Tourism Demand Forecasting with Neural Network Models: Different Ways of Treating Information

OSCAR CLAVERIA[1]*, ENRIC MONTE[2] and SALVADOR TORRA[3]

[1]*AQR-IREA, University of Barcelona, Barcelona, Spain*
[2]*Department of Signal Theory and Communications, Polytechnic University of Catalunya, Barcelona, Spain*
[3]*Riskcenter-IREA, Department of Econometrics, University of Barcelona, Barcelona, Spain*

ABSTRACT

This paper aims to compare the performance of three different artificial neural network techniques for tourist demand forecasting: a multi-layer perceptron, a radial basis function and an Elman network. We find that multi-layer perceptron and radial basis function models outperform Elman networks. We repeated the experiment assuming different topologies regarding the number of lags used for concatenation so as to evaluate the effect of the memory on the forecasting results. We find that for higher memories, the forecasting performance obtained for longer horizons improves, suggesting the importance of increasing the dimensionality for long-term forecasting. Copyright © 2014 John Wiley & Sons, Ltd.

INTRODUCTION

International tourism is one of the fastest growing industries nowadays. As a result of this rise in the number of tourists and the importance of the tourism sector, tourism demand forecasting has become increasingly important. Some of the reasons for this growing interest, apart from the constant growth of world tourism, are related to the availability of more advanced forecasting techniques and the requirement for more accurate forecasts of tourism demand at the destination level. Balaguer and Cantavella-Jordá (2002) showed the importance of tourism in the Spanish long-run economic development. Catalonia is a region of Spain and one of the world's major tourist destinations. More than 15 million foreign visitors came to Catalonia in 2012, a 3.7% rise with respect to the previous year. Tourism accounts for 12% of GDP and provides employment for 15% of the working population in Catalonia. These figures show to what extent accurate forecasts of tourism volume play a major role in tourism planning as they enable destinations to predict infrastructure development needs.

The last couple of decades have seen many studies of international tourism demand forecasting, but few studies have used artificial neural networks (ANNs). ANNs have been applied in the many fields but only recently to tourism demand forecasting (Kon & Turner, 2005; Palmer *et al.*, 2006; Cho, 2009; Chen, 2011; Teixeira & Fernandes, 2012). Despite that there is no consensus on the most appropriate approach to forecast tourism demand, it is generally believed that the non-linear methods outperform the linear methods in modelling economic behaviour (Choudhary & Haider, 2012; Cang, 2013). These non-linear models are still limited in that an explicit relationship for the data series has to be hypothesized with little knowledge of the underlying data-generating process (Zhang *et al.*, 1998).

Since there are too many possible non-linear patterns, the formulation of a non-linear model to a particular data set is a very difficult task. In a recent meta-analysis of published tourism forecasting studies, Kim and Schwartz (2013) find that forecast accuracy is closely associated with data characteristics.

Artificial NNs are data-driven approaches and are capable of performing non-linear modelling without *a priori* knowledge about the relationships between input and output variables. As opposed to traditional approaches to time-series prediction, the specification of ANN models does not depend on a previous set of assumptions. Thus, ANNs are a more general and flexible modelling tool for forecasting. The suitability of artificial intelligence techniques to handle non-linear behaviour explains why ANNs have become an essential tool for economic forecasting. The flexibility of the structure of ANNs allows for the introduction of knowledge about the nature of the analysis to be carried, tailoring the topology of the network in order to make use of the specific properties of the problem at hand. The fact that tourism data are characterized by strong seasonal patterns and volatility makes it a particularly interesting field in which to apply different types of NN architectures (Chen & Wang, 2007; Medeiros *et al.*, 2008; Hadavandi *et al.*, 2011; Hong *et al.*, 2011; Shahrabi *et al.*, 2013; Pai *et al.*, 2014). In spite of the increasing interest in machine learning methods for time-series forecasting, very few studies compare the accuracy of different NN architectures for tourism demand forecasting.

The main objective of this study is to improve forecasts of tourism demand and to compare the performance of three different ANN models: the multi-layer perceptron (MLP) network, the radial basis function (RBF) network and the Elman network. Each architecture represents a different learning paradigm and therefore deals with data in a different manner. With this aim, we undertake an out-of-sample forecasting competition to analyse the forecasting accuracy of each ANN model to predict inbound international tourism demand from all visitor markets to Catalonia. We compute several

*Correspondence to: Oscar Claveria, AQR-IREA, Department of Econometrics and Statistics, University of Barcelona, Diagonal 690, 08034 Barcelona, Spain. E-mail: oclaveria@ub.edu

measures of forecast accuracy and the Diebold–Mariano test for significant differences between each two competing series for different forecast horizons (one, three and six months) in order to assess the value of the different models. We also evaluate the effect of the memory on the forecasting results by repeating the experiment assuming different topologies regarding the number of input neurons, which determines the number of prior time points to be used in each forecast. To our knowledge, this is the first study to compare the forecasting performance of MLP, RBF and Elman ANNs and to analyse the effect of the memory values on the forecasting accuracy.

The rest of the paper is organized as follows. First, we review the literature related to tourism demand forecasting with ANNs. In the next section, a brief description of each type of NN used in the analysis is made. Then, an analysis of the data set is provided. In the subsequent section, results of the out-of-sample forecasting competition are presented and discussed. The final section concludes.

