APPLICATION NOTE

New developments in the forecasting of monthly overnight stays in the North Region of Portugal*

Isabel Silva[a] and Hugo Alonso[b]

[a]Faculdade de Engenharia, Universidade do Porto and CIDMA, Portugal; [b]Universidade de Aveiro, CIDMA and Universidade Lusófona do Porto, Portugal

ARTICLE HISTORY
Compiled July 23, 2020

ABSTRACT
The Tourism sector is of strategic importance to the North Region of Portugal and is growing. Forecasting monthly overnight stays in this region is, therefore, a relevant problem. In this paper, we analyse data more recent than those considered in previous studies and use them to develop and compare several forecasting models and methods. We conclude that the best results are achieved by models based on a non-parametric approach not considered so far for these data, the singular spectrum analysis.

KEYWORDS
Forecasting; neural networks; overnight stays; singular spectrum analysis; time series

1. Introduction

Over the last years, the Tourism sector has been growing in the North Region of Portugal, creating jobs and attracting wealth [see 15, and references therein]. Forecast the monthly overnight stays in the region is an important issue, because it gives an early view of tourism demand and makes easier to manage tourist accommodations. It is clear that accurate forecast of tourism demand is crucial for administrators, policy makers and investors.

There are some studies in the literature where this forecasting problem was already considered. In [1], two autoregressive integrated moving average models and a neural network delivered one month ahead forecasts based on the overnight stays in the previous twelve months. This study considered data until 2006 and concluded that the best results were produced by the neural network. In turn, in [12], a linear regression model and a neural network used the monthly sunshine hours time series to predict the monthly overnight stays time series. The data considered reached 2010 and the best performance was achieved again by the neural network. Finally, in [15–17], several neural network models, differing in their architectures and input data, were developed and compared. The data considered went until 2009 in [16], 2010 in [17] and 2015

* Accepted authors manuscript (AAM) published in Journal of Applied Statistics, DOI: 10.1080/02664763.2020.1795812. Final source webpage: https://doi.org/10.1080/02664763.2020.1795812
Isabel Silva's email: ims@fe.up.pt
Hugo Alonso's email: hugo.alonso@ua.pt

in [15]. Hence, previous works focused on developing neural networks for monthly overnight stays forecasting in the North Region of Portugal and such non-linear models delivered the best forecasts. Neural networks present the advantage of making no assumptions about the data distribution and the time series stationarity [6]. This is also the case with singular spectrum analysis [4]. However, as far as we know, neural networks were not compared with this technique. Singular spectrum analysis was already successfully applied to tourism demand forecasting, but in other places, like in the United States of America [5] and South Africa [13].

The aim of this work is twofold. Firstly, we develop and compare several approaches to the problem of forecasting monthly overnight stays in the North Region of Portugal, namely: neural networks, singular spectrum analysis, the seasonal naïve method, seasonal autoregressive integrated moving average models and exponential smoothing models. Thus, we carry on a comparative study more complete than those shown in previous studies, with newer forecasting techniques.

Secondly, the behaviour of the monthly overnight stays time series may have changed as a consequence of the increasing tourism demand and the fact that the North Region of Portugal has won several prestigious tourism awards in the last years. Therefore, we want to verify if the good performance of the forecasts in previous works is maintained. Note that we consider more recent data, until 2017, while previous works have considered data until 2015, at most.

The remainder of this paper is organized as follows. The next section presents the data used in this work. The forecasting methods and models are described in Section 3 and the results are shown in Section 4. Finally, Section 5 presents the conclusions and future work.

2. Data

We used data concerning the monthly overnight stays time series in the North Region of Portugal (this region includes the districts of Viana do Castelo, Braga, Porto, Vila Real and Bragança, and partially the districts of Aveiro, Viseu and Guarda). The data refer to the period between January 2009 and June 2017, totalling 102 observations. Figure 1 depicts the time series. It is clear that there is an annual seasonality: monthly overnight stays start with a minimum value in January, then increase, attaining the maximum in August, and finally decrease until December. Moreover, there is an upward trend since 2013 and a larger amplitude from 2013 to 2015. This is confirmed by the sample measures of the data presented in Table 1. As we can see, the annual mean of the monthly overnight stays tends to increase from year to year and the variability around the mean, measured by the coefficient of variation, remains close to 30% until 2012 and around 35% up to 2015, decreasing again in 2016.

Table 1. Sample measures for the monthly overnight stays in the North Region of Portugal.

Year	2009	2010	2011	2012	2013	2014	2015	2016	2017[a]
Mean overnight stays (thousands)	356	370	379	378	440	505	509	574	548
Coefficient of variation (%)	32	29	33	32	36	36	34	30	29

[a] Only first semester.

