Comparative Analysis of Deseasonalization and Detrending Methods in Energy Consumption Forecasting

Tainá de Souza Coimbra
School of Electrical and Computer Engineering (FEEC)
Universidade Estadual de Campinas (UNICAMP)
Campinas, Brazil
tainacoimbra1996@gmail.com

Levy Boccato
School of Electrical and Computer Engineering (FEEC)
Universidade Estadual de Campinas (UNICAMP)
Campinas, Brazil
https://orcid.org/0000-0001-9319-9829

Hugo Valadares Siqueira
Department of Electrical Engineering
Federal University of Technology - Parana
Ponta Grossa, Brazil
hugosiqueira@utfpr.edu.br
https://orcid.org/0000-0002-1278-4602

Romis Attux
School of Electrical and Computer Engineering (FEEC)
Universidade Estadual de Campinas (UNICAMP)
Campinas, Brazil
https://orcid.org/0000-0002-2961-4044

***Abstract*—Energy consumption forecasting is a valuable tool for management decision-making that can lower expenses and enhance efficiency in a power system. Several time series prediction models can address this issue, ranging from statistical models to complex neural networks. With the purpose of increasing prediction accuracy, we tackle the problem of time series stationarity by considering an Autoregressive (AR) and Multilayer Perceptron (MLP) model. This paper presents a comparative analysis of both models regarding different deseasonalization and detrending methods, which converts the time series to stationary. This is a necessary condition for linear statistical models such as Autoregressive (AR). Our focus is to investigate whether it also improves neural network performance and which stationarity method produces the best result. For this study, we address an energy consumption series from the southeast region of Brazil. The computational results reveal that removing the trend by differencing and removing seasonality by normalization leads to the lowest errors.**

***Index Terms*—Energy consumption prediction, Auto-regressive Model, Neural Networks, Time Series, Stationarity, Seasonality, Trend.**

I. Introduction

Energy consumption and demand time series prediction are tasks associated with multiple applications. For instance, monitoring and forecasting energy consumption over time can help preventing energy deficits, while industries can use it to secure the best energy contracts with powerhouses. To accomplish prediction tasks, we can employ a number of approaches, including linear statistical methods, nonlinear filters and neural networks (including deep learning paradigms) [1]–[4].

Classical linear models, which are commonly understood under the aegis of the Box & Jenkins methodology, include: autoregressive (AR), moving average (MA), autoregressive and moving average (ARMA) models; autoregressive integrated moving average (ARIMA) models [5] and variants such as seasonal autoregressive integrated moving average (SARIMA) and seasonal autoregressive integrated moving average with exogenous factors (SARIMAX) models; and the Holt-Winters approach [6], [7]. Linear methods for time series forecasting are widely used in the literature, and can even outperform more complex machine learning models in some cases [8], [9].

However, a clear drawback is that they do not encompass nonlinear relationships. To address this limitation, we may resort to neural network models, such as Multilayer Perceptron (MLP), Extreme Learning Machines (ELM). Additionally, deep learning models such as Temporal Convolutional Neural Networks (TCN) and Long-Short Term Memory (LSTM) are also utilized to capture complex temporal patterns [10]–[14].

One requirement for employing linear models is stationarity, a condition that can be reached or approximated by eliminating the trend and seasonal components from the time series. This process can be done through various techniques, such as differencing, curve fitting, moving average and normalization [13], [15]. Stationarity can also be desirable when nonlinear models (e.g., neural networks) are employed [16], [17].

The problems of energy consumption and demand prediction have been addressed in a number of works in the literature, but a systematic analysis of the impact of deseasoning and detrending in their context should be investigated. Therefore the roles these procedures play in the performance of a representative nonlinear predictor, a Multilayer Perceptron (MLP), are compared to an Autoregressive (AR) as baseline.

The structure of this paper is as follows: section II brings an explanation of the key concepts regarding time series; sections III and IV describe the methodology, elucidating the models and their respective parameters, as well as the deseasolization and detrending methods employed, along with the evaluation metrics; finally, section V presents and discusses the results, while section VI summarizes our conclusions and perspectives.

