



Production

ISSN: 0103-6513

ISSN: 1980-5411

Associação Brasileira de Engenharia de Produção

Bianchesi, Natália Maria Puggina; Matta, Cláudia Eliane da; Streitenberger, Simone  
Carneiro; Romão, Estevão Luiz; Balestrassi, Pedro Paulo; Costa, Antônio Fernando Branco  
A nonlinear time-series prediction methodology based on neural networks and tracking signals

Production, vol. 32, e20220064, 2022

Associação Brasileira de Engenharia de Produção

DOI: <https://doi.org/10.1590/0103-6513.20220064>

Available in: <https://www.redalyc.org/articulo.oa?id=396769689030>

- How to cite
- Complete issue
- More information about this article
- Journal's webpage in redalyc.org


UABEP  
redalyc.org

Scientific Information System Redalyc

Network of Scientific Journals from Latin America and the Caribbean, Spain and  
Portugal

Project academic non-profit, developed under the open access initiative

# A nonlinear time-series prediction methodology based on neural networks and tracking signals

Natália Maria Puggina Bianchesi<sup>a\*</sup> , Cláudia Eliane da Matta<sup>a</sup> ,  
Simone Carneiro Streitenberger<sup>a</sup> , Estevão Luiz Romão<sup>a</sup> , Pedro Paulo Balestrassi<sup>a</sup> ,  
Antônio Fernando Branco Costa<sup>a</sup> 

<sup>a</sup>Universidade Federal de Itajubá, Itajubá, MG, Brasil

\*natalia.bianchesi@unifei.edu.br, naty\_bianchesi@hotmail.com.br

## Abstract

**Paper aims:** This paper presents a nonlinear time series prediction methodology using Neural Networks and Tracking Signals method to detect bias and their responsiveness to non-random changes in the time series.

**Originality:** This study contributes with an innovative approach of nonlinear time series prediction methodology. Furthermore, the Design of Experiments was applied to simulate datasets and to analyze the results of Average Run Length, identifying in which conditions the methodology is efficient.

**Research method:** Datasets were generated to simulate different nonlinear time series by changing the error of the series. The methodology was applied to the datasets and the Design of Experiments was implemented to evaluate the results. Lastly, a case study based on total oil and grease was performed.

**Main findings:** The results showed that the proposed prediction methodology is an effective way to detect bias in the process when an error is introduced in the nonlinear time series because the mean and the standard deviation of the error have a significant impact on the Average Run Length.

**Implications for theory and practice:** This study contributes to a discussion about time series prediction methodology since this new technique could be widely used in several areas to improve forecast accuracy.

## Keywords

Nonlinear time series. Time series forecasting. Neural networks. Tracking signals. Design of Experiments.

**How to cite this article:** Bianchesi, N. M. P., Matta, C. E., Streitenberger, S. C., Romão, E. L., Balestrassi, P. P., & Costa, A. F. B. (2022). A nonlinear time-series prediction methodology based on neural networks and tracking signals. *Production*, 32, e20220064. <https://doi.org/10.1590/0103-6513.20220064>

Received: May 13, 2022; Accepted: Sept. 9, 2022.

## 1. Introduction

The modelling of time series is extremely important to make good inferences about the future, which provides a strong theoretical foundation for information processing and decision analysis, which has been an important domain of research (Pant & Kumar, 2022). Analysis and prediction for time series provide a better method of decision support (Hu et al., 2020).

Human activities and nature produce nonlinear time series every day, such as medical observations, financial recordings, physiological signals, and weather data (Qian et al., 2020). In general, time series is found in any domain of applied science and engineering which involves temporal measurements.

The problems of time series forecasting have been paid great attention since it was proposed (Hu et al., 2020). Many examples are found in the literature in several areas of knowledge. There are papers about the use of time series to forecast the daily average water level of a hydrological station (Wang & Lou, 2019); the



effects of adjuvant endocrine therapy on the health-related quality of life (Xiao et al., 2020); supply chains (Mircetic et al., 2022) and econometric approaches (Matta et al., 2021).

Researchers are often searching for the best forecasting method that results in the most approximated predicted value of real data. Some studies were conducted to evaluate some time series forecasting methods (Liu et al., 2021; Verma et al., 2021; Mao & Xiao, 2019).

Traditional methods include time series regression, Auto-Regressive Integrated Moving Average (ARIMA), and exponential smoothing based on linear models (Kumar & Murugan, 2017). All these methods assume a linear relationship among the past values of the forecast variable and therefore nonlinear patterns cannot be captured by these models (Wong et al., 2010).