LITERATURE REVIEW

Originally, ANNs were developed to mimic basic biological neural systems. ANNs are composed of a number of interconnected processing nodes called neurons. Through an activation transfer function, each node processes the input signal received from other nodes and produces a transformed output signal to other nodes. Some of the characteristics that make ANNs more advantageous over other methods in terms of robustness and tolerance to error are their ability of parallel processing, distributed memory and adaptability (Palmer *et al.*, 2006). The introduction of the backpropagation algorithm by Rumelhart *et al.* (1986) fostered the development in the use of ANNs for forecasting. The literature comparing the forecasting performance of ANNs to traditional statistical methods such as regression-based and autoregressive integrated moving average (ARIMA) models is vast and growing. In recent works, Lin *et al.* (2011) and Claveria and Torra (2014) find that ARIMA models outperform ANN models; nevertheless, most studies provide empirical evidence in favour of ANNs (Tang *et al.*, 1991; Weigend & Gershenfeld, 1993; Zhang *et al.*, 1998; Marcellino, 2005). Teräsvirta *et al.* (2005) obtain more accurate forecasts with ANN models at long forecast horizons.

Many different ANN models have been developed since the 1980s. NNs can be classified into two major types of architectures depending on the connecting patterns of the different layers: feed-forward networks, where the information runs only in one direction, and recurrent networks, in which there are feedback connections from outer layers of neurons to lower layers of neurons, which takes into account the temporal structure of the data. Feed-forward NNs were the first ANN devised. The most widely used feed-forward topology in time-series forecasting is the MLP network. MLP networks have been widely used for tourism demand forecasting (Pattie & Snyder, 1996; Law, 1998, 2000, 2001; Law & Au, 1999; Uysal & El Roubi, 1999, Burger *et al.*, 2001; Tsaur *et al.*, 2002). These studies provide empirical results indicating that NNs outperform regression models and time-series models in terms of forecasting accuracy, especially for series without obvious patterns.

A special class of multi-layer feed-forward architecture with two layers of processing is the RBF network (Broomhead and Lowe, 1988). The fact that RBF networks have the advantage of not suffering from local minima in the same way as MLP networks explains their increasing use in tourism demand forecasting. Chen (2011) used combinations of backpropagation and support vector regression (SVR) networks to forecast Taiwanese outbound tourism demand, obtaining the best performance with the SVR combination models. Cang (2013) combined different time-series linear models as inputs to MLP, RBF and SVR networks to forecast inbound tourist arrivals to the UK, finding evidence in favour of all non-linear combinations of models with respect of the usual linear combination models that currently dominate in the tourism forecasting literature.

Contrary to feed-forward networks, recurrent NNs are models with bidirectional data flow. While a feed-forward network propagates data linearly from input to output, recurrent networks also propagate a temporal feedback from the outer layers to the lower layers. There are many recurrent architectures: fully recurrent, simple recurrent, bidirectional recurrent, etc. A special case of recurrent networks are the Elman networks. While MLP NNs are increasingly used with forecasting purposes, other more computationally expensive architectures such as the Elman NN have been scarcely used in tourism demand forecasting. Cho (2003) used the Elman architecture to predict the number of arrivals from different countries to Hong Kong. He found that Elman NNs provided better forecasts than exponential smoothing and ARIMA models. Teixeira and Fernandes (2012) compared the forecasting performance of feed-forward, cascade-forward and recurrent networks but did not find significant differences between the different architectures.

Regarding their learning strategy, ANNs can also be classified into two major types of architectures: supervised and unsupervised learning networks. In supervised learning networks, weights are adjusted to approximate the output to a target value for each pattern of entry. Support vector machines (SVMs) and MLP networks are examples of supervised learning models. In non-supervised learning networks, the subjacent structure of data patterns is explored so as to organize such patterns according to their correlations. Kohonen self-organizing maps are the most used non-supervised models. Some NNs combine both learning methods, so part of the weights are determined by a supervised process while the rest are determined by unsupervised learning. This is known as hybrid learning. An example of a hybrid model is the RBF network. Therefore, each network is suited to a combination of a learning paradigm (supervised and non-supervised learning), a learning rule related to the gradient cost function and a learning algorithm (forwardpropagation, backpropagation, etc.). The different learning paradigms represent alternative approaches to the treatment of information.

In this study, we focus on three NN architectures (MLP, RBF and Elman) that represent three learning paradigms:

supervised, hybrid and recurrent. Therefore, each ANN model deals with data in a different manner. In spite of the recent advances in SVM-based models (Pai & Hong, 2005; Chen & Wang, 2007), in this study, we have not used the SVM network as a benchmark because we aim to compare techniques that are mathematically similar. While the general operation performed by MLP, RBF and Elman networks is to transform the input to an internal representation and then into the desired output, the SVM technique is based on finding a set of inputs that are representative of the desired regression.

METHODOLOGY

Neural networks are flexible structures capable of learning sequentially from observed data. This feature makes ANNs specially suitable for time-series forecasting. Obtaining a reliable neural model involves selecting a large number of parameters experimentally: determining the number of input nodes, hidden layers, hidden nodes and output nodes, the activation function, the training algorithm, the training and the test samples, as well as the performance measures for cross validation (Zhang *et al.*, 1998). This range of different choices allows choosing the optimal topology of the ANN, while the weights of the model are estimated by gradient search. A complete summary on ANN modelling issues can be found in Bishop (1995) and Haykin (1999).