II. Main Concepts

A. Stationarity

A time series can be seen as a sample of a discrete-time random process, which is composed of a sequence of random variables $\{x_{t_k}, k > 0\}$. The process is said to be *strict-sense stationary* if the following equality holds for all $k \in \mathbb{N}^+$, and for all $\Delta \in \mathbb{Z}$:

$$f_{x_{t_1},...,x_{t_k}}(x_1, ..., x_k) = f_{x_{t_1+\Delta},...,x_{t_k+\Delta}}(x_1, ..., x_k) \quad (1)$$

In practical scenarios, it is often very stringent to ensure that a general joint distribution be invariant. When it is possible to deal exclusively with first- and second-order statistics, *wide-sense stationarity* (WSS) can be used as an alternative. WSS is based on the following conditions:

$$E\{x_t\} = \mu \quad (2)$$

$$E\{x_{t+\tau}\ x_t^*\} = \gamma(\tau) \quad (3)$$

where μ is a constant value is $\gamma(.)$ is the autocorrelation function [5], [18], [19].

According to (2) and (3), the statistical mean of the random process must remain constant over time, and the autocorrelation must depend only on the time difference between points. From (3), it is possible to state that the variance of the process must also be constant. Assuming that the process is also *ergodic* [18], we can employ a time series (i.e. a single realization of the underlying stochastic process) to estimate the required statistics.

B. Autocorrelation Function (ACF)

The objective of the autocorrelation function is to measure the correlation between a time series and a delayed version thereof. It helps to identify any significant patterns or relationships between past observations and future values within the time series data. The normalized autocorrelation at lag k is defined by Equation (4) [5]:

$$\rho_k = \frac{E[(x_t - \mu)(x_{t+k} - \mu)]}{\sqrt{E[(x_t - \mu)^2]E[(x_{t+k} - \mu)^2]}} \quad (4)$$

C. Partial Autocorrelation Function (PACF)

The partial autocorrelation function (PACF) provides valuable insights into determining the order of a linear model, especially the AR. By analyzing the PACF, we can identify the direct influence of each lag on the current observation, allowing us to determine the appropriate lag order for autoregressive (AR) models.

Partial autocorrelations can be described as functions of the autocorrelations as in Equation 5. Consider ϕ_{kj} the j-th coefficient in an autoregressive representation of order k:

$$\rho_j = \phi_{k1}\rho_{j-1} + ... + \phi_{k(k-1)}\rho_{j-k+1} + \phi_{kk}\rho_{j-k} \quad j = 1, 2, ..., k \quad (5)$$

The last coefficient ϕ_{kk} is called the partial autocorrelation function of a process $\{z_t\}$ at lag k. Actually, ϕ_{kj} from Equation 5 are the regression coefficients of the linear regression z_t on $z_{t-1}...z_{t-k}$ [5]. Therefore, the partial autocorrelation can be defined as the correlation between the residuals from two linear regressions as Equation 6:

$$\phi_{kk} = corr[z_t - \hat{z}_t, z_{t-k} - \hat{z}_{t-k}] \quad (6)$$

where $\hat{z}_t$ is the best linear predictor of z_t based on its past coefficients $z_{t-1}...z_{t-k}$.

D. Decomposition

A time series decomposition can be additive or multiplicative, depending on the statistical nature of the data [6]. Considering the energy consumption time series x_t to be additive, it can be written as a sum of three components, as shown in Equation (7):

$$x_t = \tilde{m}_t + \tilde{s}_t + a_t \quad (7)$$

where $\tilde{m}_t$ represents the trend, which is related to the long term dynamics, $\tilde{s}_t$ is the seasonal component, which describes patterns that repeat periodically over time, and a_t is the noise component. The variable a_t can be considered approximately stationary, while $\tilde{m}_t$ and $\tilde{s}_t$ are non-stationary components, as they cause the mean and variance to vary over time [5].

III. Models

A. Autoregressive (AR) Model

A given process x_t is called an auto-regressive process of order p, abbreviated as AR(p), if [5]:

$$x_t = \phi_1 x_{t-1} + ... + \phi_p x_{t-p} + e_t \quad (8)$$

where e_t is a Gaussian white noise which represents the estimation error and ϕ_i are constant coefficients.

B. Multilayer Perceptron (MLP)

A Multilayer Perceptron (MLP) performs a nonlinear mapping of a set of inputs by means of one or more intermediate layers of neurons followed by a (typically) linear output layer. As indicated by Equation (9), each neuron can be thought of as a linear combiner followed by a nonlinear memoryless function $f(\cdot)$ [20]:

$$v_j = f\left(\sum_{i=0}^{m} a_{ij}x_i\right), \qquad j = 1, \dots, n \quad (9)$$

where v_j is the output of the j-th neuron, $\mathbf{x}$ is the input vector of dimension m, a_{ij} are the weights relative to each input x_i and neuron j, and n is the size of the hidden layer.