The Artificial Neural Network (ANN) models have been proposed during the last few years for obtaining accurate forecasting results (Bandeira et al., 2020) and these were attempts to improve the conventional linear and nonlinear approaches. In this context, approaches based on ANN for time series forecasting have produced convincing results in recent decades for nonlinear models (Corzo & Solomatine, 2007; Hippert & Taylor, 2010; Xiao et al., 2012).

Considering the dynamic feature of time series forecasting, it is not unusual to find that the model needs to be explicitly updated after the passage of a certain number of time periods (Deng et al., 2004). Therefore, the process of monitoring plays an important role in accurate forecasting (Makridakis & Wheelwright, 1989). The monitoring is important to determine if a deviation occurs in the time series and to identify if a corrective action in the model needs to be taken to ensure that the forecasting process is brought back under control. Improper monitoring of results may imply uncertain predictions and incorrect decisions. Nevertheless, questions of when and how to update the model parameters are yet to be answered. Some studies were proposed to compare some monitoring methods for time series (Gardner Junior, 1985; Superville, 2019; Brence & Mastrangelo, 2006).

Different monitoring approaches have been proposed in the forecasting area. The tracking signal methods have been used to check the bias of forecasting methods (Sabeti et al., 2016) and also warn when there are unexpected outcomes from the forecast (Kumar & Murugan, 2017). Tracking signals can automatically detect changes in the forecast errors when the forecast is misbehaving. Various tracking signal measures exist, one of the earliest being the cumulative sum (CUSUM) tracking signal first proposed by Brown (1959).

Although tracking signals has been a common practice in traditional time series forecasting, this matter has not been widely addressed in intelligent time series forecasting as ANN. Some studies used tracking signals during the ANN training (Yu & Lai, 2005; Kumar & Murugan, 2017) to select the best ANN model, but it is not commonly used for monitoring forecasting.

Intelligent time series forecasting can learn and/or infer from historical data; however, the notion that an intelligent mechanism is capable of resolving all problems automatically is a misconception (Deng et al., 2004).

Despite similar issues have been noted by several authors (Berry & Linoff, 1997; Deboeck, 1994; Gardner Junior, 1985; Superville, 2019; Brence & Mastrangelo, 2006) very few research reports are found in the literature that have addressed this issue when using artificial neural network and tracking signals.

Therefore, this paper proposes a nonlinear time series prediction methodology based on neural network forecasting using the tracking signal method to detect bias and their responsiveness to non-random changes in the time series. However, different from many studies published in the area, this study generated a synthetic dataset to simulate different changes in the time series, compares the monitoring performance using the concept of Average Run Length (ARL) and applied Design of Experiments (DOE) to evaluate the performance of tracking signals, identifying in which conditions the predictor is efficient.

The paper is structured as follows: Section 2 consists of a brief background review of nonlinear time series forecasting based on multilayer perceptron and tracking signals, respectively. Section 3 contains the simulation study showing the experimental design and the forecasting system and Section 4 presents a case study. Finally, Section 5 states the main conclusions of this paper and some discussions.

## 2. Background and literature review

### 2.1. Nonlinear time-series

In time series models, historical data of the variable to be forecast are analyzed in an attempt to identify a data pattern. Then, assuming that it will continue in the future, this pattern is extrapolated to produce forecasts (Armstrong, 2001; Krishnamurthy, 2006). Classical time series models can be classified into two categories: linear models and nonlinear models.

Linear time series models have been widely used in recent years. According to Balestrassi et al. (2009), a stochastic time series  $Y_t$  is said to be linear if it can be written as shown in Equation 1. Any stochastic process that does not satisfy the condition of Equation 1 is said to be nonlinear (Tsay, 2005).

$$Y_t = \mu + \sum_{i=0}^{\infty} \varphi_i \alpha_{t-i} \quad (1)$$

where  $\mu$  is a constant,  $\varphi_i$  are real numbers with  $\varphi_0 = 1$  and  $\{\alpha_t\}$  is a sequence of independent and identically distributed (IID) random variables with a well-defined distribution function. The distribution of  $\alpha_t$  is continuous and  $E(\alpha_t) = 0$ .

