$\varphi(L)\Delta_S y_t = \theta(L)z_t, \quad (10)$$

$$y_t = \mu + \sum_{j=1}^{N_s} \sum_{i=1}^{S_j} b_{ji} d_{ji} + z_t. \quad (11)$$

For (10)  $L$  is the lag operator,  $\Delta_S$  is the seasonal difference operator,  $\varphi$  and  $\theta$  are the coefficients of the autoregressive and moving average process respectively. For the deterministic case,  $\mu$  is the level of the time series and  $b_{ji}$  the coefficients for the seasonal dummies  $d_{ji}$ , while  $S_j$  is the periodicity of seasonality  $j = 1, \dots, N_s$ . For both equations  $z_t$  follows  $N(0, \sigma^2)$ . We simulate four different levels of noise, no noise ( $\sigma = 0$ ), low noise ( $\sigma = 5$ ), medium noise ( $\sigma = 10$ ) and high noise ( $\sigma = 15$ ). All time series have a level  $\mu = 100$ . A set of different frequencies are used to simulated quarterly data ( $S = 4$ ) and monthly data ( $S = 12$ ). For each frequency and noise level we produce 30 time series, resulting in 240 stochastic and 240 deterministic time series. We also create a small subset of double seasonal daily data ( $S = 7, 365$ ), using only 5 time series per noise level, resulting in 40 time series in total. In total 520 time series are simulated. The sample sizes of the quarterly, monthly and daily time series are 600, 900 and 1825 observations respectively, allowing several

seasonal cycles to be observed. The time series are then split in 3 equal sized subsets for training, validation and out-of-sample testing.

Additionally five real time series are used. These are all monthly time series corresponding to US air passenger miles, average bus ridership for Portland Oregon, total number of room nights and takings in Victoria and number of serious injuries and deaths in UK road accidents. These time series are 216, 114, 186, 186 and 192 months long respectively and can be found at [34]. The validation and the test sets are set to 24 months.

### B. Experimental setup

The forecasting horizon for quarterly, monthly and daily synthetic time series is set to 8, 24 and 50 observations respectively, while for the real data this is set to 12 months. To assess the forecasting performance of the NNs the symmetric mean absolute percent error (sMAPE) is used. This has the advantage of being scale independent, allowing to aggregate results over time series and is less biased than the commonly used mean absolute percentage error (MAPE). It computes the absolute error in percent between the actuals  $X_t$  and the forecast  $F_t$  for all periods  $t$  of the test set of size  $n=h$  for each time origin:

$$SMAPE = \frac{1}{n} \sum_{t=1}^n \left( \frac{|X_t - F_t|}{(|X_t| + |F_t|)/2} \right) 100. \quad (12)$$

All models are evaluated using a rolling time origin evaluation, producing multiple forecasts for each time series, resulting in more accurate estimation of the forecasting error. Furthermore, the error estimation is robust against irregular origins [35]. The accuracy of the competing NN models is evaluated for statistically significant differences (at 5% significance level) using the nonparametric Friedman test and Nemenyi tests, to facilitate an evaluation of nonparametric models without the need to relax assumptions of ANOVA or similar parametric tests [36].

### C. Methods

A single MLP setup is used to forecast all time series. The only variable part of the MLP is the input vector which is the result of the INF input variable selection algorithm. The neural network, uses a single hidden layer with 6 nodes and hyperbolic tangent activation functions. The networks are trained using the Levenberg-Marquardt algorithm, which requires setting the  $\mu_{LM}$  and its increase and decrease steps. Here  $\mu_{LM}=10^{-3}$ , with an increase step of  $\mu_{inc}=10$  and a decrease step of  $\mu_{dec}=10^{-1}$ . For a detailed description of the algorithm and the parameters see [37]. The maximum training epochs are set to 1000. The training can stop earlier if  $\mu_{LM}$  becomes equal of greater than  $\mu_{max}=10^{10}$  or the validation error increases for more than 50 epochs. This is done to avoid over-fitting. When the training is stopped the network weights that give the lowest validation error are