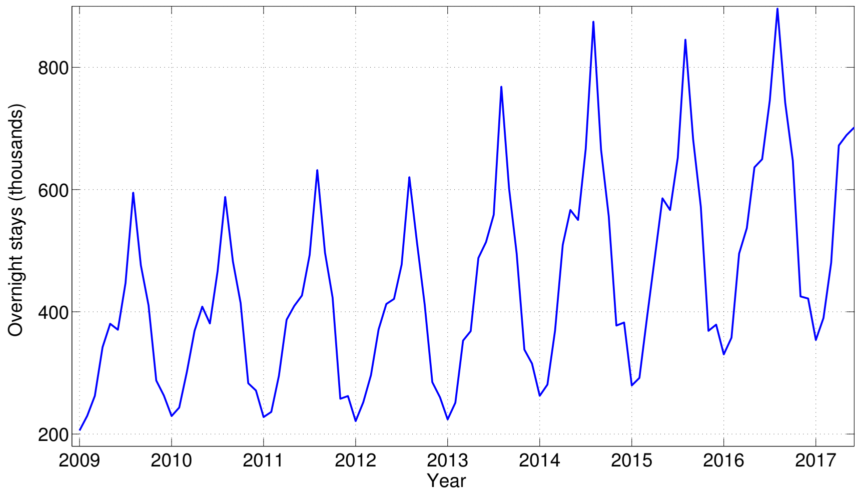

Figure 1. Monthly overnight stays in the North Region of Portugal.

3. Methods and Models

In this section, we briefly explain the several methods and models used to forecast the time series of the monthly overnight stays in the North Region of Portugal, namely neural networks, singular spectrum analysis, the seasonal naïve method, seasonal autoregressive integrated moving average models and exponential smoothing models.

3.1. Neural Networks

In this paper, we consider multilayer feedforward neural networks [6]. Let $x_1, \ldots, x_n$ be the n inputs and y^{net} the only output of a multilayer feedforward neural network with one hidden layer of m neurons, as shown in Figure 2. The neurons in the input layer do not process data and serve only to forward it to the neurons in the next layer. The neurons in the hidden layer have a sigmoid activation function, namely the hyperbolic tangent. The output of the i-th hidden neuron is given by

$$z_i = g\left(x_1, \ldots, x_n \,|w_{i1}, \ldots, w_{in}, w_{i0}\right) = \tanh\left(\sum_{j=1}^{n} w_{ij}x_j + w_{i0}\right), \qquad i = 1, \ldots, m,$$

where w_{ij} is the weight of the connection from the j-th input neuron, with $j = 1, \ldots, n$, to the i-th hidden neuron and w_{i0} is a weight called the bias of the i-th hidden neuron. The neuron in the output layer has a linear activation function, namely the identity. Its output is given by

$$y^{net} = h\left(z_1, \ldots, z_m \,|w_1, \ldots, w_m, w_0\right) = \sum_{i=1}^{m} w_i z_i + w_0,$$

where w_i is the weight of the connection from the i-th hidden neuron, with $i = 1, \ldots, m$, to the output neuron and w_0 is a weight called the bias of the output neuron. Hence, the neural network implements a function

$$y^{net} = f(x_1, \ldots, x_n \mid m, \mathbf{w}) = \sum_{i=1}^{m} w_i \tanh\left(\sum_{j=1}^{n} w_{ij} x_j + w_{i0}\right) + w_0,$$

parameterized in m, the number of hidden neurons, and $\mathbf{w}$, the vector of the $m(n+2)+1$ network weights. The values of these parameters can be determined as explained next.

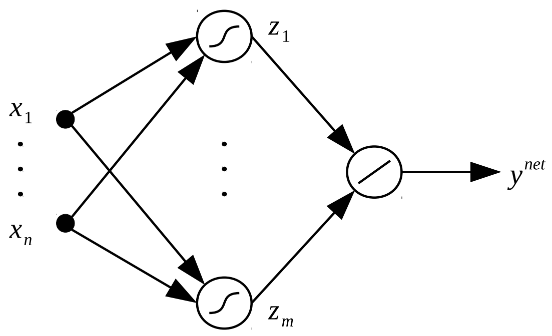

Figure 2. Multilayer feedforward neural network.

Suppose that the data available to determine the values of the neural network parameters, m and $\mathbf{w}$, are split into two sets: a training set

$$T = \left\{ \left(x_1^{(T,\ell)}, \ldots, x_n^{(T,\ell)}; y^{(T,\ell)} \right) \right\}_{\ell=1}^{n_T},$$

with n_T cases, where $y^{(T,\ell)}$ is the desired output of the network for the input $(x_1^{(T,\ell)}, \ldots, x_n^{(T,\ell)})$, and a validation set

$$V = \left\{ \left(x_1^{(V,\ell)}, \ldots, x_n^{(V,\ell)}; y^{(V,\ell)} \right) \right\}_{\ell=1}^{n_V},$$

with n_V cases, where $y^{(V,\ell)}$ is the desired output of the network for the input $(x_1^{(V,\ell)}, \ldots, x_n^{(V,\ell)})$. Define the training error as

$$E_T(m, \mathbf{w}) = \sum_{\ell=1}^{n_T} \left(y^{(T,\ell)} - f\left(x_1^{(T,\ell)}, \ldots, x_n^{(T,\ell)} \mid m, \mathbf{w} \right) \right)^2$$

and the validation error as

$$E_V(m, \mathbf{w}) = \sum_{\ell=1}^{n_V} \left(y^{(V,\ell)} - f\left(x_1^{(V,\ell)}, \ldots, x_n^{(V,\ell)} \mid m, \mathbf{w} \right) \right)^2.$$