However, the time series problems found in human activities and nature are mostly nonlinear. Therefore, the research for time series data prediction is mainly focused on nonlinear models. Thus, there has been increasing interest in extending the classical framework of Box & Jenkins (1970) to incorporate nonstandard properties, such as nonlinearity, non-Gaussianity, and heterogeneity.

Many nonlinear time series models have been proposed in the statistical literature, such as Threshold Autoregressive Nonlinear Autoregressive, Smooth Transition Autoregressive, and Bilinear Model (Tong, 1978; Amiri, 2015; Hamilton, 1989). A deeper theory about nonlinear time series can be found in Priestley (1980). The basic idea underlying these nonlinear models is to let the conditional mean evolve over time according to some simple parametric nonlinear function.

Table 1 shows some nonlinear time series implemented and simulated for the present study. In each case,  $\varepsilon_t \sim N(0, 1)$  is assumed to be independent and identically distributed. These three time series models are chosen to represent a variety of problems that have different time series characteristics. For example, some of the series have autoregressive (AR) or moving average (MA) correlation structures. The AR part involves regressing the variable on its own lagged values, while the MA part involves modeling the error term as a linear combination of error terms occurring contemporaneously and at various times in the past.

In contrast to the traditional piecewise linear model that allows for model changes to occur in the time space, the TAR model uses threshold space to improve the linear approximation. A time series is said to follow

Table 1. Nonlinear Time Series Models.

Model	Equation
Bilinear (BL1)	$y_t = 0.7y_{t-1}\varepsilon_{t-2} + \varepsilon_t$
Nonlinear moving average (NMA)	$y_t = \varepsilon_t - 0.3\varepsilon_{t-1} + 0.2\varepsilon_{t-2} + 0.4\varepsilon_{t-1}\varepsilon_{t-2} - 0.25\varepsilon_{t-2}^2$
Smooth transition autoregressive (STAR1)	$y_t = 0.8y_{t-1} - 0.8y_{t-1} \left[ 1 + \exp(-10y_{t-1}) \right]^{-1} + \varepsilon_t$

a k-regime self-exciting TAR (SETAR) model with a threshold variable. A criticism of the SETAR model is that its conditional mean equation is not continuous. The thresholds are the discontinuity points of the conditional mean function. In response to this criticism, smooth TAR models (STAR) have been proposed (Chan & Tong, 1986).

The Nonlinear Moving Average specifies that the output variable depends nonlinearly on the current and various past values of a stochastic term (De Gooijer & Hyndman, 2006).

The Bilinear model is a natural extension of nonlinearity employing the second-order terms in the expansion to improve the approximation (Tsay, 2005). This model was introduced by Granger & Anderson (1978) and has been widely investigated. Properties of bilinear models such as stationarity conditions are often derived by putting the model in a state space form and by using the state transition equation to express the state as a product of past innovations and random coefficient vectors.

Although the properties of these models tend to overlap somewhat, each is able to capture a wide variety of nonlinear behavior. In most time series, however, this kind of modeling is even more complex due to some features like high frequency, daily and weekly seasonality, calendar effect on weekends and holidays, high volatility and, presence of outliers (Balestrassi et al., 2009).

## 2.2. Artificial neural networks for time-series forecasting

Forecasting methods predict values in the future based on a given time series dataset, which considers assumptions in the future by evaluating historical data (Santos et al., 2020).

Artificial Neural Networks are one of the most used forecasting methods and are widely accepted as a technology offering an alternative way to tackle complex and ill-defined problems (Yu & Lai, 2005). The main reason for the increased popularity of ANNs is that these models are able to approximate almost any nonlinear function arbitrarily close. The ANN model can approximate any well-behaved nonlinear relationship to an arbitrary degree of accuracy, in much the same way that an ARMA model provides a good approximation of general linear relationships (Chen & Chen, 1995; Hornik, 1993).

Examples of ANN for nonlinear time series include Multilayer Perceptron (MLP), Radial Basis Function (RBF), Generalized Regression Neural Network (GRNN), and Support Vector Machine (SVM).

Multilayer Perceptron is one of the most popular network types (Aizenberg et al., 2016; Olson et al., 2012), and in many problems, domains seem to offer the best possible performance to describe a relationship between independent and dependent variables (Kialashaki & Reisel, 2013). MLP is a feedforward network trained with backpropagation learning algorithms (Zhai et al., 2016), and consists of one input layer, one or more hidden layers, and one output layer, as shown in Figure 1. A hidden layer is a group of neurons that have a specific function and are processed as a whole. Theoretical results prescribe that an MLP with one hidden layer (three-layer perceptron) is capable of approximating any continuous function (Hornik, 1993).