TABLE I  
SMAPE ACCURACY

		Deterministic			
	Noise	Primed NN	Normal NN	Stoch NN	EXSM
Train	No	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>
	Low	4.37%	<u>4.29%</u>	4.61%	4.70%
	Medium	8.78%	<u>8.65%</u>	9.25%	9.39%
	High	13.43%	<u>13.25%</u>	14.32%	14.39%
Valid	No	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>
	Low	<u>4.40%</u>	4.43%	4.76%	4.60%
	Medium	<u>8.84%</u>	8.95%	9.61%	9.25%
	High	<u>13.32%</u>	13.43%	14.68%	13.96%
Test	No	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>
	Low	<u>4.47%</u>	4.51%	4.86%	4.55%
	Medium	<u>8.90%</u>	9.02%	9.77%	9.13%
	High	<u>13.82%</u>	14.02%	15.18%	14.14%
		Stochastic			
	Noise	Primed NN	Normal NN	Stoch NN	EXSM
Train	No	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>
	Low	4.69%	<u>4.64%</u>	4.69%	4.85%
	Medium	9.78%	<u>9.61%</u>	9.76%	10.08%
	High	<u>16.99%</u>	17.10%	<u>16.99%</u>	18.06%
Valid	No	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>
	Low	<u>4.67%</u>	<u>4.67%</u>	<u>4.67%</u>	4.86%
	Medium	<u>9.80%</u>	<u>9.80%</u>	9.81%	10.22%
	High	16.25%	<u>16.20%</u>	16.25%	17.30%
Test	No	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>	<u>0.00%</u>
	Low	5.05%	5.19%	5.05%	<u>4.95%</u>
	Medium	<u>10.24%</u>	10.74%	<u>10.24%</u>	10.25%
	High	<u>16.52%</u>	16.98%	<u>16.52%</u>	16.76%
		Overall			
Subset	Primed NN	Normal NN	Stoch NN	EXSM	
Train	7.25%	<u>7.19%</u>	7.45%	7.68%	
Valid	<u>7.16%</u>	7.18%	7.47%	7.52%	
Test	<u>7.37%</u>	7.56%	7.70%	7.47%	

Results are provided by noise level and nature of the time series components for each training, validation and test subsets. The best performing model in each row is underlined.

TABLE III  
SMAPE ACCURACY

		Real Data			
Subset	Primed NN	Normal NN	Stoch NN	EXSM	
Train	8.52%	8.74%	7.86%	<u>7.82%</u>	
Valid	<u>5.06%</u>	6.26%	5.91%	6.83%	
Test	<u>7.86%</u>	11.67%	11.95%	10.72%	

The best performing model in each row is underlined.

used. In all experiments the 1000 epochs training limit was never exceeded due to the early stopping criterion. Each MLP is initialised 40 times with randomised starting weights to accommodate the nonlinear optimisation and to provide an adequate sample to estimate the distribution of the forecast errors in order to conduct the statistical tests. The MLP initialisation with the lowest error for each time series on the validation dataset is selected to predict all values of the test set. Lastly, the time series and all explanatory variables that are not binary are linearly scaled between  $[-0.5, 0.5]$ .

To illustrate the advantage of priming the regression (the model is named *prime\_NN*) as discussed in section II, two alternative inputs methodologies are considered; the first, *normal\_NN* uses a normal stepwise regression. Also to illustrate the importance of selecting between stochastic and deterministic components a third model uses only stochastic

inputs (*stoch\_NN*). The latter is the basis of the proposed specification methodologies in several papers (for e.g. see [38, 39]).

Any empirical evaluation of time series methods requires the comparison of their accuracy with established statistical benchmark methods, in order to assess the increase in accuracy and its contribution to forecasting research. This is often overlooked in NN experiments [40]. In this analysis seasonal exponential smoothing models (*EXSM*) are used. Note that it is impossible to apply these models fully automatic, since a human expert must decide whether there is seasonality in the time series or not and what is the relevant frequency. To overcome this limitation we used the output of the INF to identify the periodicities to setup the benchmark models automatically. This model family is selected as a benchmark due to its proven track record in univariate time series forecasting [3]. For more details on exponential smoothing models and the guidelines that were used to implement them in this analysis see [41].

