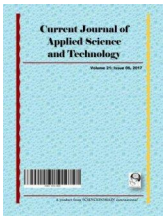

Current Journal of Applied Science and Technology

30(2): 1-12, 2018; Article no.CJAST.44836
ISSN: 2457-1024
(Past name: British Journal of Applied Science & Technology, Past ISSN: 2231-0843, NLM ID: 101664541)

Performance Analysis of Bagging Feed-Forward Neural Network for Forecasting Building Energy Demand

E. Juan Zarate Perez[1*], Mariana Palumbo Fernández[2] and Ana Lúcia Torres Seroa da Motta[1]

[1]Federal Fluminense University, Rua Passos da Patria 156, Niterói/RJ – 24210330, Brazil.
[2]Universitat Rovira i Virgili, Avinguda dels Països Catalans 26, Tarragona/CT – 43007, Spain.

Authors' contributions

This work was carried out in collaboration between all authors. Author EJZP designed the study and managed the revision with author MPF. Author ALTSM supervised the work and performed the first numerical analysis. Author EJZP continued the study with further numerical and statistical analysis and wrote the first draft of the manuscript. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/CJAST/2018/44836
Editor(s):
(1) Dr. Sylwia Myszograj, Professor, Department of Water Technology, Sewage and Wastes, University of Zielona Gora, Poland.
Reviewers:
(1) Lifeng Wu, Nanjing University of Aeronautics and Astronautics, China.
(2) Irshad Ullah, Pakistan.
(3) Qiang Lai, East China Jiaotong University, China.
Complete Peer review History: http://www.sciencedomain.org/review-history/26900

Original Research Article

Received 12 July 2018
Accepted 22 October 2018
Published 29 October 2018

ABSTRACT

Forecast models play a fundamental role in anticipating the effects of the energy demand in buildings to addressing the energy crisis. A forecast model for anticipating from one to three days every 30 min of the building energy demand is presented. In this model, a feed-forward artificial neural network (ANN) is combined with bootstrap aggregation techniques, using a Box–Cox transformation, seasonal and trend decomposition using loess, and a moving block bootstrap technique. An analysis was conducted using the data provided by a building's energy demand; the data were collected during a period of four months, with readings every 10 s and averages of the values obtained every 30 min. The feed-forward neural-network method combined with bootstrap aggregation techniques consistently outperformed the forecasting accuracy of the original feed-forward neural network through cross-validation in the root mean square error (RMSE) and the mean absolute percentage error. From cross-validation in-sample period, used for the initial

**Corresponding author: E-mail: ezarate@id.uff.br;*

parameter estimation and model selection, it is concluded that a feed-forward neural network with the original data gives a slightly lower RMSE compared with the data generated by bootstrapped versions. However, in the cross-validation out-of-sample period used to evaluate forecasting performance, it has better consistency throughout all horizons for the ANN model combined with bootstrap aggregation techniques than for the ANN original model. The results are statistically significant according to the Ljung–Box test, which verifies that the forecast errors are not correlated and validates the proposed model.

Keywords: Feed-forward; neural networks; energy demand forecasting; ANN; Bootstrap; MBB.

1. INTRODUCTION

Environmental concerns, such as global warming and climate change, together with other factors, such as the growing demand for electricity and the adoption of the precept of sustainable development, encourage the process of reducing the use of fossil fuels [1]. The growth in electricity demand is especially relevant in the residential sector, which currently consumes approximately 40% of global energy resources and generates about one-third of greenhouse gas emissions [2]. In this sector, there is also a marked variability in demand resulting from variations in daily and seasonal environmental conditions, which generates problems of simultaneity between demand and energy production [3].

Therefore, modelling and forecasting the energy consumption of buildings will play a fundamental role if urban areas are to reduce their overall energy consumption. Accurate modelling and forecasting of building energy demand enable numerous energy management and efficiency applications, such as demand response (DR) programs [4], urban energy infrastructure planning, estimating improvements to building energy performance, informing early-stage design decisions, and optimising building heating, ventilation, and air conditioning systems [5].

Conventionally, building-scale electricity demand appraisals have been made using engineering software packages based on data processing of structural, geometric, and material building properties. The difficulty in reaching and validating such information is a problem for wide-scale energy forecasting [5]. In response, there is a growing interest in statistical and machine-learning techniques that have proved to be more accurate and fast approaches and that forgo the demanding input requirements of theoretical formulations in favour of a practical set of historically recorded time series energy data [6].

Most of the works found in the literature have in common that they seek an economic goal. However, the incentives and economic sanctions established by the network operators reflect technical issues, such as line congestion or network stability. A recent, systematic review that identified 50 different methods between 1985 and June 2017 was presented by Kuster et al. [7], who identified three main approaches in terms of noteworthiness: time series models, regression-based formulations, and artificial neural networks (ANNs).

ANN methods have been widely used to forecast electric demand, by virtue of their superior performance. These models allow complex nonlinear relationships between the response variable and its predictors [8]. The current disadvantage is that data collection is deepened, which creates a broader data record, with a greater particularity level and, consequently, a higher noise level [9]. The noise level influences the ANN learning process, decreasing the capacity for generalisation and causing an excess of training (overfitting). However, this effect can be reduced by stabilising the variance and previous smoothing of the time series with bootstrap techniques [10].

In this article, the performance analysis of a univariate autoregressive model based on ANN structure to forecast building energy demand is presented, and the effect of variance stabilisation and previous smoothing of the time series is demonstrated in the results. The objective is to appraise the performance of the model to reduce forecasting errors, leading to more real and high-performance solutions for decision making.