Multi-layer perceptron neural network
Multi-layer perceptron networks consist of multiple layers of computational units interconnected in a feed-forward way. MLP networks are supervised NNs that use as a building block a simple perceptron model. The topology consists of layers of parallel perceptrons, with connections between layers that include optimal connections. The number of neurons in the hidden layer determines the MLP network's capacity to approximate a given function. In order to solve the problem of overfitting, the number of neurons was estimated by cross validation. In this work, we used the MLP specification suggested by Bishop (1995) with a single hidden layer and an optimum number of neurons derived from a range between 5 and 25:

$$y_t = \beta_0 + \sum_{j=1}^{q} \beta_j g\left(\sum_{i=1}^{p} \varphi_{ij} x_{t-i} + \varphi_{0j}\right)$$
$$\left\{x_{t-i} = \left(1, x_{t-1}, x_{t-2}, \cdots, x_{t-p}\right)', i = 1, ..., p\right\}$$
$$\left\{\varphi_{ij}, i = 1, ..., p, j = 1, \cdots, q\right\}$$
$$\left\{\beta_j, j = 1, ..., q\right\}$$

(1)

where y_t is the output vector of the MLP at time t, g is the non-linear function of the neurons in the hidden layer, x_{t-i} is the input value at time $t-i$ where i stands for the memory (the number of lags that are used to introduce the context of the actual observation.), q is the number of neurons in the hidden layer, φ_{ij} are the weights of neuron j connecting the

input with the hidden layer, and β_j are the weights connecting the output of the neuron j at the hidden layer with the output neuron. Note that the output y_t in our study is the estimate of the value of the time series at time $t+1$, while the input vector to the NN will have a dimensionality of $p+1$.

We considered an MLP $(p; q)$ architecture that represented the possible non-linear relationship between the input vector x_{t-i} and the output vector y_t. The parameters of the network (φ_{ij} and β_j) were estimated by means of the Levenberg–Marquardt algorithm, which is a quasi-Newton algorithm. The training was performed by iteratively estimating the value of the parameters by local improvements of the cost function. To avoid the possibility that the search for the optimum value of the parameters finishes in a local minimum, we used a multi-starting technique that initializes the NN several times for different initial random values returning the best result.

Radial basis function neural network
Radial basis function networks consist of a linear combination of RBFs such as kernels centred at a set of centroids with a given spread that controls the volume of the input space represented by a neuron (Bishop, 1995). RBF networks typically include three layers: an input layer, a hidden layer and an output layer. The hidden layer consists of a set of neurons, each of them computing a symmetric radial function. The output layer also consists of a set of neurons, one for each given output, linearly combining the outputs of the hidden layer. The output of the network is a scalar function of the output vector of the hidden layer. The equations that describe the input/output relationship of the RBF are

$$y_t = \beta_0 + \sum_{j=1}^{q} \beta_j g_j(x_{t-i})$$
$$g_j(x_{t-i}) = \exp\left(-\frac{\sum_{j=1}^{p}\left(x_{t-i} - \mu_j\right)^2}{2\sigma_j^2}\right)$$
$$\left\{x_{t-i} = \left(1, x_{t-1}, x_{t-2}, ..., x_{t-p}\right)', i = 1, ..., p\right\}$$
$$\left\{\beta_j, j = 1, ..., q\right\}$$

(2)

where y_t is the output vector of the RBF at time t, β_j are the weights connecting the output of the neuron j at the hidden layer with the output neuron, q is the number of neurons in the hidden layer, g_j is the activation function, which usually has a Gaussian shape, x_{t-i} is the input value at time $t-i$ where i stands for the memory (the number of lags that are used to introduce the context of the actual observation), $\varpi\mu_j$ is the centroid vector for neuron j, and the spread σ_j is a scalar that measures the width over the input space of the Gaussian function, and it can be defined as the area of influence of neuron j in the space of the inputs. Note that the output y_t in our study is the estimate of the value of the time series at time $t+1$, while the input vector to the NN will have a dimensionality of $p+1$.

In order to assure a correct performance, before the training phase, the number of centroids and the spread of each centroids have to be selected. The spread σ_j is a hyperparameter selected before determining the topology of the network, and it was determined by cross validation on the training database. The training was performed by adding the centroids iteratively with the spread fixed. Then, a regularized linear regression was estimated to compute the connections between the hidden and the output layer. Finally, the performance of the network was computed on the validation data set. This process was repeated until the performance on the validation database ceased to decrease.

Elman neural network

An Elman network is a special architecture of the class of recurrent NNs, and it was first proposed by Elman (1990). The architecture is also based on a three-layer network but with the addition of a set of context units that allow feedback on the internal activation of the network. There are connections from the hidden layer to these context units fixed with a weight of one. At each time step, the input is propagated in a standard feed-forward fashion, and then, a backpropagation type of learning rule is applied. The output of the network is a scalar function of the output vector of the hidden layer:

$$y_t = \beta_0 + \sum_{j=1}^{q} \beta_j z_{j,t}$$

$$z_{j,t} = g\left(\sum_{i=1}^{p} \varphi_{ij} x_{t-i} + \varphi_{0j} + \delta_{ij} z_{j,t-1} \right)$$

$$\left\{ x_{t-i} = \left(1, x_{t-1}, , x_{t-2}, \cdots, , x_{t-p} \right)', i = 1, \ldots, p \right\} \quad (3)$$

$$\left\{ \phi_{ij}, i = 1, \ldots, p, j = 1, \ldots, q \right\}$$

$$\left\{ \beta_j, j = 1, \ldots, q \right\}$$

$$\left\{ \delta_{ij}, i = 1, \ldots, p, j = 1, \ldots, q \right\}$$

where y_t is the output vector of the Elman network at time t, $z_{j,t}$ is the output of the hidden layer neuron j at the moment t, g is the non-linear function of the neurons in the hidden layer, x_{t-i} is the input value at time $t-i$ where i stands for the memory (the number of lags that are used to introduce the context of the actual observation), ϕ_{ij} are the weights of neuron j connecting the input with the hidden layer, q is the number of neurons in the hidden layer, β_j are the weights of neuron j that link the hidden layer with the output, and δ_{ij} are the weights that correspond to the output layer and connect the activation at moment t. Note that the output y_t in our study is the estimate of the value of the time series at time $t+1$, while the input vector to the NN will have a dimensionality of $p+1$.