Fixing $m = k$, for a certain $k \in \mathbb{Z}^+$, let $\mathbf{w} = \mathbf{w}^{(k)}$ represent a solution to the non-linear least squares problem

$$\min_{\mathbf{w}} E_T(m = k, \mathbf{w}),$$

found by applying a suitable optimization algorithm, like Levenberg-Marquardt's [10]. The sequence of the training errors $E_T\left(m = 1, \mathbf{w} = \mathbf{w}^{(1)}\right)$, $E_T\left(m = 2, \mathbf{w} = \mathbf{w}^{(2)}\right)$, ... tends to decrease with m, which is a measure of the network complexity (the higher the value of m, the greater the network complexity). In turn, the sequence of the validation errors $E_V\left(m = 1, \mathbf{w} = \mathbf{w}^{(1)}\right)$, $E_V\left(m = 2, \mathbf{w} = \mathbf{w}^{(2)}\right)$, ... tends to decrease until a certain value of m, say $m = k^\star$, and then starts to increase. In this context, we take $m = k^\star$

and $\mathbf{w} = \mathbf{w}^{(k^\star)}$ for the neural network parameters.

In this paper, we apply multilayer feedforward neural networks to monthly overnight stays forecasting, using Matlab. The output y^{net} of each network corresponds to an overnight stays forecast for a given month. In what concerns the inputs $x_1, \ldots, x_n$, we consider two possibilities, motivated by previous successful works [1, 12, 15–17]. In the first case, we take two inputs, corresponding to the year and the month for which a forecast is desired. This approach assumes that the trend of the time series is given by the year and the seasonality by the month. In the second case, we take twelve inputs, corresponding to the overnight stays in the twelve months previous to the one for which a forecast is desired. This is justified by the fact that the time series shows an annual seasonality. In both cases, the training data refer to the years 2009-2014 and the validation data to the year 2015.

3.2. Singular Spectrum Analysis

The Singular Spectrum Analysis (SSA) is a technique for time series analysis and forecasting that decomposes a time series into a small number of independent and interpretable components that can be considered as trend, oscillatory components and noise. No stationarity assumptions or parametric models for the time series are needed. SSA has been widely applied on several fields, see [3, 4] for references.

Basic SSA technique consists of two stages, namely *decomposition and reconstruction*, and then the reconstructed series is used for forecasting. Consider a real-valued time series $Y_T = [\; y_1 \quad y_1 \quad \cdots \quad y_T \;]$ and a window length L $(1 < L < T)$. The decomposition stage starts with the *embedding* step, where the so called trajectory matrix (a $L \times K$ Hankel matrix) is built with L-lagged vectors $X_i = (y_i, \ldots, y_{i+L-1})'$, for $i = 1, 2, \ldots, K$, where $K = N - L + 1$ and v' indicates transpose of v, that is,

$$\mathbf{X} = [\; X_1 \quad X_2 \quad X_3 \quad \cdots \quad X_K \;] = \begin{bmatrix} y_1 & y_2 & y_3 & \cdots & y_K \\ y_2 & y_3 & y_4 & \cdots & y_{K+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_L & y_{L+1} & y_{L+2} & \cdots & y_T \end{bmatrix}.$$

The second step of the decomposition is the *Singular Value Decomposition* (SVD) of $\mathbf{X}$ such that

$$\mathbf{X} = \mathbf{X}_1 + \mathbf{X}_2 + \cdots + \mathbf{X}_d,$$

where

$$\mathbf{X}_i = \sqrt{\lambda_i}\, V_i\, U_i{}',$$

$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L$ are the eigenvalues and $U_1, U_2, \ldots, U_L$ are the corresponding eigenvectors of $\mathbf{X}\mathbf{X}'$, $V_i = \mathbf{X}\, U_i / \sqrt{\lambda_i}$ and

$$d = \operatorname{rank}(\mathbf{X}) = \max\{i : \lambda_i > 0\} \leq L.$$

The collection (λ_i, U_i, V_i) is called the i-th eigentriple of the SVD.

The first step of the reconstruction stage is *grouping*, where the set index $1, \ldots, d$ is partitioned into M disjoint subsets $I_1, \ldots, I_M$, where $I_k = \{i_{k_1}, \ldots, i_{k_p}\}$. Then the so called resultant matrix $\mathbf{X}_{I_k}$ corresponding to the group I_k is defined as $\mathbf{X}_{I_k} = \mathbf{X}_{i_{k_1}} + \cdots + \mathbf{X}_{i_{k_p}}$, and it is usually associated with a pattern of interest (trend or seasonality, for instance). Then the result of this step is the grouped matrix decomposition