In view of the combinatorial essence of the problem, the technical implications are evaluated using a neural network autoregression (NNAR) model, which is a feed-forward neural network with one hidden layer. As previously mentioned, it is integrated with bootstrap techniques that make it possible to generate a new time series

similar to the observed series, which it will use in the training of the NNAR model.

2. METHODOLOGY

Next, the different points of the research methodology applied are described. Initially, a Box–Cox transformation is made to the historical data, followed by seasonal and trend decomposition using loess (STL). The remaining component is initialised using the moving block bootstrap (MBB). After bootstrapping the remainder, the trend and seasonality are combined with the bootstrapped remainder, and the Box–Cox transformation is inverted. For each horizon, the final resulting forecast is calculated from the mean of the forecasts from the single models of the neural network. Fig. 1 gives an illustration of a methodology sequence.

2.1 Data Collection

This study uses the data of the building energy demand (kW) collected in the state of Rio de Janeiro, Brazil (Fig. 2). The data were collected during a period of four months (the first four months of 2017), with readings every 10 s and averages of the values obtained every 30 min. The collected data were initially stored in the measuring device and subsequently transferred to a computer.

The data record was made with the EnergyLOG *plus* meter, which issued monitoring and indicating the quality and consumption of electrical energy for both residential and commercial applications. This device stores the measured values of the electrical network for periods configured by the user. It uses the true RMS method to measure the active, reactive and apparent power. A true-RMS measurement can accurately measure both pure waves and the more complex non-sinusoidal waves.

2.2 Neural-network Models

Artificial Intelligence is an application speciality of computer science in almost all domains of science and technology. Given the great importance of the forecast models, artificial neural networks have become an active area of research [11]. Studies in the literature show models in relation to the ANN, such as for the model of the two-phase deviation factor of a gas condensate fluid [12], forecasting of oilfield scale formation [13], and application of neural networks for forecasting the workability of self-compacting concrete [14]. Evaporation from a reservoir in semiarid environments has been estimated using the ANN and climate-based models [15]. The modelling accuracy of the aerodynamic curve of a wind turbine has been improved using neural networks [16].

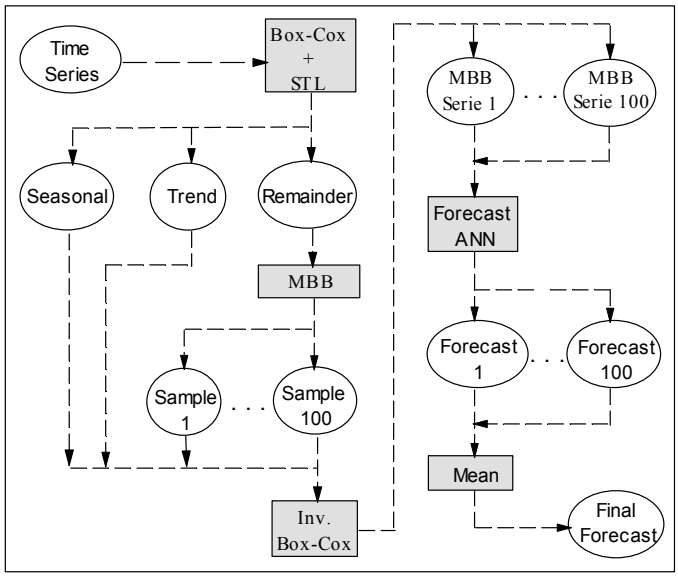

Fig. 1. Illustration of methodology sequence

This methodology combines ANN methods with bootstrap (MBB) techniques, with the purpose of comparing performance with the original ANN method presented in section 2.2

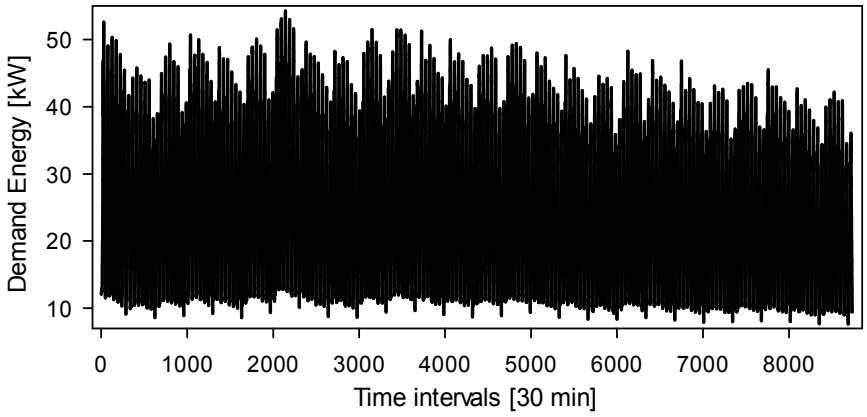

Fig. 2. Building energy demand data collected for in this study was used to evaluate the performance of the proposed forecast models

A study of the selection of input variables to the ANN for forecasting hospital inpatient flows [17] has been done. An artificial network was used for forecasting water uptake under shallow saline groundwater conditions [18]. An ANN model was developed for the comparative study of different general circulation models (GCMs) (a GCM is a type of climate model) for streamflow forecast [19]. Neural network models were used for forecasting the wellhead pressure–flow-rate relationship for Niger Delta oil wells [20]. In this work, a model related to the univariate feed-forward neural network is presented, under the hypothesis that this univariate model, when combined with bootstrap aggregation techniques, can obtain a better forecast performance of building energy demand.