The training of the network was performed by backpropagation through time, which is a generalization of backpropagation for feed-forward networks. The parameters of the Elman NN were estimated by minimizing an error cost function, which takes into account the whole time series. In order to minimize total error, gradient descent was used to change each weight in proportion to its derivative with respect to the error, provided that the non-linear activation functions are differentiable. A major problem with gradient descent for standard recurrent architectures is that error gradients vanish exponentially quickly with the size of the time lag. Recurrent NNs cannot be easily trained for large numbers of neuron units or input units and may behave chaotically and present scaling issues.

DATA

Monthly data of tourist arrivals over the time period January 2001 to July 2012 were provided by the Direcció General de Turisme de Catalunya and the Statistical Institute of Catalonia (Institut d'Estadística de Catalunya). Following Narayan (2003), we have computed some of the most commonly used methods to test the unit root hypothesis: the augmented Dickey–Fuller (ADF) test and the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test. While the ADF tests the null hypothesis of a unit root in x_t, the KPSS statistic tests the null hypothesis of stationarity.

As it can be seen in Table 1, in most countries, we cannot reject the null hypothesis of a unit root at the 5% level. Similar results are obtained for the KPSS test, where the null hypothesis of stationarity is rejected in most cases. When the tests were applied to the first difference of individual time

Table 1. Unit root tests on the seasonally adjusted series of tourist arrivals

Country	Test for I(0)		Test for I(1)		Test for I(2)	
	ADF	KPSS	ADF	KPSS	ADF	KPSS
France	−2.39	0.60	−3.19	0.64	−5.11	0.12
UK	−1.63	0.38	−2.98	0.51	−18.92	0.12
Belgium and the Netherlands (NL)	−3.56	0.24	−2.49	0.21	−8.43	0.02
Germany	−1.93	0.50	−3.54	0.33	−8.76	0.15
Italy	−1.58	0.71	−3.55	0.52	−5.47	0.26
USA and Japan	2.08	1.19	−4.77	0.39	−6.92	0.02
Northern countries	−1.14	1.24	−3.88	0.06	−11.41	0.03
Switzerland	−3.26	0.38	−6.14	0.07	−6.20	0.16
Russia	1.80	1.06	−3.62	0.65	−8.37	0.04
Other countries	−1.33	1.30	−4.53	0.07	−9.88	0.02
Total	−1.98	0.87	−2.97	0.29	−12.51	0.06

Note: estimation period January 2001–July 2012.
Tests for unit roots. ADF, augmented Dickey and Fuller (1979) test, the 5% critical value is −2.88; KPSS, Kwiatkowski, Phillips, Schmidt and Shin (1992) test, the 5% critical value is 0.46.

series, the null of non-stationarity is strongly rejected in most cases. In the case of the KPSS test, we cannot reject the null hypothesis of stationarity at the 5% level in any country. These results imply that differencing is required in most cases and prove the importance of deseasonalizing and detrending tourism demand data before modelling and forecasting.

We use the year-on-year rates of the trend-cycle component of the series to eliminate both linear trends as well as seasonality. The seasonally adjusted series are obtained using a Census X13 filter. Table 2 shows a descriptive analysis of year-on-year rates of the trend-cycle series between January 2002 and July 2012. During this period, Russia and the Northern countries experienced the highest growth in tourist arrivals. Russia is also the country that presents the highest dispersion in growth rates, while France shows the highest levels of skewness and kurtosis.

RESULTS

In this section, we compare the forecasting performance of three different ANN architectures (MLP, RBF and Elman NNs) to predict arrivals to Catalonia from the different visitor countries. Following Bishop (1995), we divide the collected data into three sets: training, validation and test sets. This division is performed in order to assess the performance of the network on unseen data. The partition of the original database in three sets is performed to better control the prediction error in the estimation process of the parameters. We use the percentages that are often used in the literature (Ripley, 1996). Based on these considerations, the first 60 monthly observations from January 2002 to December 2006 (50%) are selected as the initial training set, the next 36 from January 2007 to December 2009 (30%) as the validation set and the last 20% as the test set.

Due to the large number of possible networks' configurations, the validation set is used for determining the following aspects of the NNs:

(1) The topology of the networks.
(2) The number of epocs for the training of the MLP NNs. The iterations in the gradient search are stopped when the error on the validation set increases.

Table 2. Descriptive analysis of the year-on-year rates of the seasonally adjusted series

| Country | Tourist arrivals | | | |
	Mean	SD	Skew.	Kurt.
France	5.06	13.69	2.13	8.93
UK	1.94	15.00	0.70	3.51
Belgium and NL	1.85	8.50	0.76	3.13
Germany	0.45	7.85	0.14	3.13
Italy	5.48	14.58	0.88	3.39
USA and Japan	4.77	11.14	−0.08	2.64
Northern countries	8.24	16.97	0.25	2.70
Switzerland	−0.21	9.86	0.28	4.93
Russia	16.06	32.12	−0.35	2.69
Other countries	6.90	10.02	−0.15	2.48
Total	3.75	7.04	−0.75	3.04

Note: SD, standard deviation; Skew., skewness; Kurt., kurtosis.