$$\mathbf{X} = \mathbf{X}_{I_1} + \cdots + \mathbf{X}_{I_M}.$$

In the last step of the reconstruction, called *diagonal averaging*, each matrix $\mathbf{X}_{I_k} = \left[x_{ij}{}^{(k)}\right]_{i,j=1}^{L,K}, k = 1, \ldots, M$, is transformed into a new time series of length T, $\tilde{\mathbf{X}}_{I_k} = \{\tilde{y}_1^{(k)}, \ldots, \tilde{y}_T^{(k)}\}$, where $\tilde{y}_t^{(k)}$ is obtained by averaging $x_{ij}{}^{(k)}$ over all $i, j : i + j = t + 1, \quad t = 1, \ldots T$. Therefore, the initial series $Y_T = [\begin{array}{cccc} y_1 & y_1 & \cdots & y_T \end{array}]$ is decomposed into a sum of M reconstructed series

$$y_t = \sum_{k=1}^{M} \tilde{y}_t^{(k)}, \ t = 1, \ldots, T.$$

In practice, to apply SSA it is necessary to choose the window length L, and the way to select the M in the grouping step. Usually, $L \approx T/2$ or depends of the periodicity of data (L proportional to the period). The inspection of the singular values (λ_i) and vectors (U_i, V_i) may help in the selection of M. A slowly decreasing sequence of singular values indicates a pure noise series, while an explicit plateaux is likely to be yielded by a pair of eigenvectors which correspond to a harmonic components. Additionally, the pattern of an eigenvector replicates the form of the time series component that produces this eigenvector. The analysis of the pairwise scatterplots of the singular vectors allows the identification of the eigentriples that correspond to the harmonic components of the series. For instance, pure harmonics create the scatterplot with the points lying on a circle. If the period of the harmonic is an integer, then in the scatterplot the points are the vertices of the regular polygon. For a detailed explanation see Section 2.4 of [4].

In SSA (recurrent) forecasting, it is assumed that the time series to be forecast can be described through the linear recurrence relations (LRR)

$$y_{T-i} = \sum_{k=1}^{L-1} a_k y_{T-i-k}, \quad 0 \leq i \leq T - L,$$

where the coefficients a_j are uniquely defined. The space spans by the trajectory matrix determines a LRR of dimension $L - 1$ that governs the series. Then, the forecasting points are obtained by the application of this LRR to the last $L - 1$ observations of the series. Suppose that $Y_T = Y_T^{(1)} + Y_T^{(2)}$, and we want to forecast $Y_T^{(1)}$ ($Y_T^{(2)}$ can be regarded as noise). The main assumption allowing SSA forecasting is that for a certain window length L, $Y_T^{(1)}$ and $Y_T^{(2)}$ are approximately strongly separable (for details see [3, 4]). In this work, the forecast of $\hat{y}_{T+1}, \ldots, \hat{y}_{T+h}$ is obtained by the SSA recurrent forecasting algorithm as

$$\hat{y}_i = \begin{cases} \tilde{y}_i, & i = 1, \ldots, T, \\ \displaystyle\sum_{j=1}^{L-1} a_j \hat{y}_{i-j}, & i = T + 1, \ldots, T + h, \end{cases}$$

where $(\tilde{y}_1, \ldots, \tilde{y}_T)$ is the reconstructed series,

$$A = (a_1, \ldots, a_{L-1}) = \frac{1}{1 - v^2} \sum_{i=1}^{r} \pi_i U_i^{\nabla}$$

is the vector of coefficients of the LRR, for $v^2 = \pi_1^2 + \cdots + \pi_r^2 < 1$, where π_i is the last component of the eigenvector U_i, and $U_i^{\nabla} \in \mathbb{R}^{L-1}$ are the first $L + 1$ components of U_i. Additionally, as referred by [4], bootstrap confidence intervals can be obtained by calculating the empirical distribution of the residuals which is used to perform bootstrap series simulation. Then, these bootstrapped series are forecast. The bootstrap confidence interval is given by the interval between $(1-\gamma)/2$-lower and upper sample quantiles. The sample mean is called average bootstrap forecast. In this work, the `Rssa` package in R [11] is used to compute the SSA forecasting (for details see [2]).

3.3. Seasonal Naïve Method

As defined by [8], when the data clearly present seasonality, the forecast can be set as the last observed value from the same *season* of the year, for instance, the same month of the previous year for monthly data or the same quarter of the previous year for quarterly data. Formally, taking into account the observations $(y_1, \ldots, y_T)$, the forecast for time $T+h$ can be defined as

$$\hat{y}_{T+h|T} = y_{T+h-mk},$$

where m is the seasonal period, and $k = \lfloor (h-1)/m \rfloor + 1$ (where $\lfloor a \rfloor$ represents the largest integer less than or equal to a). In this work it is used the `snaive()` function from the `forecast` package [9] in R [11].