Recent work has been presented on forecasting the load of daytime cooling energy for institutional buildings [18] using this univariate feed-forward neural network method, in which the results were optimal with regard to performance measurements. With respect to the bootstrap approach, there exist research studies of integration with exponential models and seasonal autoregressive integrated moving average [21,22], where the forecast performance was consistently improved.

2.2.1 Neural-network architecture

ANNs are forecasting methods that are based on simple mathematical models of the brain for information processing [8]. A neural network can be regarded as a network of neurons that are in neat layers. The inputs from the bottom layer and the outputs (forecasts) form the top layer. There may also be intermediate layers holding hidden neurons. The coefficients associated with these predictors are called “weights.” The weights are selected in the neural network setting using a learning algorithm. When adding an intermediate layer, the neural network becomes nonlinear.

That means it is a multilayer feed-forward network, in which each layer of nodes receives inputs from the preceding layers. Here, each layer of nodes receives inputs from the previous layers, while the outputs of the nodes in one layer are inputs for the next layer. The inputs for each node are combined using a weighted linear interaction. The result is then modified by a nonlinear function before being output. For example, the inputs into the hidden neuron j in Fig. 3 are combined linearly as expressed in Equation (1) [8].

$$z_j = b_j + \sum_{i=1}^{4} w_{i,j} x_i \quad (1)$$

The feed-forward neural network shown in Fig. 3 contains four predictors in the input layer and three neurons in the hidden layer. Thus, the parameters b_1, b_2, b_3, and $W_{1,1}$, ... , $W_{4,3}$ are found during the training stage, in which the network learns using the observed data [23]. The data used in this study for this purpose is shown in Fig. 2. The network is usually trained several times using different random starting points, and the results are averaged. The result of the linear combination is then modified by a nonlinear function before the output of each layer using a sigmoid. It is denoted in Equation (2) [8].

$$s_{(z)} = \frac{1}{1+e^{-z}} \quad (2)$$

2.2.2 Neural-network autoregression

With time series data, lagged values of the time series can be used as inputs to a neural network, expressed as an NNAR model. In this article, only the feed-forward neural networks with one hidden layer are considered, and it is denoted as NNAR (p, P, k) [m] to indicate that there are p nonseasonal lagged inputs, P seasonal lagged inputs, and k neurons in the hidden layer. [m] represents the frequency and predictors in the input layer shown in Equation (3).

$$y_{(t)} = (y_t, y_{t-1}, \cdots, y_{t-p}, y_{t-m}, y_{t-2m}, y_{t-Pm}) \quad (3)$$

The nnetar() function in R Core Team fits an NNAR(p, P, k)m model [24], and it is necessary to install the forecast package [25]. Here, when the values of p and P are not specified, they are selected automatically. For seasonal time series, the default values are P=1 and p is chosen from the optimal linear model fitted to the seasonally adjusted data, according to the AIC-Akaike information criterion. If k is not specified, it is set to (p+P+1)/2, approaching the nearest integer value. Because the data (Fig. 2) was obtained with readings from every 10 s and averages of the values every 30 min, the daily frequency [m] equals 48 [8]. Fig. 4 gives the forecast one day in advance, taking as inputs the first seven days of the data collection.

To perform forecasts of the example shown in Fig. 4, the steps described in Fig. 1 were used for developing the MBB-ANN model. The first seven days of the sample from Fig. 2 (336 records) were selected and a Box-Cox transformation was performed. Then, from the STL decomposition the seasonal, trend, and remainder components were obtained. The bootstrap aggregation technique (MBB) was applied to the remainder component,

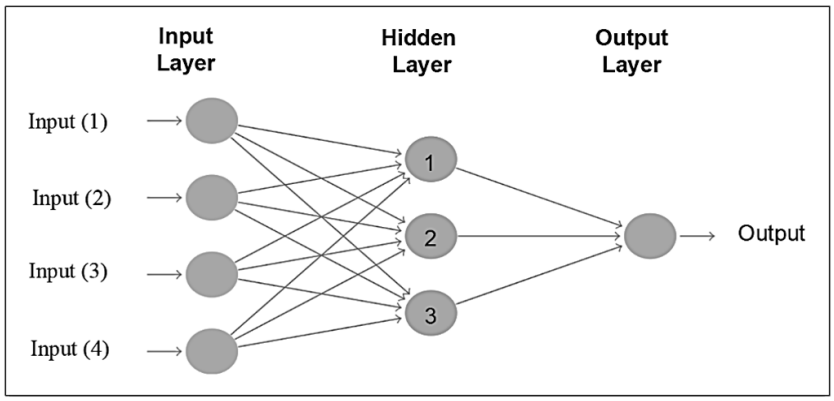

Fig. 3. Multilayer feed-forward network
Source: Adapted from [8]