(3) The number of neurons in the hidden layer for the RBF. The sequential increase in the number of neurons at the hidden layer is stopped when the error on the validation increases.
(4) The value of the spread σ_j in the RBF NN. As the value of the spread increases, a much higher number of centroids are needed.

To make the system robust to local minima, we apply the multi-starting technique, which consists on repeating each training phase several times. We repeat the training three times so as to obtain a low value of the performance error. The selection criterion for the topology and the parameters is the performance on the validation set. The results that are presented correspond to the selection of the best topology, the best spread in the case of the RBF NNs and the best training strategy in the case of the Elman NNs. Forecasts for one, three and six months ahead are computed in a recursive way, which implies that models are reestimated in each period and for each forecasting horizon. All NNs are implemented using Matlab™ and its NN toolbox.

In order to summarize this information, we use the two most commonly used measures of forecasting accuracy: the root-mean-squared error (RMSE) and the mean absolute percentage error (MAPE). The results of our forecasting competition are shown in Tables 3 and 4. We also used the Diebold–Mariano test (Table 5) for significant differences between each two competing series for each forecast horizon in order to assess the value of the different models. We repeat the experiment assuming different topologies regarding the memory values. These values represent the number of lags introduced when running the models, ranging from one to three months for all the architectures. Therefore, when the memory is zero, the forecast is performed using only the current value of the time series, without any additional temporal context.

When analysing the forecast accuracy for tourist arrivals, MLP and RBF networks show lower RMSE and MAPE values than Elman networks, especially for shorter horizons. RBF networks display the lowest RMSE and MAPE values in most countries when the memory is zero. When the forecasts are obtained incorporating additional lags of the time series, the forecasting performance of MLP networks improves. The lowest RMSE and MAPE value are obtained with the MLP network for France (for one month ahead) when using a memory of three lags.

When testing for significant differences between each two competing series (Table 5), we find that MLP and RBF networks significantly outperform Elman networks in all countries and for all forecasting horizons. A possible explanation for this result is the length of the time series used in the analysis. The fact that the number of training epocs had to be low in order to maintain the stability of the network suggests that this network architecture requires longer time series. For long training phases, the gradient sometimes diverged. The worse forecasting performance of the Elman NNs compared with that of MLP and RBF architectures for topologies with no memory indicates that the feedback topology of the Elman network could not capture the specificity of the time series. These results are

in contrast to those obtained by Teixeira and Fernandes (2012) and Cho (2003), who obtained a good forecasting performance with Elman networks.

When comparing the forecasting performance between MLP and RBF networks, we find that the RBF architecture produces the best forecasts when the memory of the network is set to zero, while the MLP architecture improves its forecasting performance when a larger number of lags are incorporated in the networks. This result can be explained because in this case, the RBF operates as a look-up table,

while the MLP tries to find a functional relationship lacking a context that might give a hint of the slope of the time series. As the number of lags increases, MLP networks obtain significantly better forecasts for some countries (France, Italy, Northern countries and the USA and Japan). This result can be explained by the fact that as the hidden neurons linearly combine the input before applying the non-linearity, additional lags can be used in a better way to estimate the different slopes and the future evolution of the series. This evidence indicates that the number of previous months used for concatenation conditions the forecasting performance of the different networks.

Table 3. MAPE (April 2010–February 2012)

	Memory (0) – no additional lags			Memory (3) – three additional lags		
	ANN models			ANN models		
France	MLP	RBF	Elman	MLP	RBF	Elman
One month	*0.33*	0.34	9.02	*0.06**	0.09	7.85
Three months	5.36	1.39	10.96	1.11	1.30	8.39
Six months	5.72	2.22	6.91	2.64	3.24	5.63
UK						
One month	*0.34*	0.57	2.55	1.59	1.32	2.00
Three months	4.92	2.81	3.31	*1.22*	2.22	2.06
Six months	8.72	3.15	2.16	3.52	2.21	12.04
Belgium and the NL						
One month	1.12	0.83	3.77	1.39	1.50	2.74
Three months	1.20	*0.79*	2.02	1.37	1.58	2.79
Six months	2.99	0.97	2.07	3.99	*0.95*	2.44
Germany						
One month	5.57	4.95	12.47	6.43	6.37	16.42
Three months	2.01	*1.83*	5.92	*5.72*	6.66	13.76
Six months	2.14	3.30	4.74	7.66	8.34	16.04
Italy						
One month	*1.32*	1.84	17.63	*0.77*	2.18	20.35
Three months	9.74	10.42	24.83	8.51	5.92	23.81
Six months	11.76	13.45	11.52	22.76	13.56	20.39
USA and Japan						
One month	0.90	*0.80*	1.52	0.49	*0.48*	2.31
Three months	1.85	1.70	4.16	1.05	1.56	2.67
Six months	1.01	0.94	3.93	1.94	1.68	1.85
Northern countries						
One month	0.42	*0.41*	2.82	0.38	*0.28*	1.59
Three months	1.49	1.13	2.19	0.52	1.11	2.05
Six months	1.39	1.17	3.52	0.92	1.02	2.83
Switzerland						
One month	1.33	1.25	2.39	1.63	1.32	1.15
Three months	0.83	0.65	1.47	1.60	1.12	1.74
Six months	0.76	*0.50*	2.35	0.95	*0.57*	1.37
Russia						
One month	0.57	*0.53*	0.74	0.49	0.52	0.69
Three months	0.62	0.54	0.72	*0.42*	0.46	0.62
Six months	0.65	0.66	0.88	0.61	0.76	1.01
Other countries						
One month	0.41	*0.35*	1.30	0.54	0.60	1.78
Three months	0.92	0.64	1.91	*0.50*	0.51	1.81
Six months	1.01	0.68	1.96	0.67	0.61	3.05
Total						
One month	*0.64*	0.65	3.55	0.60	*0.57*	2.64
Three months	2.02	0.73	3.14	1.29	0.85	2.85
Six months	3.25	0.77	2.75	1.70	2.20	2.64