3.4. Seasonal Autoregressive Integrated Moving Average Models

The pioneer work of Box and Jenkins allows to model and to forecast dependent data by the past values of an independent variable plus its own past values through the so called Autoregressive Moving Average Model (ARMA). Several modifications were made to the classical ARMA model to consider seasonal and nonstationary behaviour. As defined in [14], the multiplicative Seasonal Autoregressive Integrated Moving Average model (SARIMA), with seasonal period S, denoted by ARIMA$(p, d, q) \times (P, D, Q)_S$, is defined by:

$$\phi(B)\Phi(B^S)(1-B^S)^D(1-B)^d y_t = \theta(B)\Theta(B^S)e_t,$$

where e_t is a white noise with variance σ_e^2, $By_t = y_{t-1}$ is the backshift operator, $\phi(z) = 1 - a_1 z - \ldots - a_p z^p$ is the AR polynomial, $\theta(z) = 1 + b_1 z + \ldots + b_q z^q$ is the MA polynomial, $\Phi(z^S) = 1 - \alpha_1 z^S - \ldots - \alpha_P z^{PS}$ is the seasonal AR polynomial and $\Theta(z^S) = 1 + \beta_1 z^S + \ldots + \beta_Q z^{QS}$ is the seasonal MA polynomial. It is assumed that the roots of the four polynomials lie outside the unit circle.

As proposed by [8], in this work is used a variation of the Hyndman-Khandakar algorithm [9] which incorporates unit root tests to select the order of the differences , d and D, minimisation of the AICc (corrected Akaike Information Criterion) to select the orders p, q, P and Q, and maximum likelihood estimation (MLE) to obtain the parameters of the chosen SARIMA model. Instead of considering every possible combination of the orders, the algorithm uses a stepwise search. The choices of the algorithm are driven by forecast accuracy. The steps of the algorithm are implemented in the `auto.arima()` function from the `forecast` package [9] in R [11].

3.5. Exponential Smoothing Forecast

In the exponential smoothing (ES) method the forecast is obtained as weighted averages of past observations, with the weights decaying exponentially as the observations come from further in the past, which means that the smallest weights are associated with the oldest observations [7]. Although the method was proposed around 1950, the ES only incorporates prediction intervals, maximum likelihood estimation and procedures for model selection with the approach based on the innovations state space model (for details see [7]). Using this approach, it was shown that all ES methods are optimal forecasts from innovation state space models. However, it is important to note that the ES method is an algorithm for producing point forecasts only when the underlying stochastic state space model gives the same point forecasts, but also provides a framework for computing prediction intervals and other properties.

A time series can be viewed as a collection of several components such as the trend (T), seasonal (S) and irregular or error (E) components, among other. These three components can be mixed in a number of different ways, providing additive, multiplicative or mixed class of models. As referred in [7], to produce a point forecast by the ES method, the trend component must be chosen in first place as a combination of a level term and a growth term. Having chosen a trend component, the seasonal component can be introduced, either additively or multiplicatively. Finally, an error is included, either additively or multiplicatively. Thus, it is possible to consider 30 potential models.

As proposed in [7], the models with additive (A), additive damped (Ad), multiplicative (M) and multiplicative damped (Md) errors can be distinguished by using the triplet (E,T,S) which refers to Error, Trend and Seasonality components. For instance, the model ETS(A,A,N) has additive errors, additive trend and no seasonality. Note that some of these models are better known under other names, for instance, Holt's linear, Holt-Winters' additive and Holt-Winters' multiplicative method. Furthermore, ETS can also be considered an abbreviation of "*E*xponen*T*ial *S*moothing".

In this work, it is used the *automatic forecasting* procedure provided by [7], where each of 30 ES methods that are appropriated are applied to the data; then the parameters of the models are optimized by using MLE method and the best model is selected according to the minimum AIC or a different information criterion like AICc or BIC (Bayesian IC), among other. Point forecasts can be obtained through the best model (with optimized parameters) for as many steps ahead as required. Finally, the underlying state space model can be used to produce prediction intervals. This algorithm is implemented through the `forecast()` function from the `forecast` package [9] in R [11].

4. Forecasting Results

In this section, the performance of the forecasting obtained by the methods and models previously described is compared.

As referred by [8], to measure the accuracy of forecasts we can verify the behaviour of a model applied to new data that were not used when fitting the model. Thus, the available set of observations can be split into two parts, training and test data, where the parameters of a forecasting method are obtained from the training data and the test data is used to assess its accuracy. Since the test data is not used to obtain the forecasts, it should indicate how well the model carried on the forecast on new data. A usual scale-dependent measure is the Root Mean Squared Error, defined by

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2},$$

while a frequently used unit-free measure is the Mean Absolute Percentage Error, MAPE, defined as

$$\text{MAPE} = \frac{100}{n} \sum_{t=1}^{n} \left| \frac{y_t - \hat{y}_t}{y_t} \right|,$$

where $\hat{y}_t$ is the forecast of the observation y_t (of the test data).

In this work, the forecasting are built with the 2009-2015 data (training and validation data) and compared with data from the year 2016 and the first semester of 2017 (test data, $n = 18$).