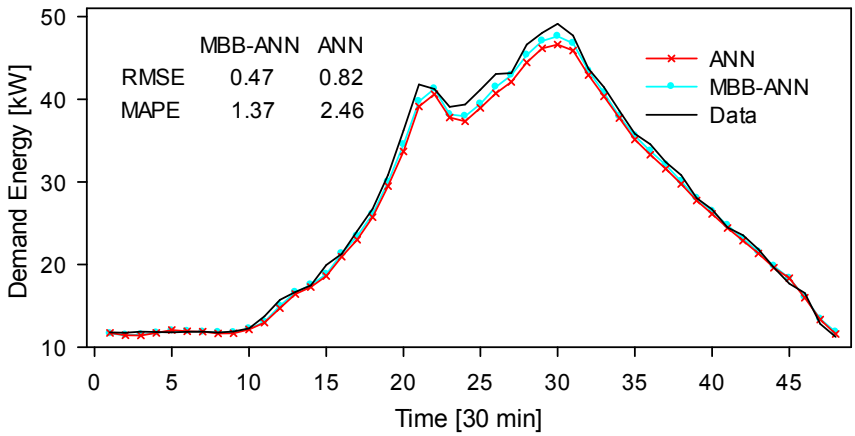

Fig. 4. Forecast one day in advance

obtaining 100 bootstrap samples and a Box-Cox transformation was made to them. Each of the inverted bootstrap series was used to train ANN models and perform 48-point forecasts (one day) with each of them (as described in Equations (1), (2), and (3)). Finally, the final forecast was obtained from the mean of the hundred models.

For the original ANN model, only the sequence developed based on equations (1), (2) and (3) was followed. Methodology to Box-Cox transformation, STL decomposition, and MBB techniques are presented in section 2.4, 2.5, and 2.6 respectively.

2.3 Estimating Forecasting Error and Cross-Validation

A forecast error is a difference between an observed value and its forecast; this means the unpredictable part of observation [8]. It is denoted in Equation (4).

$$e_t = \hat{y}_{t-} y_t \tag{4}$$

The two most commonly used scale-dependent measures are based on absolute errors or square errors and are given by the mean absolute percentage error (MAPE) and root means square error (RMSE), respectively. These are denoted in Equations (5) and (6).

$$MAPE = \left(\sum_{t=T+1}^{T+h} \left| \frac{e_t}{y_t} \right| / h \right) x\ 100\% \tag{5}$$

$$RMSE = \sqrt{\sum_{t=T+1}^{T=h} (e_t)^2 / h} \tag{6}$$

One of the most robust standard procedures performed for model evaluation in regression is K-fold cross-validation (CV). When machine-learning methods are used for the forecast, CV is appropriate to control overfitting the data. CV can appropriately control overfitting in this enforcement. If the models underfit the data and lead to strongly correlated errors, the CV method to be prevented as in equal a case they may output an underestimation of the error. However, this occurrence can be easily perceived by checking the residuals for serial correlation utilizing the Ljung–Box test [26]. The Ljung–Box test is implemented in R in the Box.test function.

2.4 The Box–Cox Transformation

A useful family of transformations that includes both logarithms and power transformations is the family of Box–Cox transformations, which are determined by the parameter λ [8]. These are commonly used transformations for stabilizing the variance of a time series and were originally proposed by Bergmeir et al. [27]. The transformation is defined as follows.

$$\omega_t = \begin{cases} Log(y_t), & \lambda=0; \\ \left(Y_t^{\lambda} - 1 \right) / \lambda, & \lambda \neq 0. \end{cases} \tag{7}$$

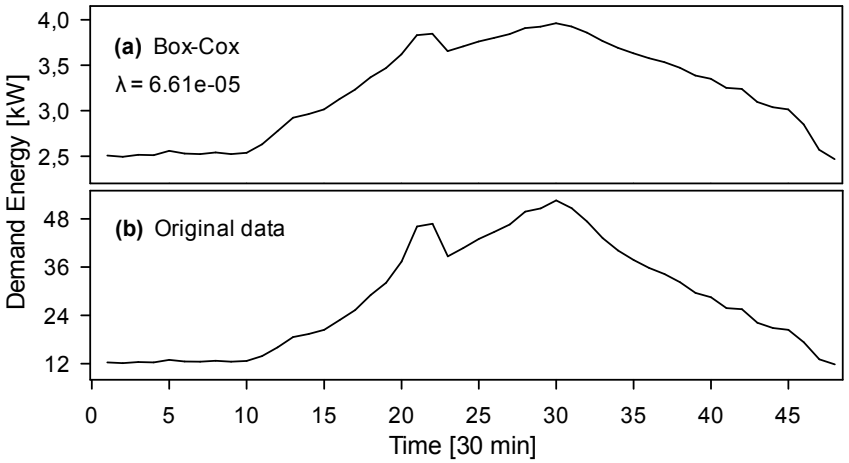

Fig. 5. Example of Box-Cox transformation

In Fig.5 (a) shows the Box-Cox transformation with λ = 6.61e-05 of the Fig. 5 (b), which corresponds to the first 48 records (1 day) taken from the collection of data shown in the Fig. 2

Depending on the parameter λ, the transformation is basically the identity (λ = 1), the logarithm (λ = 0), or a transformation somewhere between. To choose the parameter λ, it is restricted to the interval [0,1], and the Guerrero method is used to choose its value, as described elsewhere [22]. The series is divided into a subseries equal to the seasonality, or of length two if the series is not seasonal. Then, the sample standard deviation (s) and mean (m) are calculated for every subseries, and λ is picked so that the coefficient of variation of s/m(1−λ) through the subseries is optimized [22].

2.5 STL Decomposition

Time series data can show a diversity of patterns, and it is frequently fruitful to split a time series into different components, each representing a subjacent pattern category. STL is a versatile and robust method for decomposing time series [28]. In STL, loess is used to divide the time series into their trend, seasonal, and remainder components. The division is additive, summing the parts to assign the original series anew, as given in Equation (8) [8].

$$Y_t = S_t + T_t + R_t \quad (8)$$

where Y_t is the time series, St is the seasonal component, T_t is the trend-cycle component, and R_t is the remainder component, all at period t.