Table 4. RMSE (2010:04–2012:02)

	Memory (0) – no additional lags			Memory (3) – three additional lags		
	ANN models			ANN models		
France	MLP	RBF	Elman	MLP	RBF	Elman
One month	0.49	*0.48*	24.38	*0.12**	0.31	18.71
Three months	6.93	1.85	20.33	2.15	1.71	18.51
Six months	10.28	3.71	17.14	6.63	5.48	13.41
UK						
One month	*3.35*	7.81	20.53	*5.02*	6.10	13.08
Three months	15.27	8.85	21.03	8.11	9.54	12.07
Six months	23.84	9.58	17.45	12.25	14.17	19.60
Belgium and the NL						
One month	9.63	6.35	19.58	8.73	8.50	17.25
Three months	7.31	*3.90*	19.69	7.38	8.33	14.04
Six months	15.30	5.07	15.87	20.06	*5.46*	12.67
Germany						
One month	9.04	8.52	18.33	10.47	9.50	17.99
Three months	6.81	5.13	22.39	8.82	8.70	13.65
Six months	11.00	*4.78*	11.56	10.02	*8.05*	17.74
Italy						
One month	*1.85*	1.93	12.37	*1.20*	1.93	14.14
Three months	4.78	5.29	16.79	7.08	4.56	15.64
Six months	10.82	10.47	16.43	14.18	10.74	14.90
USA and Japan						
One month	6.00	*4.96*	15.26	5.94	*5.84*	18.87
Three months	11.15	9.88	24.53	8.86	11.13	20.51
Six months	12.73	15.08	20.31	11.28	10.95	13.28
Northern countries						
One month	5.34	*5.27*	22.77	*3.56*	3.80	20.48
Three months	11.71	11.25	20.04	5.15	7.65	16.87
Six months	16.19	15.10	26.69	15.19	12.09	28.67
Switzerland						
One month	12.13	10.86	26.52	14.63	12.03	12.26
Three months	7.90	*5.92*	16.65	15.71	11.26	19.08
Six months	11.14	5.95	26.31	11.84	*7.37*	15.29
Russia						
One month	33.38	*28.64*	38.66	*25.91*	28.46	36.93
Three months	39.13	32.53	35.19	25.99	28.93	34.12
Six months	39.64	37.38	56.48	37.11	41.42	59.06
Other countries						
One month	3.22	*2.90*	13.70	2.94	3.06	14.45
Three months	7.61	6.38	15.79	3.54	*2.89*	16.89
Six months	9.48	8.87	15.88	7.11	6.52	20.22
Total						
One month	3.94	*3.90*	17.25	*4.14*	4.23	15.75
Three months	11.40	4.83	17.72	7.28	5.28	15.32
Six months	21.84	4.27	13.86	14.05	13.28	12.89

Note: The entries in italics are the best model for each country.
*Best model.

Note: The entries in italics are the best model for each country.
*Best model.

Table 5. Diebold–Mariano loss-differential test statistic for predictive accuracy (2.028 critical value)

	Memory (0) – no additional lags			Memory (3) – three additional lags		
	MLP versus RBF	MLP versus Elman	RBF versus Elman	MLP versus RBF	MLP versus Elman	RBF versus Elman
France						
One month	0.88	−6.12*	−6.08*	−2.23*	−6.19*	−6.14*
Three months	1.38	−4.37*	−5.05*	0.33	−10.12*	−12.05*
Six months	1.36	−1.95	−3.64*	0.33	−3.11*	−3.92*
UK						
One month	−1.62	−7.10*	−4.68*	−1.13	−4.20*	−3.17*
Three months	0.46	−1.65	−2.58*	−1.42	−1.70	−1.11
Six months	2.01	0.65	−2.40*	−1.24	−1.69	−0.88
Belgium and the NL						
One month	2.50*	−2.38*	−3.26*	0.36	−3.19*	−3.28*
Three months	2.27*	−2.62*	−3.59*	−0.09	−2.91*	−2.49*
Six months	2.19*	−0.47	−2.91*	1.67	0.61	−2.54*
Germany						
One month	2.58*	−3.51*	−3.85*	1.64	−1.99	−2.38*
Three months	1.86	−3.62*	−3.92*	−0.34	−1.79	−1.84
Six months	0.79	−1.72	−3.11*	0.82	−1.20	−1.75
Italy						
One month	−1.33	−5.83*	−5.74*	−2.89*	−9.01*	−8.81*
Three months	−0.38	−5.10*	−4.78*	1.57	−4.41*	−6.29*
Six months	−0.57	−2.53*	−1.77	1.03	−0.25	−1.85
USA and Japan						
One month	4.49*	−4.98*	−5.98*	−0.62	−4.77*	−4.90*
Three months	0.64	−4.63*	−5.46*	−2.73*	−6.09*	−3.56*
Six months	0.14	−0.94	−0.93	−0.54	−0.89	−0.07
Northern countries						
One month	1.11	−5.55*	−5.54*	−0.12	−4.00*	−3.83*
Three months	1.44	−2.90*	−3.06*	−3.32*	−7.12*	−4.68*
Six months	0.77	−2.65*	−2.81*	0.62	−6.64*	−5.81*
Switzerland						
One month	1.52	−2.76*	−3.02*	2.29*	2.68*	0.50
Three months	1.96	−1.61	−1.96	2.85*	0.01	−2.22*
Six months	1.33	−5.08*	−8.10*	1.33	−1.50	−2.65*
Russia						
One month	1.40	−1.97	−3.07*	−2.01	−3.47*	−2.69*
Three months	1.40	0.48	−1.33	−1.65	−1.74	−0.88
Six months	0.91	−3.18*	−3.10*	−1.62	−4.54*	−2.99*
Other countries						
One month	0.80	−5.48*	−6.34*	−0.25	−5.69*	−5.64*
Three months	2.94*	−3.07*	−3.91*	0.72	−6.56*	−7.17*
Six months	1.36	−2.01	−2.69*	0.03	−3.75*	−4.69*
Total						
One month	−0.50	−6.55*	−6.96*	0.45	−4.01*	−4.00*
Three months	1.02	−2.92*	−4.23*	0.38	−7.46*	−5.79*
Six months	2.21*	0.91	−3.66*	−0.45	−0.75	−0.55