In what concerns neural networks (ANNs), as referred before, we have implemented two architectures. The first one (denoted by ANN Year-Month) uses two inputs, corresponding to the year and the month for which a forecast is desired. In turn, the second approach (ANN 12 Months) uses twelve inputs, corresponding to the overnight stays in the twelve months previous to the one for which a forecast is desired. The procedure we followed to select the number of hidden neurons in both cases was described in Subsection 3.1 and is illustrated next. Figure 3 shows MAPE as a function of the number of hidden neurons in the ANN Year-Month. It can be seen that, in the training set, MAPE tends to zero as the number of neurons increases. Hence, one hidden layer with a sufficiently large number of neurons is enough for this architecture to fit the training data. However, as is well known, the greater the number of hidden neurons, the greater the possibility of overfitting. In the validation set, MAPE decreases until 5 neurons and then tends to increase, that is, the generalization ability of the architecture tends to degrade beyond that point. Therefore, in the ANN Year-Month, we chose the optimal number of hidden neurons equal to 5. Figure 4 shows MAPE as a function of the number of hidden neurons in the ANN 12 Months. In this architecture, following a similar reasoning, we chose the optimal number of hidden neurons equal to 2.

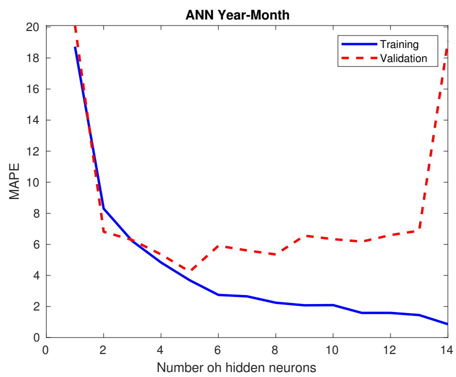

Figure 3. ANN Year-Month: MAPE as a function of the number of hidden neurons.

Since the data are monthly counts, the chosen SSA window length is $L = 36$ (a multiple of 12). The eigenvalues (λ_i, for $i = 1, \dots, 15$) are shown in Figure 5 and we can see several plateaux, which may correspond to sinusoidal waves. This is confirmed by Figure 6, where we can see 6 almost regular polygons or stars (right panel) corresponding to the paired eigentriples (2-3, 4-5, 6-7, 9-10, 10-11, 13-14). Therefore in the grouping step, the inspection of Figures 5 and 6 lead us to conclude that the eigentriples 1, 8, 12 and 15 correspond to the trend and the eigentriples 2-7, 9, 11, 13 and 14 represent the seasonal component. Each component is forecast independently by using the bootstrap with recurrent algorithm and then aggregated (further details can be requested to the authors).

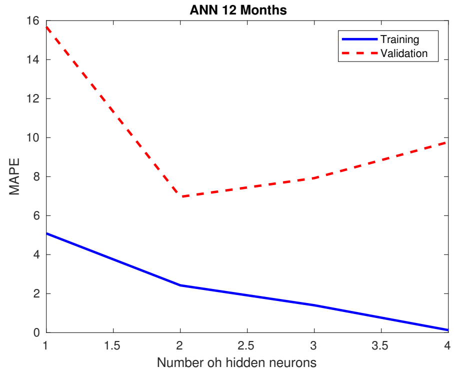

Figure 4. ANN 12 Months: MAPE as a function of the number of hidden neurons.

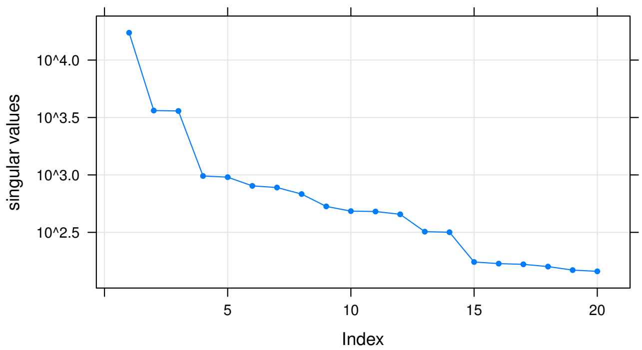

Figure 5. The first 20 eigenvalues of the monthly overnight stays in the North Region of Portugal ($L = 36$).

For the SARIMA specification, the `auto.arima()` function fits a SARIMA$(2,1,2)(0,1,1)_{12}$ model to the training set given by

$$(1 - a_1 B - a_2 B^2)(1 - B^{12})(1 - B)y_t = (1 + b_1 B + b_2 B^2)(1 + \beta_1 B^{12})e_t,$$

where $\hat{\sigma}_e^2 = 645.8$, and the parameters estimates are $(\hat{a}_1, \hat{a}_2, \hat{b}_1, \hat{b}_2, \hat{\beta}_1) = (0.2333, -0.6479, -0.7110, 0.9591, -0.3093)$ with s.e. $(0.1120, 0.1211, 0.0895, 0.0885, 0.1218)$, respectively.

The exponential smoothing method, through the `forecast` function, supplies a ETS(M,Ad,M) which represents a damped trend with multiplicative seasonal component and multiplicative errors (also called a damped multiplicative Holt-Winters' method), defined by [7] as:

$$\begin{array}{ll} \hat{y}_{t+h|t} = (l_t + \phi_h b_t) s_{t-m+h_m^+}, & \text{(Forecasting equation)} \\ l_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(l_{t-1} + \phi b_{t-1}), & \text{(Level equation)} \\ b_t = \beta^*(l_t - l_{t-1}) + (1-\beta^*)\phi b_{t-1}, & \text{(Trend equation)} \\ s_t = \gamma y_t/(l_{t-1} + \phi b_{t-1}) + (1-\gamma) s_{t-m}, & \text{(Seasonal equation)} \end{array},$$

where $m = 12$ denotes the frequency, $h_m^+ = [(h-1) \mod m] + 1$, $\phi_h = \phi + \phi^2 + \cdots + \phi^h$, and the parameter estimates are $\hat{\alpha} = 0.519, \hat{\beta}^* = 0.0086, \hat{\gamma} = 1e-04, \hat{\phi} = 0.9773$ and $\hat{\sigma} = 0.0504$. The `forecast` function also returns the estimates for the initial states l_0, b_0, s_{-i}, for $i = 0, \ldots, 11$.