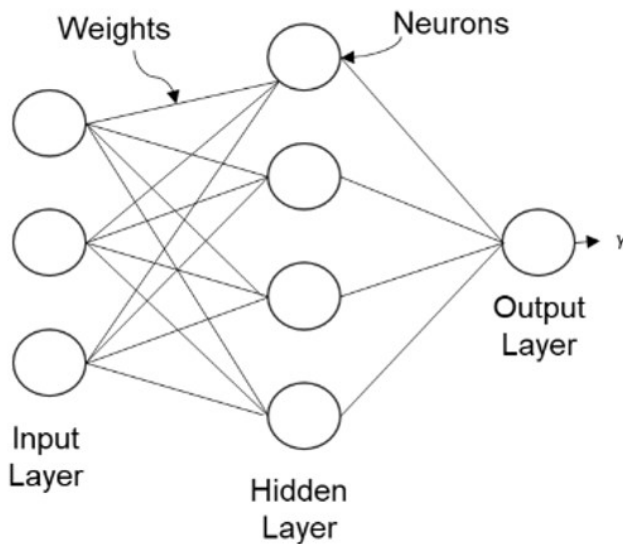


Figure 1. Multilayer feedforward ANN structure. Source: Balestrassi et al. (2009).

Each layer consists of neurons, and the neurons in two adjacent layers are fully connected with respective weights, while the neurons within the same layer are not connected (Balestrassi et al., 2009). For each neuron in the hidden or output layer, the following input-output transformation is defined in Equation 2.

$$v = f\left(\sum_{h=1}^H w_h u_h + w_0\right) \quad (2)$$

where  $v$  is the output,  $H$  is the total number of neurons in the previous layer,  $u_h$  is the output of the  $h^{th}$  neuron in the previous layer,  $w_h$  is the corresponding connection weight,  $w_0$  is the bias,  $f$  is the nonlinear activation function.

A backpropagation neural network is called self-adaptive because simulative neurons in the network organize themselves continuously according to the feedback of the output and the whole network (Haykin, 2009).

At the time of training, the values of  $w_h$  are continuously adjusted according to the feedback obtained from the real value of the response variable (Mo et al., 2017), and the weight and biases are optimized based on the minimization of the sum of the squares of the difference between the desired output and an estimated output (Balestrassi et al., 2009). The training of the network is carried out until it reaches a stable state, i.e., when there are no more significant changes in the values of synaptic weights (Haykin, 2009). The learning rate represents the rate at which the weights are adjusted. A higher learning rate allows the network to converge more rapidly, however, the chances of a non-optimal solution are greater.

Therefore, ANN can be widely used for the purpose of modelling nonlinear problems. One of the main advantages of ANN is that it is not necessary to know in advance a mathematical model that represents the data set (Chang & Tseng, 2017). Thus, ANN can describe nonlinear processes with good accuracy (Sun et al., 2017).

### 2.3. Tracking signals

Monitoring is an important component of a time series forecasting system since there is no guarantee that the past behavior and characteristics of the system continue in the future. In the forecasting and time series fields, tracking signals are used to monitor forecasting systems. Tracking signals have been applied to forecast errors and have proven useful in determining whether processes should be allowed to continue uninterrupted or if intervention is required to bring the process back in control (Krishnamurthy, 2006).

Tracking signals, in general, are ratios of the forecast errors by the mean absolute deviation. If the ratio exceeds a pre-specified limit, the forecasting approach is reexamined to see whether the pattern has changed and whether some action needs to be taken (Krishnamurthy, 2006).

The first proposal of tracking signals was made by Brown (1959) and subsequently analyzed by several researchers, including Trigg (1964), and Gardner Junior (1985).

One more common tracking signal, shown in Equation 3, compares the cumulative sum (CUSUM) of the errors at the end of each period with an unsmoothed mean absolute average (MAD). The CUSUM tracking signal is presented in the most standard production management texts and recommended by The Association for Operations Management (Ravi, 2014).

$$CTS_t = \frac{CUSUM_t}{MAD_t} \quad (3)$$

where,

$$CUSUM_t = e_t + CUSUM_{t-1} \quad (4)$$

$$e_t = X_t - F_t \quad (5)$$

$$MAD_t = \frac{AD_t}{t} \quad (6)$$

$$AD_t = |e_t| + AD_{t-1} \quad (7)$$

where the forecast error,  $e_t$ , is the actual time series value,  $X_t$ , minus the forecast  $F_t$ , and  $AD_t$  is the cumulative sum of the absolute errors.