#### IV. RESULTS

The ability of the INF to automatically and correctly identify the underlying frequencies in each time series was assessed by comparing the outputted frequencies with the true data generating process. The success rate was 100%. The performance of the method was not affected by the different frequencies or the different structure of the time series.

Using the outputs of the INF the inputs were selected using the primed regression and the other two competing alternatives. Table I provides the aggregated accuracy statistics for the simulated data. These are aggregated by the nature of the components of the time series (deterministic or stochastic) and noise levels to assess the conditions under which each model performs better. Furthermore, in table I the benchmark model accuracy is provided. Note that in

TABLE II  
TEST SET STATISTICAL TESTS

Friedman p-value		
0.000		
Rankin		
Method	Mean rank	g
Primed_NN	50.31	1
Normal_NN	64.37	2
Stoch_NN	66.82	3

The three alternative models are significantly different. The Nemenyi test provides the ranking of the models, considering individual statistically significant differences between models. Models that have a difference of mean ranks less than 0.2 are not significantly different under the 5% significance level. The model with the lowest mean rank performs statistically the best.

order to make this model fully automated the identified frequency output of the INF was used, resulting in a hybrid INF-EXSM method. In the same table, overall accuracies per sample and model are also reported. For both the deterministic and the stochastic case the *Normal\_NN* fits better to the training set, but is less accurate than the *Primed\_NN* in both the validation and test sets. With the exception of the stochastic low noise case, where EXSM performs the best, the proposed algorithm (*Primed\_NN*)

consistently outperforms all other models. All models are able to perfectly model time series with no noise. As expected the *Stoch\_NN* that models only the stochastic part of the time series performs better when there are stochastic components in the time series. Similarly the *Normal\_NN* performs better when there are deterministic components. *Primed\_NN* that can distinguish between the two cases, performs best. Note that when the accuracy in the test set is considered the benchmark *EXSM* comes second only to *Primed\_NN*, demonstrating the adequacy of this benchmark as it has been observed in previous studies [3, 8].

Testing for statistically significant differences among the different models a Friedman test is first applied to identify if at least one model is different under 5% significance level and then the post-hoc Nemenyi test is used to provide the ranking. The results are found in table II. The *Primed\_NN* is found to significantly outperform other alternatives.

Table III provides the results for the real time series. *Primed\_NN* is found to outperform other NN specifications and the benchmark *EXSM* in agreement with the results presented in table I.

#### V. CONCLUSION

This paper proposes a methodology for automatically identifying the input vector for forecasting with NNs for time series that exhibit unknown frequencies. An Iterative Neural Filter (INF) is proposed for automatic frequency identification and feature extraction. The output of this filter is transformed to best fit the stochastic or deterministic nature of the time series and then inputted in a primed regression that extends the input vector to capture the dynamic structure of the time series. The principle of the methodology is to combine the flexible nature and approximation capabilities of the NNs with efficient filter approaches to produce a robust and accurate automatic input selection methodology that is scalable.

The performance of the proposed methodology was demonstrated using simulated and real time series. The data were pooled to produce an heterogeneous dataset and the INF was asked to automatically identify the frequency of each time series and identify and periodicities that are present. The filter successfully identified a set of preliminary inputs that were then fed to the rest of the methodology to eventually produce input vectors for a fixed setup MLP, in a fully automated process. The proposed methodology outperformed other NN alternatives and statistical benchmark models.

This study proposes a solution to the automatic specification of the input vector for neural networks for forecasting. The architecture of the NN was fixed, which may have penalised their performance. We need to evaluate strategies for automatically readjusting the setup of the NN, ideally without using wrappers, in order to retain the scalability of the proposed algorithm.

Last but not least, the proposed INF was used in order to fully automated exponential smoothing models, which

require a human expert to set the relevant time series frequency. It was shown that a hybrid INF-statistical model approach is readily implementable, providing a fully automated solution with well accepted and established forecasting methods.