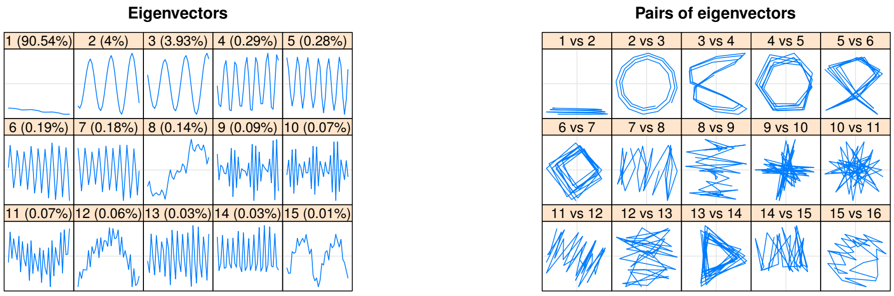

Figure 6. The first 15 eigenvectors (left panel) and scatterplots for eigenvector pairs (right panel) of the monthly overnight stays in the North Region of Portugal ($L = 36$).

The forecasting obtained by the several models/methods are shown in Figure 7. As can be seen in the figure, the forecast points obtained by SARIMA model, exponential smoothing method, seasonal naïve method and ANN Year-Month are less than the observations, while for SSA and ANN 12 Month some forecast points are greater than the observation and other are less than the observation. Their accuracy is evaluated by calculating MAPE and RMSE, shown in Table 2. As can be seen, the SSA forecast is the best in terms of these two accuracy measures. Next, we have the ANN 12 Months and exponential smoothing forecasts. The worst results were obtained with the seasonal naïve approach, followed by the ANN Year-Month and SARIMA models.

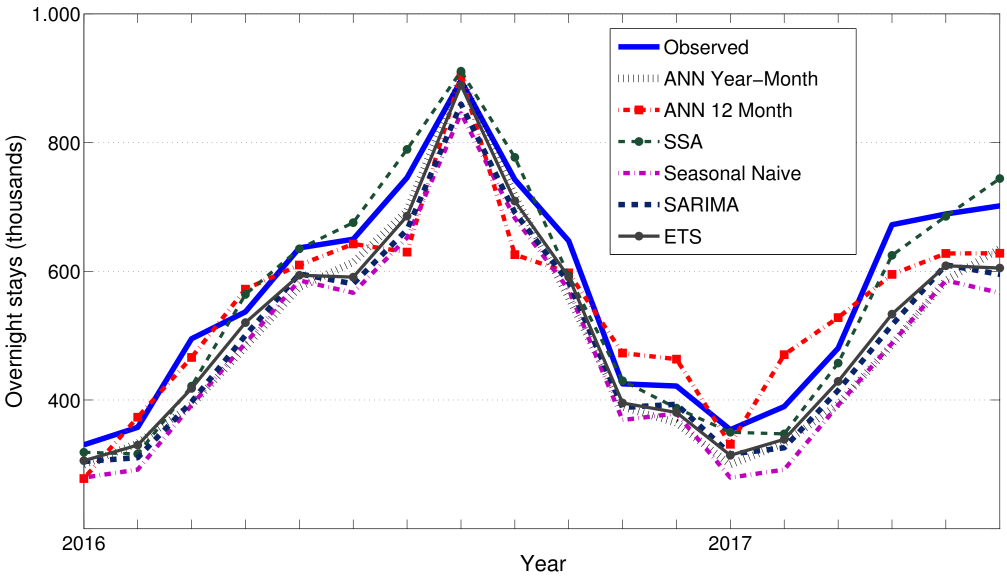

Figure 7. Forecasting comparison.

Note that the previous work which considered more recent data (until 2015) is [15]. In that study, the author presented the forecasting results of a multilayer feedforward neural network with the year and month as inputs, *i.e.*, a network similar to our first architecture. The optimal number of hidden neurons was equal to 8 and it was chosen following the same procedure as we followed. The author got a MAPE of 5.26% on test data corresponding to the year 2015. Here, we got 11.62% (more than double) on test

Table 2. Forecasting accuracy comparison.

Method/Model	MAPE	RMSE
ANN Year-Month	11.62	74.30
ANN 12 Months	9.26	59.53
SSA bootstrap	5.70	36.98
Seasonal naïve	15.11	88.45
SARIMA model	11.24	70.20
Exponential smoothing	9.42	60.22

data corresponding to the year 2016 and first semester of 2017. Therefore, the same neural network architecture seems to be inadequate to forecast more recent data. On the contrary, the SSA approach presented here was able to provide a good forecast, with a MAPE of 5.70%, close to 5.26%.