In detail, the steps applied across STL decomposition are [22]: (i) detrending, (ii) cycle–subseries smoothing, (iii) low-pass filtering of smoothed cycle–subseries, (iv) detrending of the seasonal series, (v) deseasonalizing the original series, using the seasonal component calculated in the preceding steps, and (vi) smoothing the deseasonalized series to obtain the trend component. In R, the STL algorithm is available by way of the stl () function. It uses it with its default parameters, i.e., the degrees for the loess fitting are d = 1 in steps (iii) and (iv), and d = 0 in step (ii). Fig. 6 gives an STL decomposition for the time series Box–Cox transformation presented in Fig. 5.

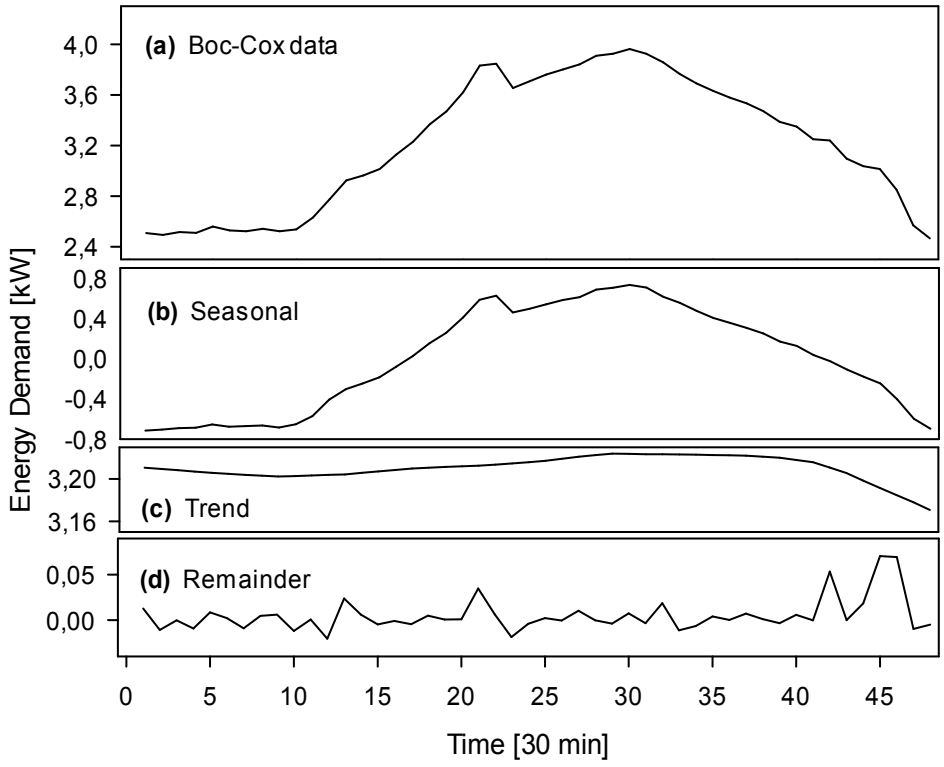

Fig. 6. Example of STL decomposition

Fig. 6 (a) corresponds to the same chart that is shown in Fig. 5 (a). Fig. 6 (b), (c) and (d) correspond to the components of STL decomposition from the data shown in Fig. 6 (a)

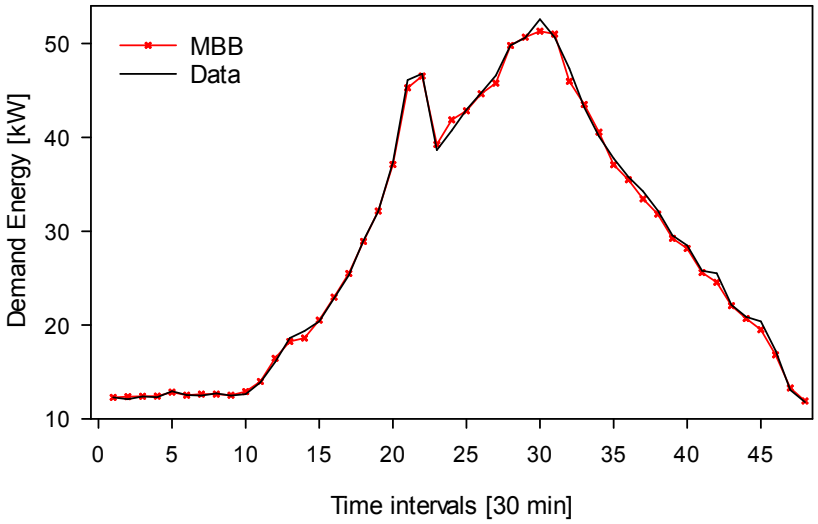

Fig. 7. Example of moving block bootstrap (MBB)

From the STL decomposition shown in Fig. 6, the MBB was applied to the remainder component, obtaining 100 bootstrap series, then, each bootstrap series was joined with the seasonal and trend components, the inverse Box-Cox transformation was made to each jointed series. Finally, it was made the mean of the 100 inverted series

2.6 Bootstrapping the Remainder

As the time series are typically autocorrelated, a requirement is a stationarity for the forecast, which is achieved by initialising the remainder of the STL decomposition by MBB technique. For this purpose, the same procedure described in previous work [22] is followed, where, for a series of length n, with a block size of l, [n / l] +2 blocks of the remaining series of an STL decomposition are indicated; then, discard a random number of values, between zero and (l − 1), from the beginning of the bootstrapped series.