Considering only one hidden layer, the outputs v_j of each neuron are linearly combined, resulting in Equation (10):

$$y_k = \sum_{j=0}^{n} b_{kj} v_j = \sum_{j=0}^{n} b_{kj} f\left(\sum_{i=0}^{m} a_{ij} x_i\right),\ k = 1, \ldots, r \quad (10)$$

where b_{kj} is the weight that corresponds to the output of neuron j and the k element of the output vector.

The MLP is trained by optimizing the weights a_i and b_i, at each step, with respect to a cost function quantifying the discrepancy between the network output values and the target (desired) values, typically the mean squared error (MSE). For a time series, the input vector is formed by lags, and the output attempts to estimate future samples.

IV. Methodology

In this paper, we employed the AR model and the MLP to evaluate the significance of the stationarization process. We performed a comparative analysis of classical methods for removing trend and seasonality during preprocessing stage.

A. Removing Trend

1) Curve Fitting: finding a polynomial function that approximates the trend. Then, remove the fitted curve from the time series [15].

2) Differencing: calculates the difference between each point and its previous value. First order differencing removes the linear trend, because it can be considered as equivalent to taking the first order derivative, making the time series trend constant. [5], [19]

B. Removing Seasonality

Upon analyzing the daily energy consumption time series, we identified two primary seasonal patterns. The first is annual seasonality, which is due to variations in energy consumption across different seasons. The second is weekly seasonality, which is due to differences in consumption patterns between weekdays and weekends. Considering this, we employed the following methods:

1) Moving Average: it seeks to estimate the seasonal component of a time series in order to remove it from the forecasting process, resulting in a subtraction with respect to the original series [13], [15]. Given that we are working with daily measurements spanning multiple years, the moving average is computed separately for each day of the year to remove the annual trend, and for each day of the week to remove the weekly trend. Equation (11) indicates both calculations:

$$\hat{\mu}_d = \frac{1}{n}\sum_{i=1}^{n} x_{i,d}, \qquad \hat{\mu}_w = \frac{1}{m}\sum_{i=1}^{m} x_{i,w} \quad (11)$$

where $\hat{\mu}_d$ is the mean regarding each day of the year and $\hat{\mu}_w$ is the mean regarding each weekday. Considering $\bar{u}_d$ and $\bar{u}_w$ as the average of all values of $\hat{\mu}_d$ and $\hat{\mu}_w$ respectively, the seasonality can be approximated by Equation (12):

$$\hat{s}_t^D = \hat{\mu}_d - \bar{u}_d, \qquad \hat{s}_t^W = \hat{\mu}_w - \bar{u}_d \quad (12)$$

Finally, the time series without seasonality can be expressed as follows:

$$z_t = x_t - \hat{s}_t^D - \hat{s}_t^W \quad (13)$$

2) Normalization: it aims to eliminate the seasonal component by subtracting the mean and dividing by the standard deviation of the data, in this case for each day of the year which the data belongs using Equation 14 and following for each weekday using Equation 15:

$$z_{i,d}^D = \frac{x_{i,d} - \hat{\mu}_d}{\hat{\sigma}_d} \quad (14)$$

$$z_{i,w}^W = \frac{z_{i,d}^D - \hat{\mu}_w}{\hat{\sigma}_w} \quad (15)$$

C. Model parameters

For both models, the dataset was divided into training with 6126 data points, and test set with 736 values. The training set covers the period from 01/01/2002 to 11/25/2020, and the test set covers from 11/26/2020 to 12/31/2022 (in the mm-dd-yyyy format). For the MLP, the last part of the training set was separeted into validation set, spanning from 10/22/2018 to the end.

1) Autoregressive (AR) Model: The training set was used to estimate the model parameters, and the test set was used to compare the predictions based on the trained model. The only hyperparameter that required adjustment was the number of lags, which we determined using the PACF.

2) MLP: The training set was utilized to experiment with different combinations of hyperparameters, specifically the learning rate and the number of neurons in the hidden layer. The tested learning rate values were 0.001, 0.005 and 0.01, and the tested number of neurons were 10, 50, 100 and 200. Each model was then validated with hold-out using the validation set, where we obtained the best models for each detrending and deseasonality method. Then, we trained each model again using the set of hyperparameters that resulted in the lowest MSE. Finally, we evaluated the model's performance by comparing the predicted outputs, scaled back to the original scale, considering the test set.