5. Conclusions and Future Work

In this work, different methodologies were developed and used to predict the time series of the monthly overnight stays in the North Region of Portugal. The forecasts given by neural networks, singular spectrum analysis, the seasonal naïve method, seasonal autoregressive integrated moving average models and exponential smoothing models were compared. Two forecasting accuracy measures were considered and singular spectrum analysis showed the best results in both of them.

In the last years, the time series of the monthly overnight stays in the North Region of Portugal registered an uncommon increase in its values and, as a consequence, several results presented in previous works are no longer valid. In particular, neural network architectures exhibiting a good performance in those works showed here a poor performance, with a forecasting error which doubled in more recent data.

There are several contributions of this paper to the literature on monthly overnight stays forecasting. Firstly, we show that the singular spectrum analysis is a reliable and robust technique for forecasting this type of data, broadening the class of techniques and methods available for monthly overnight stays forecasting. Furthermore, for the data considered in this work, singular spectrum analysis clearly outperforms SARIMA and exponential smoothing models as well as some ANN architectures, which are methods traditionally used for forecasting of the tourism demand. Finally, SSA is a technique which is able to cope well with the seasonality and trend presented in the analysed data, allowing the analysis and forecasting of each component separately.

In the future, we plan to develop forecasting models of monthly overnight stays for the different types of accommodations that exist in the North Region of Portugal, namely hotels, guest-houses or hostels, and tourist apartments. Furthermore, we intend to include economic and marketing factors as explicative variables, when possible, in the different methodologies used in this work in order to improve their forecasting performance, specially for neural networks, since we believe that systematic political and promotional policy changes can explain the behaviour of the analysed dataset in the last few years.

Acknowledgements

This work was partially supported by the Center for Research and Development in Mathematics and Applications (CIDMA) through the Portuguese Foundation for Science and Technology (FCT - Fundação para a Ciência e a Tecnologia), references UIDB/04106/2020 and UIDP/04106/2020. The authors also thank to the anonymous referees for their comments.

References

[1] P. Fernandes, J.P. Teixeira, J. Ferreira and S. Azevedo, *Modelling tourism demand: A comparative study between artificial neural networks and the Box-Jenkins methodology*, Rom. J. Econ. Forecast. 5 (2008), pp. 30–50.

[2] N. Golyandina and A. Korobeynikov, *Basic Singular Spectrum Analysis and forecasting with R*, Comput. Stat. Data Anal. 71 (2014), pp. 934–954.

[3] N. Golyandina, A. Korobeynikov and A. Zhigljavsky, *Singular Spectrum Analysis with R*, Springer-Verlag Berlin Heidelberg, 2018.

[4] N. Golyandina and A. Zhigljavsky, *Singular Spectrum Analysis for Time Series*, Springer-Verlag Berlin Heidelberg, 2013.

[5] H. Hassani, A. Webster, E.S. Silva and S. Heravi, *Forecasting U.S. Tourist arrivals using optimal Singular Spectrum Analysis*, Tourism Management 46 (2015), pp. 322–335.

[6] S. Haykin, *Neural networks and learning machines*, 3rd ed., Pearson Education, Inc., New Jersey, 2009.

[7] R.J. Hyndman, A.B. Koehler, J.K. Ord and R.D. Snyder, *Forecasting with Exponential Smoothing: The State Space Approach*, Berlin, Springer-Verlag, 2008.

[8] R.J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed., OTexts, 2018.

[9] R.J. Hyndman and Y. Khandakar, *Automatic time series forecasting: The forecast package for R*, J. Stat. Softw. 27 (2008), pp 1–22.

[10] S.S. Rao, *Engineering Optimization: Theory and Practice*, 4th ed., John Wiley & Sons, Inc., New Jersey, 2009.

[11] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, 2008. Software available at http://www.R-project.org

[12] N. Santos, P. Fernandes and J.P. Teixeira, *A comparison of linear and non linear models to forecast the tourism demand in the North of Portugal*, Ciencias Administrativas. Teoría y Praxis (2014), pp. 91–104.

[13] A. Saayman and I. Botha, *Non-linear models for tourism demand forecasting*, Tourism Econ. 23 (2017), pp. 594–613.

[14] R.H. Shumway and D.S. Stoffer, *Time Series Analysis and Its Applications - With R Examples*, 4th ed., Springer International Publishing, 2017.

[15] J. Silva, *Previsão das dormidas mensais nos alojamentos turísticos da região Norte de Portugal*, MSc thesis, Universidade Lusófona do Porto, 2017.

[16] J.P. Teixeira and P. Fernandes, *Tourism time series forecast*, in *Improving Organizational Effectiveness with Enterprise Information Systems*, J.E. Varajão et al., eds., IGI Global, Hershey, 2015, pp. 72–87.

[17] J.P. Teixeira and P. Fernandes, *Tourism time series forecast: different ANN architectures with time index input*, Proc. Technol. 5 (2012), pp. 445–454.