Finally, to obtain a series with an identic length to that of the original series, discard just as many possible values as necessary to achieve the needed length. This procedure ensures that the bootstrapped series does not necessarily start or finish on a block boundary. The number of bootstrapped versions to generate is equal to 100 [22]. Fig. 7 gives an example corresponding to the first 48 records, which includes the previous steps of Box–Cox transformation and STL decomposition.

3. RESULTS AND DISCUSSION

The results discussed in this section are based on the architecture of the feed-forward neural network presented in Fig. 8. The inputs in the input layer $E_{(t)}$, $E_{(t-1)}$, $E_{(t-2)}$, ... , $E_{(t-n)}$, $E_{(t-p)}$, $E_{(t-m)}$,$E_{(t-2m)}$, $E_{(t-Pm)}$ are consecutive data of the building energy demand, and $E_{(t)+1}$ is the output layer or estimate of the forecast. H_1, H_2, H_3, ..., H_n are neurons in the hidden layer.

3.1 Forecasting One Day in Advance

Fig. 9 shows the result of the cross-validation one day in advance every half hour. For the first test, the data from the first seven days were used; for the second test, the data from the first eight days were used. It continued in the same way until day 119 of a total of 120 days of sample data. The next 48 values (one day) of every subseries were utilised to evaluate performance with the forecast data. For the cross-validation in-sample period used for the initial parameter estimation and model selection, the feed-forward neural network with the original data gives a slightly lower RMSE compared with the data generated by bootstrapped versions.

This difference may be because the bootstrap series generated is slightly smooth, giving rise to a slightly higher RMSE for the training of the model. However, the cross-validation out-of-sample period used to evaluate forecasting performance has better consistency throughout the horizon for the ANN model combined with bootstrap aggregation techniques than that for the ANN original model.

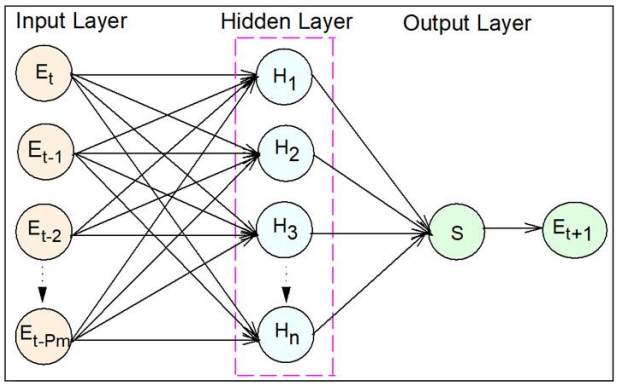

Fig. 8. Feed-forward Neural Networks architecture used in this paper

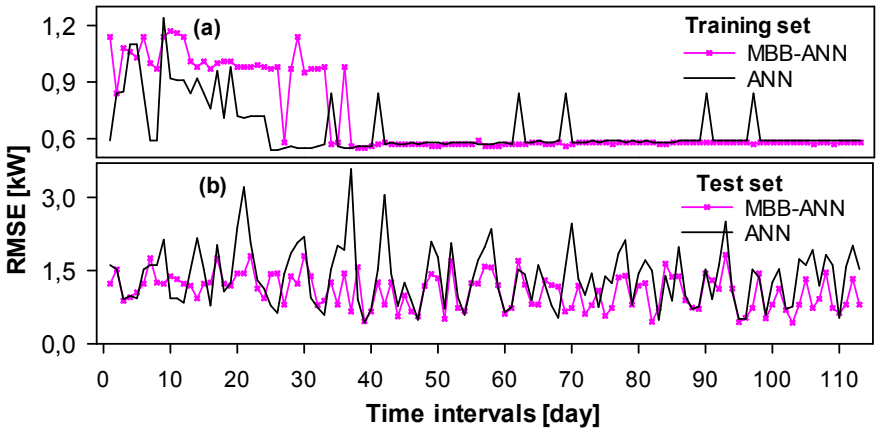

Fig. 9. Cross-validation of one-day forecast

Fig. 9 (a) shows the RMSE for the fitting and selection of the model, while Fig. 9 (b) shows the RMSE for the forecasts in each test of the cross-validation set, respectively. Subhead (MBB-ANN) corresponds to the neural network model combined with bootstrap techniques, and (ANN) corresponds to the original neural network model

3.2 Forecasting Two Days in Advance

For the first test, the first 15 days were utilised for training the neural network, and the first 16 days for the second test, successively, until completing the 118 days throughout the horizon of the sample data. The next 96 values (two days) of every subseries were utilised to evaluate performance with the forecast data. Fig. 10 shows that the ANN model combined with bootstrap aggregation techniques outperforms consistently the original ANN method.

3.3 Forecasting Three Days in Advance

The result of the cross-validation for three-day forecasts is shown in Fig. 11. For the first test, the first 21 days were utilized for training the neural network, and the first 22 days for the second test, successively, until completing the 117 days throughout the horizon of the sample data. The next 144 values (three days) of every subseries were utilised to evaluate performance with the forecast data. Of the cross-validation in-sample period for the initial parameter estimation and model selection, the feed-forward neural network with the original data gives an RMSE slightly smaller than the model of training by bootstrapped versions.