D. Evaluation Metrics

To evaluate and compare the studied approaches, we explored the Mean Squared Error (MSE), a commonly used metric, and the Mean Average Percentage Error (MAPE), which are defined in Equations (16) and (17), respectively. It is worth mentioning that the MAPE does not depend on the scale of the data since it calculates a percentage error.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (16)$$

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \quad (17)$$

where y_i is the actual output, $\hat{y}_i$ is the estimated output and n is the number of observations.

E. Dataset

The chosen database is provided by the National Electric System Operator (ONS), and contains the daily electric load (in megawatts, MW) in the southeast region of Brazil from 2000 to 2022. It is a time series with daily data, containing about 7700 measurements [21].

V. RESULTS

In the experiments, we considered all the valid combinations of trend removing and deseasonalization techniques, which led to the nine preprocessing methods indicated in Table I. By applying each method to the original time series, we obtained the approximately stationary series displayed in Figure 1.

TABLE I: Preprocessing methods obtained by combining the removing trend and deseasonalization techniques.

	Trend		Seasonality	
Method ID	Curve fitting	Differencing	Moving Average	Normalization
1				
2	✓			
3		✓		
4			✓	
5				✓
6	✓		✓	
7	✓			✓
8		✓	✓	
9		✓		✓

It is pertinent to remark that for methods that kept the original scale of the time series, like the moving average and curve fitting, we also applied a standardization procedure to the training and test set. It should not be confused with the normalization method explained in Section IV-B for seasonal component removal. The purpose of standardization is not to remove any components, but to adjust the values to have a zero mean and unit variance. Therefore, Method 1 in Table I involves the standardized time series, as well as Methods 2, 4 and 6.

A. Preprocessing

The first step is to identify if each time series is actually stationary, and for that we used a hypothesis test called Augmented Dickey Fuller Test (ADF Test) [22]. If the p-value is below 0.05, we can reject the null hypothesis of the time series not being stationary. This suggests that we have likely eliminated the trend and seasonal component, leaving primarily the random part. In fact, according to Figure 1, it appears that the more effective the stationarity method, the closer the time series resembles a white noise.

The p-values from the ADF Test for each method can be seen in Table II. It is possible to observe that the times series without seasonality using moving average (Method 4) was not considered stationary by the ADF Test. Additionally, the time series without any stationarity removal method (Method 1), and without seasonality through normalization (Method 5) were considered stationary by the ADF Test, but the relatively high p-values suggest that they are in the edge between stationary and nonstationary.

To identify trend and seasonal patterns, we also analyzed the autocorrelation function for each method, which are exhibited in Figure 2. For Method 1 in Table I, the ACF preserves the shape of the ACF from the original time series, as expected. On the other hand, Method 2, which removes trend by curve fitting, preserves all seasonalities, while Method 3, which involves differencing, appears to remove not only the trend but also annual seasonality. Although the seasonality removal Methods 4 and 5 mentioned in IV-B are effective, they still exhibit values for all lags, which is likely due to the trend. The best one appears to be Method 9, as it removes any indication of both seasonality and trend.

B. Model results

Following the procedure outlined in Section IV-C, we applied the AR model for each scenario, and the resulting MSE and MAPE values are presented in Table II.

TABLE II: MSE and MAPE of each model, for each method and p-value of ADF Test.

		AR		MLP	
Method ID	p-value	MSE	MAPE	MSE	MAPE
1	1.67E-02	7.97E+06	5.57E-02	1.18E+06	1.91E-02
2	1.30E-10	6.64E+06	5.21E-02	1.32E+06	2.13E-02
3	2.08E-30	2.27E+06	3.02E-02	1.06E+06	1.81E-02
4	1.72E-01	2.99E+06	2.93E-02	1.84E+06	2.49E-02
5	3.01E-02	6.03E+06	4.99E-02	1.47E+06	1.82E-02
6	9.34E-12	2.65E+06	2.89E-02	1.28E+06	2.11E-02
7	2.28E-13	4.12E+06	4.04E-02	1.64E+06	2.06E-02
8	4.03E-24	9.58E+05	1.77E-02	1.10E+06	1.88E-02
9	2.07E-30	**1.24E+06**	**1.94E-02**	**1.02E+06**	**1.69E-02**

There seems to be a correlation between the ADF p-value and the error observed: as the p-value increases, so does the magnitude of the MSE and MAPE, which is expected for a model that requires stationarity. The only outlier is Method 3, which involves removing trend by differencing: although it has a very low p-value, the errors are still relatively high compared to the other methods. It appears that the ADF test primarily detects trend and is not particularly effective in identifying seasonal patterns. This is observed in Method 3, which clearly exhibits a weekly seasonality despite a very low p-value.