#### REFERENCES

- [1] G. Q. Zhang, B. E. Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *International Journal of Forecasting*, vol. 14, no. 1, pp. 35-62, Mar, 1998.
- [2] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.
- [3] S. Makridakis, and M. Hibon, "The M3-Competition: results, conclusions and implications," *International Journal of Forecasting*, vol. 16, no. 4, pp. 451-476, Oct-Dec, 2000.
- [4] J. S. Armstrong, "Findings from evidence-based forecasting: Methods for reducing forecast error," *International Journal of Forecasting*, vol. 22, no. 3, pp. 583-598, 2006.
- [5] D. W. Bunn, "Non-traditional methods of forecasting," *European Journal of Operational Research*, vol. 92, no. 3, pp. 528-536, Aug, 1996.
- [6] N. Kourentzes, "PhD Thesis: Input variable selection for time series forecasting with artificial neural networks - an empirical evaluation across varying time series frequencies," Department of Management Science, Lancaster University, Lancaster, 2009.
- [7] U. Anders, and O. Korn, "Model selection in neural networks," *Neural Networks*, vol. 12, no. 2, pp. 309-323, Mar, 1999.
- [8] K. P. Liao, and R. Fildes, "The accuracy of a procedural approach to specifying feedforward neural networks for forecasting," *Computers & Operations Research*, vol. 32, no. 8, pp. 2151-2169, Aug, 2005.
- [9] G. P. Zhang, B. E. Patuwo, and M. Y. Hu, "A simulation study of artificial neural networks for nonlinear time-series forecasting," *Computers & Operations Research*, vol. 28, no. 4, pp. 381-396, Apr, 2001.
- [10] G. P. Zhang, "An investigation of neural networks for linear time-series forecasting," *Computers & Operations Research*, vol. 28, no. 12, pp. 1183-1202, Oct, 2001.
- [11] M. Nelson, T. Hill, W. Remus *et al.*, "Time series forecasting using neural networks: Should the data be deseasonalized first?," *Journal of Forecasting*, vol. 18, no. 5, pp. 359-367, Sep, 1999.
- [12] S. F. Crone, and N. Kourentzes, "Forecasting seasonal time series with multilayer perceptrons - an empirical evaluation of input vector specifications for deterministic seasonality," *WORLDCOMP 2009, DMIN 2009, Las Vegas*, pp. 232-238, 2009.
- [13] G. P. Zhang, and D. M. Kline, "Quarterly time-series forecasting with neural networks," *Ieee Transactions on Neural Networks*, vol. 18, no. 6, pp. 1800-1814, Nov, 2007.
- [14] B. Curry, "Neural networks and seasonality: Some technical considerations," *European Journal of Operational Research*, vol. 179, no. 1, pp. 267-274, May, 2007.
- [15] R. L. Goodrich, "The Forecast Pro methodology," *International Journal of Forecasting*, vol. 16, no. 4, pp. 533-535, Oct-Dec, 2000.
- [16] R. J. Hyndman, and Y. Khandakar, "Automatic time series forecasting: The forecast package for R," *Journal of Statistical Software*, vol. 27, no. 3, pp. 1-22, Jul, 2008.
- [17] S. Makridakis, S. C. Wheelwright, and R. J. Hyndman, *Forecasting: Methods and Applications*, 3rd ed., p. 642: John Wiley & Sons, Inc., 1998.
- [18] R. Fildes, and C. Beard, "Forecasting Systems for Production and Inventory Control," *International Journal of Operations & Production Management*, vol. 12, no. 5, pp. 4-27, 1991.
- [19] R. Fildes, K. Nikolopoulos, S. F. Crone *et al.*, "Forecasting and operational research: a review," *Journal of the Operational Research Society*, vol. 59, no. 9, pp. 1150-1172, Sep, 2008.
- [20] C. W. J. Granger, "Extracting information from mega-panels and high-frequency data," *Statistica Neerlandica*, vol. 52, no. 3, pp. 258-272, Nov, 1998.