Note: Diebold–Mariano test statistic with NW estimator. Null hypothesis: The difference between the two competing series is non-significant. A negative sign of the statistic implies that the second model has bigger forecasting errors.
*Significant at the 5% level.

The differences between countries can be partly explained by different patterns of consumer behaviour, but they are also related to the variability due to the size of the sample, being France as the most important visitor market. When comparing the results for different prediction horizons, as it could be expected, the forecasting performance improves for shorter forecasting horizons. Nevertheless, we find that there is an interaction between the memory and the forecasting horizon. As it can be seen in Tables 3 and 4, as the number of lags used in the networks increases, the forecasting performance obtained for longer horizons (three and six months) improves.

CONCLUSIONS

The increasing importance of the tourism sector worldwide has led to a growing interest in new approaches to tourism demand forecasting. New methods provide more accurate estimations of anticipated tourist arrivals for effective policy planning. Artificial intelligence techniques such as ANNs have attracted increasing interest to refine the predictions of tourist arrivals at the destination level. From the wide array of NN models, we have focused on three different architectures that represent three alternative ways of handling information: the MLP NN, the RBF NN and the Elman recursive NN.

The main objective of this study is to improve forecasts of tourism demand by using different ANN models and to compare the forecasting performance of the different architectures. Each architecture represents a different learning paradigm and a different way of estimating the parameters of the model. First, we predict inbound international tourism demand from all visitor markets to Catalonia and for different forecast horizons. We then test for significant differences between each two competing series in order to assess the value of the different models. Finally, we evaluate the effect of the memory on the forecasting results by repeating the experiment assuming different topologies regarding the number of prior time points to be used in each forecast.

When comparing the forecasting accuracy of the different techniques, we find that MLP and RBF NNs outperform Elman NNs. These results are in contrast with the evidence found in previous studies and suggest that issues related with the divergence of the Elman NN may arise when using dynamic networks with forecasting purposes. The comparison of the forecasting performance between MLP and RBF NNs permits to conclude that the RBF networks significantly outperform the MLP networks when no additional lags are introduced in the networks. On the contrary, when the input has a context of the past, MLP networks show a better forecasting performance.

We also find that as the amount of previous months used for concatenation increases, the forecasts obtained for longer horizons improve, suggesting the importance of increasing the dimensionality of the input to networks for long-term forecasting. An input that takes into account a larger number of prior time points might capture not only the trend of the current value but also possible cycles that influence the forecast. These results show that the number of lags introduced in the networks plays a fundamental role on the forecasting performance of the different architectures.

This study contributes to the literature and to the tourism industry by highlighting the most suitable NNs and how to implement them in order to improve the forecasting accuracy of tourism demand. Nevertheless, this study is not without its limitations. The overparametrization problem found with Elman networks could be partially solved if longer time series of tourist arrivals were available. A question to be considered in further research is whether a combination of forecasts from alternative topologies and different time aggregations could improve the accuracy of tourism demand forecasting. Finally, another question to be addressed in further research is whether these results apply to different data pre-processing methods.

ACKNOWLEDGEMENTS

REFERENCES

Balaguer J, Cantavella-Jordá M. 2002. Tourism as a long-run economic growth factor: the Spanish case. *Applied Economics* **34**: 877–884.

Bishop CM. 1995. *Neural networks for pattern recognition*. Oxford University Press: Oxford.

Broomhead DS, Lowe D. 1988. Multi-variable functional interpolation and adaptive networks. *Complex Systems* **2**: 321–355.

Burger C, Dohnal M, Kathrada M, Law R. 2001. A practitioners guide to time-series methods for tourism demand forecasting: A case study for Durban, South Africa. *Tourism Management* **22**: 403–409.

Cang S. 2013. A comparative analysis of three types of tourism demand forecasting models: individual, linear combination and non-linear combination. *International Journal of Tourism Research* **15**. Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/jtr.1953.