Subsequently, we applied the same procedure to the MLP using the model parameters described in Section IV-C. In order to attain a more robust analysis of the MLP, we trained the model 20 times and present the results in the form of a boxplot in Figure 3. To compare with the AR model, the average values of MAPE and MSE from the 20 independent experiments with the MLP are also shown in Table II.

Based on Figure 3 and Table II, we can see that the most favorable outcomes for the MLP, in terms of both MSE and MAPE, are obtained by employing differencing to remove trend and employing normalization to remove seasonality (Method 9). As the AR model, this outcome seems to be related to how effective is the stationary method applied.

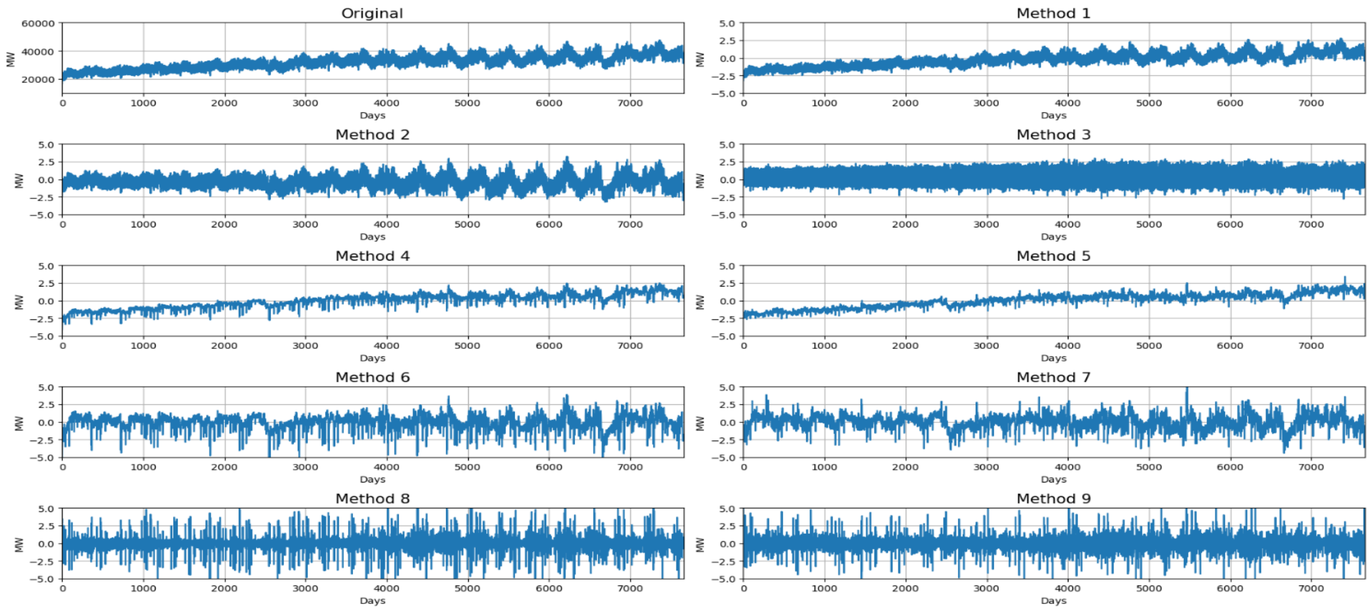


Fig. 1: Each stationarity method applied to the whole time series (train + test data)