- [21] J. W. Taylor, L. M. de Menezes, and P. E. McSharry, "A comparison of univariate methods for forecasting electricity demand up to a day ahead," *International Journal of Forecasting*, vol. 22, no. 1, pp. 1-16, Jan-Mar, 2006.
- [22] N. Kourentzes, and S. F. Crone, "Automatic modelling of neural networks for time series prediction - in search of a uniform methodology across varying time frequencies," *European Symposium on Time Series Prediction, ESTSP '08, Porvoo, Finland*, 2008.
- [23] S. F. Crone, and N. Kourentzes, "Input-variable Specification for Neural Networks - an Analysis of Forecasting low and high Time Series Frequency," *Proceedings of the International Joint Conference on Neural Networks, IJCNN'09, Atlanta, USA, IEEE*, pp. 3221-3228, 2009.
- [24] E. S. Gardner, "Exponential smoothing: The state of the art - Part II," *International Journal of Forecasting*, vol. 22, no. 4, pp. 637-666, 2006.
- [25] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251-257, 1991.
- [26] U. Anders, O. Korn, and C. Schmitt, "Improving the pricing of options: A neural network approach," *Journal of Forecasting*, vol. 17, no. 5-6, pp. 369-388, Sep-Nov, 1998.
- [27] G. P. Zhang, and M. Qi, "Neural network forecasting for seasonal and trend time series," *European Journal of Operational Research*, vol. 160, no. 2, pp. 501-514, Jan, 2005.
- [28] M. Qi, and G. P. Zhang, "Trend time-series modeling and forecasting with neural networks," *Ieee Transactions on Neural Networks*, vol. 19, no. 5, pp. 808-816, May, 2008.
- [29] S. Heravi, D. R. Osborn, and C. R. Birchenhall, "Linear versus neural network forecasts for European industrial production series," *International Journal of Forecasting*, vol. 20, no. 3, pp. 435-446, Jul-Sep, 2004.
- [30] E. Ghysels, and D. R. Osborn, *The Econometric Analysis of Seasonal Time Series*, Cambridge: Cambridge University Press, 2001.
- [31] F. Canova, and B. E. Hansen, "Are seasonal patterns constant over time - a test for seasonal stability," *Journal of Business & Economic Statistics*, vol. 13, no. 3, pp. 237-252, Jul, 1995.
- [32] M. Efron, "Multiple Regression Analysis," *Mathematical Methods for Digital Computers*, A. Ratson and H. Wilf, eds.: Wiley, 1960.
- [33] D. Montgomery, A. Peck, and G. Vining, *Introduction to Linear Regression Analysis, 4th Edition*: Wiley, 2006.
- [34] R. Hyndman. "Time Series Data Library," 01/04/2010; <http://robjhyndman.com/TSDL/>.
- [35] L. Tashman, "Out-of-sample tests of forecasting accuracy: an analysis and review," *International Journal of Forecasting*, vol. 16, pp. 437-450, 2000.
- [36] D. Janez, and ar, "Statistical Comparisons of Classifiers over Multiple Data Sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1-30, 2006.
- [37] M. T. Hagan, H. B. Demuth, and M. H. Beale, *Neural Network Design*, Boston: PWS Publishing, 1996.
- [38] N. R. Swanson, and H. White, "Forecasting economic time series using flexible versus fixed specification and linear versus nonlinear econometric models," *International Journal of Forecasting*, vol. 13, no. 4, pp. 439-461, Dec, 1997.
- [39] S. D. Balkin, and J. K. Ord, "Automatic neural network modeling for univariate time series," *International Journal of Forecasting*, vol. 16, no. 4, pp. 509-515, Oct-Dec, 2000.
- [40] M. Adya, and F. Collopy, "How effective are neural networks at forecasting and prediction? A review and evaluation," *Journal of Forecasting*, vol. 17, no. 5-6, pp. 481-495, Sep-Nov, 1998.
- [41] E. J. Gardner, "Exponential smoothing: The state of the art--Part II," *International Journal of Forecasting*, vol. 22, no. 4, pp. 637-666, 2006.