Chen K. 2011. Combining linear and nonlinear model in forecasting tourism demand. *Expert Systems with Applications* **38**: 10368–10376.

Chen K, Wang CH. 2007. Support Vector Regression with Genetic Algorithms in Forecasting Tourism Demand. *Tourism Management* **28**: 215–226.

Cho V. 2003. A comparison of three different approaches to tourist arrival forecasting. *Tourism Management* **24**: 323–330.

Cho V. 2009. A study on the temporal dynamics of tourism demand in the Asia Pacific Region. *International Journal of Tourism Research* **11**: 465–485.

Choudhary A, Haider A. 2012. Neural network models for inflation forecasting: An appraisal. *Applied Economics* **44**: 2631–2635.

Claveria O, Torra S. 2014. Forecasting tourism demand to Catalonia: Neural networks vs. time series models. *Economic Modelling* **36**: 220–228.

Elman JL. 1990. Finding structure in time. *Cognitive Science* **14**: 179–211.

Hadavandi E, Ghanbari A, Shahanaghi K, Abbasian S. 2011. Tourist arrival forecasting by evolutionary fuzzy systems. *Tourism Management* **32**: 1196–1203.

Haykin S. 1999. *Neural networks. A comprehensive foundation*. Prentice Hall: New Jersey.

Hong W, Dong Y, Chen L, Wei S. 2011. SVR with Hybrid Chaotic Genetic Algorithms for Tourism Demand Forecasting. *Applied Soft Computing* **11**: 1881–1890.

Kim D, Schwartz Y. 2013. The accuracy of tourism forecasting and data characteristics: a meta-analytical approach. *Journal of Hospitality Marketing amd Management* **22**: 349–374.

Kon SC, Turner WL. 2005. Neural network forecasting of tourism demand. *Tourism Economics* **11**: 301–328.

Law R. 1998. Room occupancy rate forecasting: A neural network approach. *International Journal of Contemporary Hospitality Management* **10**: 234–239.

Law R. 2000. Back-propagation learning in improving the accuracy of neural network-based tourism demand forecasting. *Tourism Management* **21**: 331–340.

Law R. 2001. The impact of the Asian financial crisis on Japanese demand for travel to Hong Kong: a study of various forecasting techniques. *Journal of Travel & Tourism Marketing* **10**: 47–66.

Law R, Au N. 1999. A neural network model to forecast Japanese demand for travel to Hong Kong. *Tourism Management* **20**: 89–97.

Lin CJ, Chen HF, Lee TS. 2011. Forecasting tourism demand using time series, artificial neural networks and multivariate adaptive regression splines: Evidence from Taiwan. *International Journal of Business Administration* **2**: 14–24.

Marcellino M. 2005. Instability and non-linearity in the EMU. In Milas C, Rothman P, van Dijk D (eds). *Nonlinear Time Series Analysis of Business Cycles*. Elsevier: Amsterdam.

Medeiros MC, McAleer M, Slottje D, Ramos V, Rey-Maquieira J. 2008. An Alternative Approach to Estimating Demand: Neural

Network Regression with Conditional Volatility for High Frequency Air Passenger Arrivals. *Journal of Econometrics* **147**: 372–383.

Narayan PK. 2003. Tourism demand modelling: some issues regarding unit roots, co-integration and diagnostic tests. *International Journal of Tourism Research* **5**: 369–380.

Pai P, Hong W. 2005. An Improved Neural Network Model in Forecasting Arrivals. *Annals of Tourism Research* **32**: 1138–1141.

Pai P, Hung K, Lin K. 2014. Tourism Demand Forecasting Using Novel Hybrid System. *Expert Systems with Applications* **41**: 3691–3702.

Palmer A, Montaño JJ, Sesé A. 2006. Designing an artificial neural network for forecasting tourism time-series. *Tourism Management* **27**: 781–790.

Pattie DC, Snyder J. 1996. Using a neural network to forecast visitor behavior. *Annals of Tourism Research* **23**: 151–164.

Ripley BD. 1996. *Pattern recognition and neural networks*. Cambridge University Press: Cambridge.

Rumelhart DE, Hinton GE, Williams RJ. 1986. Learning representations by backpropagation errors. *Nature* **323**: 533–536.

Shahrabi J, Hadavandi E, Asadi S. 2013. Developing a Hybrid Intelligent Model for Forecasting Problems: Case Study of Tourism Demand Time Series. *Knowledge-Based Systems* **43**: 112–122.

Tang Z, Almeida C, Fishwick PA. 1991. Time series forecasting using neural networks vs. *Box-Jenkins methodology. Simulation* **57**: 303–310.

Teixeira JP, Fernandes PO. 2012. Tourism Time Series Forecast – Different ANN Architectures with Time Index Input. *Procedia Technology* **5**: 445–454.

Teräsvirta T, van Dijk D, Medeiros MC. 2005. Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. *International Journal of Forecasting* **21**: 755–774.

Tsaur SH, Chiu YC, Huang CH. 2002. Determinants of guest loyalty to international tourist hotels: a neural network approach. *Tourism Management* **23**: 397–405.

Uysal M, El Roubi MS. 1999. Artificial neural Networks versus multiple regression in tourism demand analysis. *Journal of Travel Research* **38**: 111–118.

Weigend AS, Gershenfeld NA. 1993. *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley: Reading, MA.

Zhang G, Putuwo BE, Hu MY. 1998. Forecasting with artificial neural networks: the state of the art. *International Journal of Forecasting* **14**: 35–62.