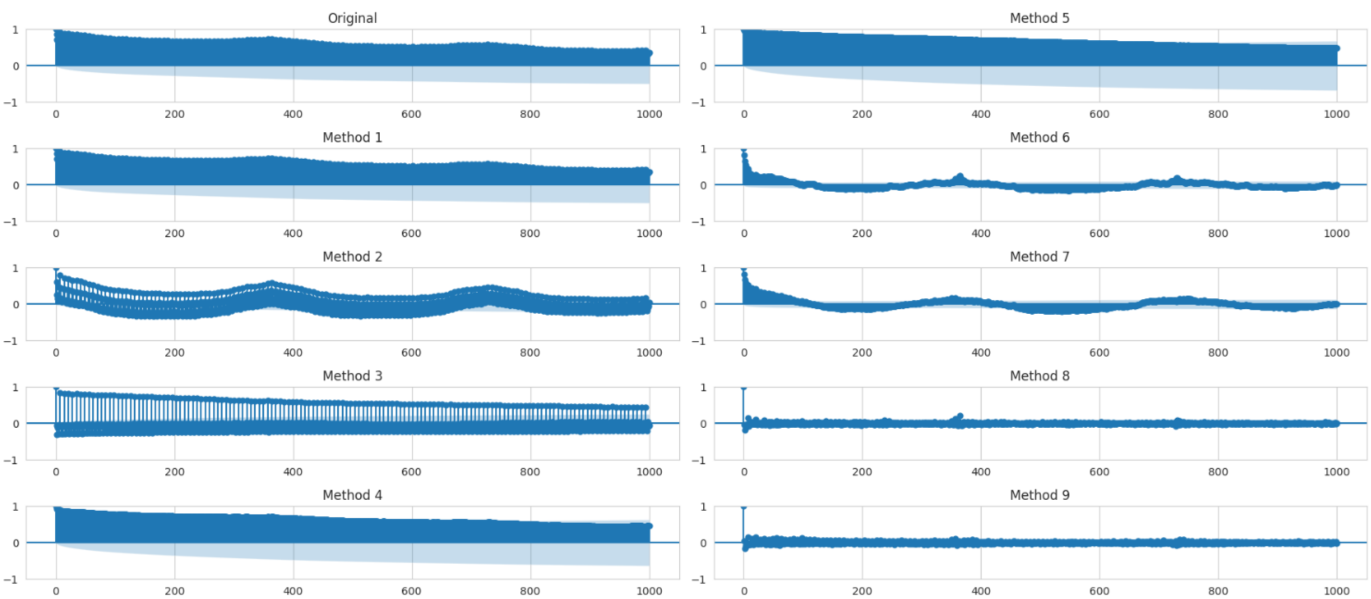
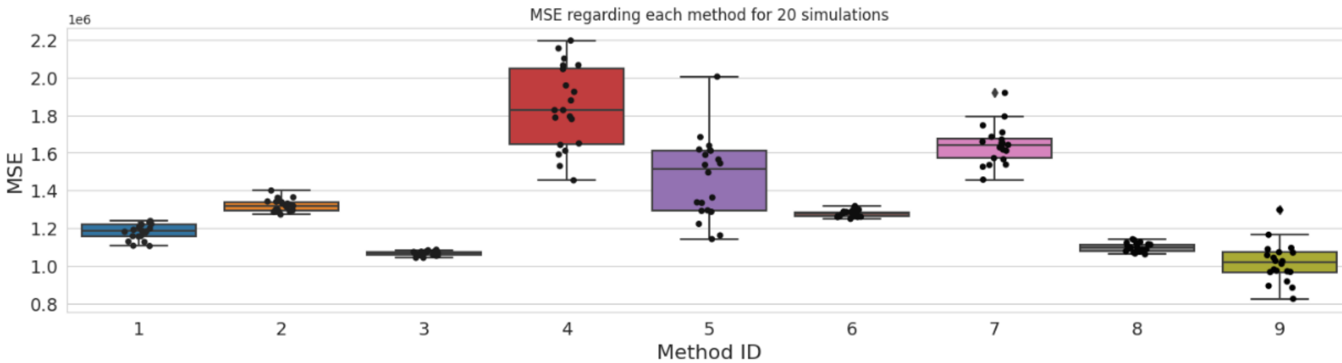


Fig. 2: Autocorrelation function for each stationarity method regarding 1000 lags on the whole time series (train + test data)

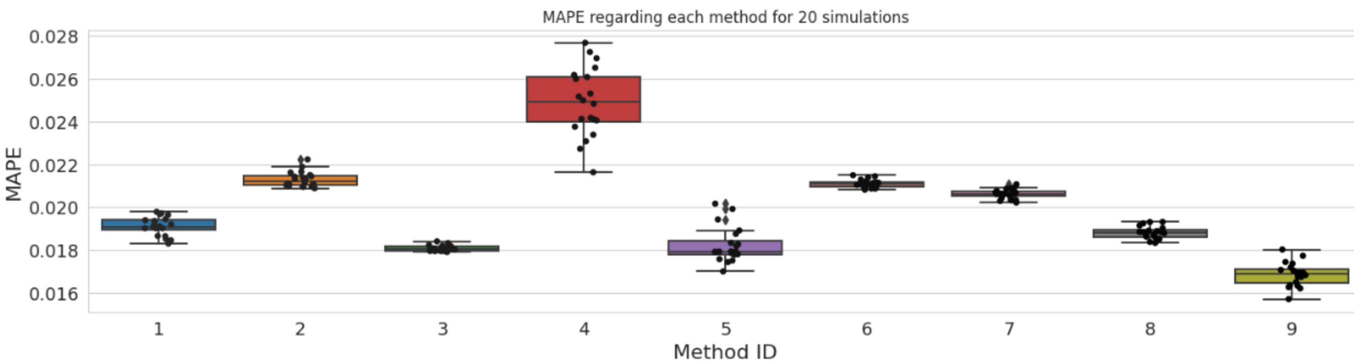
However, it is worth noting that using only differencing also yielded very favorable results for the MLP, which is different from the AR.