Nevertheless, as in the two previous cases, for the cross-validation out-of-sample period used to evaluate forecasting performance, the ANN model combined with bootstrap aggregation techniques has better consistency throughout the horizon than that for ANN original model, in this case, with a very significant difference.

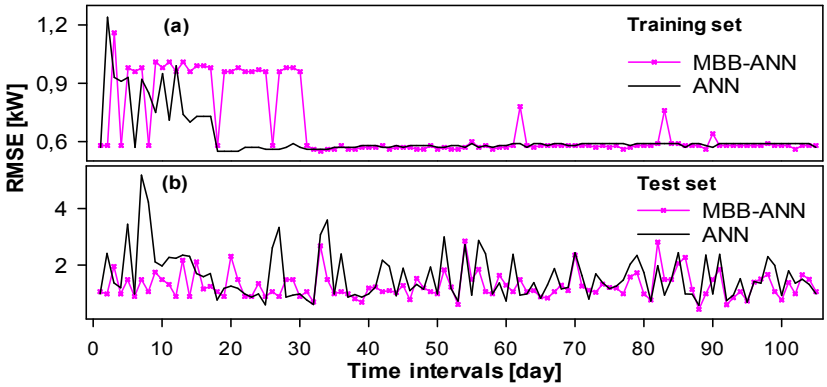

Fig. 10. Cross-validation of two-day forecast

Fig. 10 (a) shows the RMSE for the fitting and selection of the model, while Fig. 10 (b) shows the RMSE for the forecasts in each test of the cross-validation set, respectively. Subhead (MBB-ANN) corresponds to the neural network model combined with bootstrap techniques, and (ANN) corresponds to the original neural network model

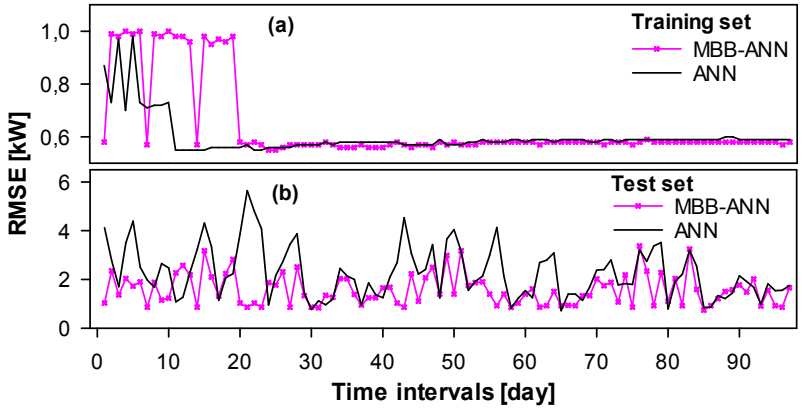

Fig. 11. Cross-validation of three-day forecast

Fig. 11 (a) shows the RMSE for the fitting and selection of the model, while Fig. 11 (b) shows the RMSE for the forecasts in each test of the cross-validation set, respectively. Subhead (MBB-ANN) corresponds to the neural network model combined with bootstrap techniques, and (ANN) corresponds to the original neural network model

Table 1. Results of the K-fold cross-validation and Ljung–Box test

Model	**MBB-ANN**						**ANN**					
Set	**Training**			**Test**			**Training**			**Test**		
Day forecast	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**	**1**	**2**	**3**
RMSE	0.71	0.69	0.64	**1.09**	**1.28**	**1.57**	0.65	0.62	0.60	1.37	1.58	2.27
MAPE	1.97	1.89	1.78	**3.38**	**3.76**	**4.55**	1.92	1.82	1.77	4.23	4.71	6.50
Ljung–Box test				**0.98**	**0.98**	**0.97**						

3.4 Mean of K-fold Cross-validation and Ljung–Box Test

Table 1 shows the mean results of the K-fold cross-validation and Ljung–Box test. For the cross-validation in-sample period used for the initial parameter estimation and model selection, the feed-forward neural network with the original data and those generated by bootstrapped versions have a similar value, slightly lower for the original data, possibly because of the smoothing obtained when using that bootstrap technique; however, the effect occurs for forecasts.

For the forecasts one day in advance, the RMSE was reduced from 1.37 to 1.09 (20% forecast error) because of the bootstrap techniques. For forecasts two days in advance, the ANN model combined with bootstrap techniques increased slightly the forecast error compared with the previous scenario (one day); however, the ANN original model that utilised the original data had a significant forecast error. For the forecast three days in advance, the ANN method that utilised the original data can be considered inconsistent; the opposite happens with the bootstrap techniques combined with the ANN model, where there was a slight increase in forecast error with each additional day added to the forecast.

Considering the superior performance of the results of the ANN model combined with bootstrap techniques, it can be concluded that they do not lead to strongly correlated errors and that the RMSE obtained is not the product of systematic underestimation of the error. The mean of the Ljung–Box test of all the subseries of the K-fold cross-validation for the bootstrap techniques combined with the ANN model has a value close to the unit for the three cases studied; thus, the Ljung–Box test shows that the results are statistically significant, verifying that the forecast errors are not correlated and validating the proposed model.

4. CONCLUSION

Artificial intelligence is used in this research to forecast the energy demand of the building. The performance of the feed-forward neural network model combined with bootstrap techniques was evaluated using the Box–Cox transformation, STL decomposition, and moving block bootstrap techniques.

The ANN method combined with bootstrap aggregation techniques consistently outperformed forecasting accuracy compared with the original ANN method in RMSE and MAPE measurement on forecasts from one to three days of anticipation. The results are shown to be statistically significant by the Ljung–Box test, which verifies that the forecast errors are not correlated and validates the proposed model.