The outcomes from both models further demonstrate that solely applying standardization (i.e., subtracting the mean of the time series and dividing by the standard deviation from the whole series) results in a higher error compared to removing trend and seasonality to achieve stationarity. Furthermore, among the components of a time series, trend removal is probably the most critical, while only removing seasonality tends to yield worse results. This observation is evident in Figure 3, in which Methods 4 and 5 exhibit a significant dispersion in the errors, especially for MSE. Also, it appears that moving average methods are not the most efficient approach.

Finally, we can observe in Figure 4 a comparison between the predicted and original time series regarding the test set using the MLP. The figure comprises two plots: (a) one for the time series without stationarity methods, which only includes standardization, and the other (b) for Method 9, which has the lowest MSE and MAPE. It seems that Method 9 is slightly superior to the one without stationarity, which appears to be underestimating. Checking the area under the curves with the RMS value, we have for Method 1 a RMS of 1087 MW, and

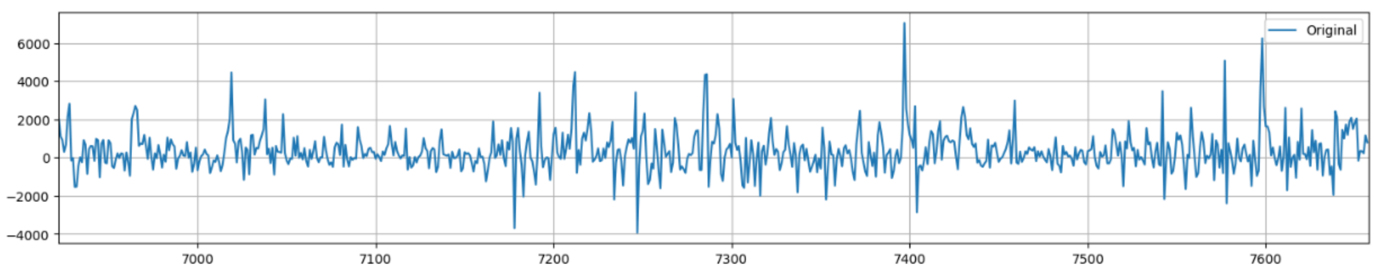


(a) MSE

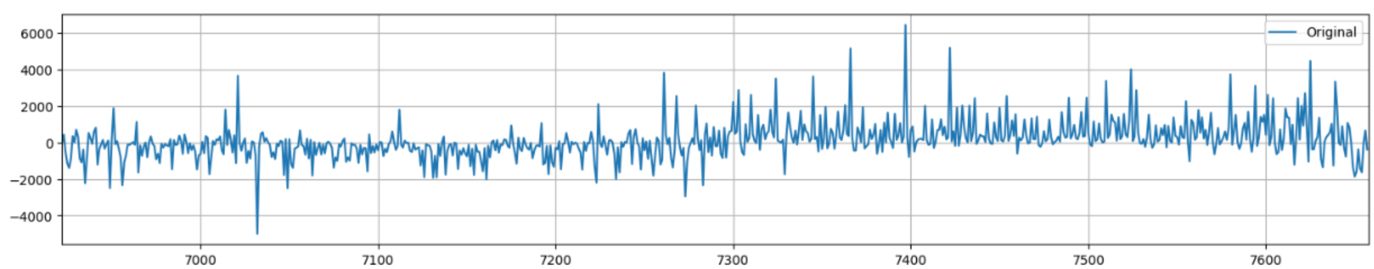


(b) MAPE

Fig. 3: Boxplot of MAPE and MSE for each method after 20 simulations



(a) Comparison between prediction and observed values of the test set without stationarity techniques



(b) Comparison between prediction and observed values of the test set using differencing and normalization (Method 9)

Fig. 4: Difference between prediction and test set

for Method 9 a RMS of 1027 MW. The only visible drawback of Method 9 is that it produces peaks in some values.