ACKNOWLEDGEMENTS

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

COMPETING INTERESTS

Authors have declared that no competing interests exist.

REFERENCES

1. Rabbani M, Dolatkhah M. Integration of demand-side management programs and supply-side alternatives for decentralized energy planning: An analysis of energy import and export effects integration of demand-side management programs and supply-side alternatives. IGI Global. 2018; 252–270.
2. Chalal ML, Benachir M, White M, Shrahily R. Energy planning and forecasting approaches for supporting physical improvement strategies in the building sector: A review. Renew. Sustain. Energy Rev. 2016;64:761–776.
3. Yutaka Kuwabata Takigawa F, Cavalcante Fernandes R, Antonio Cardoso Aranha Neto E, Tenfen D, Taghori Sica E. Energy management by the consumer with photovoltaic generation: Brazilian Market. IEEE Lat. Am. Trans. 2016;14(5):2226–2232.
4. Shao Z, Chao F, Yang S-L, Zhou K-L. A review of the decomposition methodology for extracting and identifying the fluctuation characteristics in electricity demand forecasting. Renew. Sustain. Energy Rev. 2017;75:123–136.
5. Jain RK, Smith KM, Culligan PJ, Taylor JE. Forecasting energy consumption of multi-family residential buildings using support vector regression: Investigating the impact of temporal and spatial monitoring granularity on performance accuracy. Appl. Energy. 2014;123:168–178.
6. Deb C, Zhang F, Yang J, Lee SE, Shah KW. A review on time series forecasting techniques for building energy consumption. Renew. Sustain. Energy Rev. 2017;74:902–924.
7. Kuster C, Rezgui Y, Mourshed M. Electrical load forecasting models: A critical systematic review. Sustain. Cities Soc. 2017;35:257–270.
8. Hyndman RJ, Athanasopoulos G. Forecasting : Principles and practice; 2014. Available:https://otexts.org

9. Jian Zheng, Cencen Xu, Ziang Zhang, Xiaohua Li. Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network" in 2017 51st Annual Conference on Information Sciences and Systems (CISS). 2017;1–6.
10. Müller MR. Análise de desempenho da rede neural artificial ARTMAP fuzzy aplicada para previsão multi-step de cargas elétricas em diferentes níveis de agregação. Repositório Institucional UNESP; 2018.
11. Yadav A, Sahu K. Wind forecasting using artificial neural networks: A Survey And Taxonomy. Int. J. Res. Sci. Eng. 2017;3(2): 148–155.
12. Akinsete O, Omotosho A. Modeling of two-phase gas deviation factor for gas-condensate reservoir using artificial neural network. Adv. Res. 2018;14(1):1–8.
13. Falode O, Udomboso C, Ebere F. Prediction of oilfield scale formation using Artificial Neural Network (ANN). Adv. Res. 2016;7(6):1–13.
14. Mazloom M. Application of neural networks for predicting the workability of self-compacting concrete. J. Sci. Res. Reports. 2013;2(1):429–442.
15. Benzaghta M. Estimation of evaporation from a reservoir in semi arid environments using artificial neural network and climate based models. Br. J. Appl. Sci. Technol. 2014;4(24):3501–3518.
16. Bustan D, Moodi H. Improving modelling accuracy of aerodynamic curve of a wind turbine using neural networks. J. Sci. Res. Reports. 2017;16(3):1–9.
17. Rasouli S, Tabesh H, Etminani K. A study of input variable selection to artificial neural network for predicting hospital inpatient flows. Br. J. Appl. Sci. Technol. 2016;18(4):1–8.
18. Ghamarnia H, Jalili Z. Artificial network for predicting water uptake under shallow saline ground water conditions. J. Sci. Res. Reports. 2015;7(5):359–372.
19. Patil M, Lal D, Karwariya S, Bhattacharya R, Behera N. Comparative study of different gcm models for stream flow prediction. Curr. J. Appl. Sci. Technol. 2018;26(5):1–12.
20. Okon A, Appah D. Neural network models for predicting wellhead pressure-flow rate relationship for niger delta oil wells. J. Sci. Res. Reports. 2016;12(1):1–14.
21. Deb C, Eang LS, Yang J, Santamouris M. Forecasting diurnal cooling energy load for institutional buildings using artificial neural networks. Energy Build. 2016;121:284–297.
22. Bergmeir C, Hyndman RJ, Benítez JM. Bagging exponential smoothing methods using STL decomposition and Box–Cox transformation. Int. J. Forecast. 2016; 32(2):303–312.
23. Box GEP, Cox DR. An analysis of transformations. Journal of the Royal Statistical Society. Series B (Methodological). Wiley Royal Statistical Society. 1964;26:211–252.
24. Pallab Kumar Datta. An artificial neural network approach for short-term wind speed forecast. Kansas State University; 2018.
25. R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria; 2018.
26. Bergmeir C, Hyndman RJ, Koo B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. Comput. Stat. Data Anal. 2018; 120:70–83.
27. Bergmeir C, Hyndman RJ, Benítez JM. Bagging exponential smoothing methods using STL decomposition and Box-Cox transformation. Int. J. Forecast; 2014.
28. Robert B. Cleveland, William S. Cleveland, Jean E. McRae. STL: A seasonal-trend decomposition procedure based on loess. J. Off. Stat. 1990;6(1):3–73.

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sciencedomain.org/review-history/26900