VI. Conclusion

This study explored the effects of different methods for removing trend and seasonal components to achieve stationarity in a electricity demand (energy consumption) time series. We used a Multilayer Perceptron (MLP) and an Autoregressive Model (AR) as predictors. The methodologies used to remove trend were curve fitting and differencing, and to remove seasonality we adopted normalization and moving average.

The autocorrelation plots indicated that certain methods were more successful in achieving stationarity, which was validated by the ADF Test. Additionally, an interesting observation regarding the ADF test is that it may encounter difficulty in detecting seasonal patterns.

With respect to the MLP, the obtained results suggested that eliminating trend and seasonality, depending on the method used, yields lower error rates compared to solely applying standardization across the entire dataset, as is commonly done.

For both AR and MLP, exploring the differencing method to remove trend in conjunction with season normalization resulted in the lowest MSE and MAPE. Furthermore, removing only seasonality seems to be insufficient to improve results, and it is best to combine it with a trend removal method. Moving average methods also proved to be not very effective.

As future works, it is essential to investigate the effect of the preprocessing stages on other forecasting models, as well as to expand the analysis by considering different time series related to electric power.

Acknowledgment

The authors thank the Brazilian National Council for Scientific and Technological Development (CNPq), processes number 315298/2020-0 and 308811/2019-4, and Araucaria Foundation, process number 51497, for their financial support.

References

[1] N.-R. Zhou, Y. Zhou, L.-H. Gong, and M.-L. Jiang, "Accurate prediction of photovoltaic power output based on long short-term memory network," *IET Optoelectronics*, vol. 14, no. 6, pp. 399–405, 2020.

[2] J. Holden, N. Reinicke, and J. Cappellucci, "Routee: A vehicle energy consumption prediction engine," *Society of Automotive Engineers Technical Paper Series*, vol. 2, no. NREL/JA-5400-78089, 2020.

[3] Y. Hu, J. Yan, D. Yan, Q. Lu, and J. Yan, "Lightweight energy consumption analysis and prediction for android applications," *Science of Computer Programming*, vol. 162, pp. 132–147, 2018.

[4] O. Laayati, M. Bouzi, and A. Chebak, "Smart energy management: Energy consumption metering, monitoring and prediction for mining industry," in *2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*. IEEE, 2020, pp. 1–5.

[5] G. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[6] F. Dama and C. Sinoquet, "Time series analysis and modeling to forecast: A survey," *arXiv preprint arXiv:2104.00164*, 2021.

[7] N. Elamin and M. Fukushige, "Modeling and forecasting hourly electricity demand by sarimax with interactions," *Energy*, vol. 165, pp. 257–268, 2018.

[8] L. Nashold and R. Krishnan, "Using lstm and sarima models to forecast cluster cpu usage," *arXiv preprint arXiv:2007.08092*, 2020.

[9] J. Palomares-Salas, J. De La Rosa, J. Ramiro, J. Melgar, A. Aguera, and A. Moreno, "Arima vs. neural networks for wind speed forecasting," in *2009 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*. IEEE, 2009, pp. 129–133.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *2004 IEEE international joint conference on neural networks (IEEE Cat. No. 04CH37541)*, vol. 2. Ieee, 2004, pp. 985–990.

[12] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.

[13] H. Siqueira, L. Boccato, R. Attux, and C. Lyra, "Unorganized machines for seasonal streamflow series forecasting," *International journal of neural systems*, vol. 24, no. 03, p. 1430009, 2014.

[14] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, "Temporal fusion transformers for interpretable multi-horizon time series forecasting," *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, 2021.

[15] P. A. Morettin, *Análise de Séries Temporais*. São Paulo: Atual, 1986.

[16] M. Nelson, T. Hill, W. Remus, and M. O'Connor, "Time series forecasting using neural networks: Should the data be deseasonalized first?" *Journal of forecasting*, vol. 18, no. 5, pp. 359–367, 1999.

[17] G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *European journal of operational research*, vol. 160, no. 2, pp. 501–514, 2005.

[18] S. Kay, *Intuitive probability and random processes using MATLAB®*. Springer Science & Business Media, 2006.

[19] C. Chatfield and H. Xing, *The analysis of time series: an introduction with R*. CRC press, 2019.

[20] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1998.

[21] "Carga de energia - dados abertos ons." https://dados.ons.org.br/dataset/carga-energia, accessed: 2023-01-11.

[22] J. J. Dolado, J. Gonzalo, and L. Mayoral, "A fractional dickey–fuller test for unit roots," *Econometrica*, vol. 70, no. 5, pp. 1963–2006, 2002.