The cumulative error can be both positive or negative, so the tracking signals can be positive or negative as well. If the forecast value is higher than the actual value then the model is in over forecasting and TS will be negative (Kumar & Murugan, 2017).

If the TS is between the control limits, then the forecast model is correctly working (Kumar & Murugan, 2017). The control limits are determined by the concept of average run length (ARL). ARL is the average length of time until the tracking signal exceeds the control limits, starting from an arbitrarily selected point in time (Sun et al., 2017). It determines the probability of detecting time series changes. If the time series doesn't have changes, the initial average run length (ARL0) measures the probability of a false alarm, defined as Type I error. So, the control limits should be defined by the simulation to yield some desired probability of getting a false alarm.

### 3. The nonlinear time-series prediction methodology

The proposed methodology has the objective of performing nonlinear time series forecasting and monitoring the errors to detect bias in the time series model assuring the forecasting accuracy.

The methodology is divided into five steps, as shown in Figure 2:

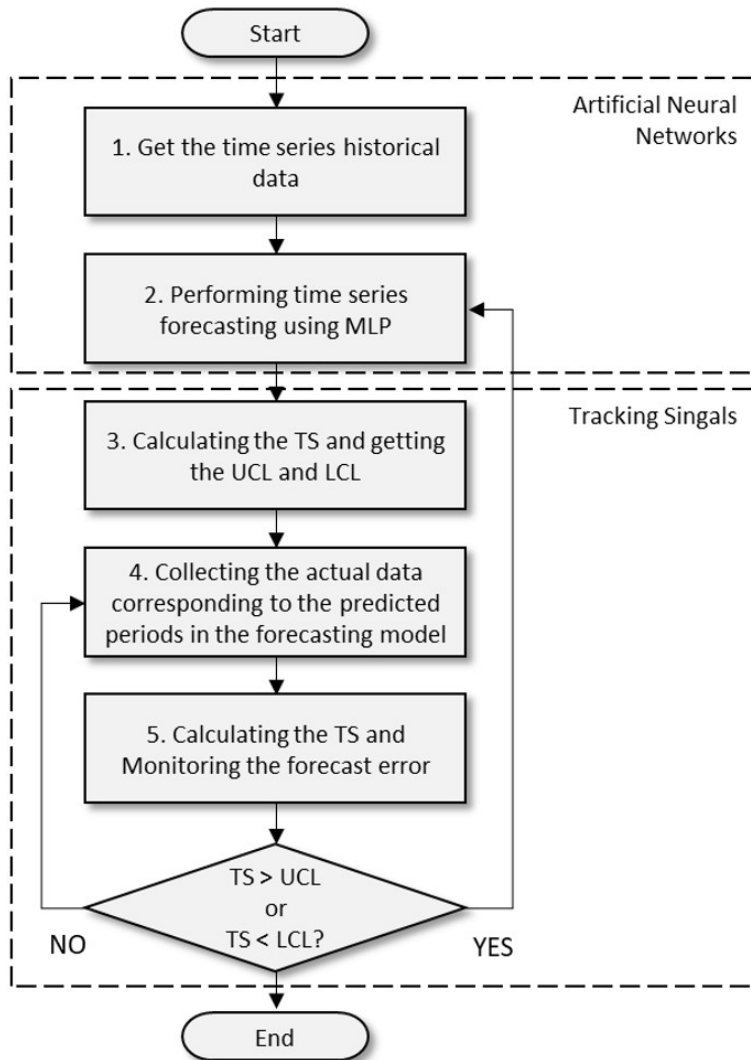


Figure 2. Prediction Methodology Flowchart. Source: Authors.

- 1) The first step is to define the variable that will be controlled and to get the time series historical data;
- 2) Considering the selected data, it is possible to perform time series forecasting using ANN and MLP;
- 3) Getting the real and the predicted values, it is possible to calculate the forecast error and obtain the TS. So, the upper control limit (UCL) and the lower control limit (LCL) of the tracking signals chart are also defined;
- 4) The actual data corresponding to the predicted periods in the forecasting model are collected and compared with the predicted values;
- 5) Then, the TS values are calculated and the forecast errors are monitored through the tracking signal chart. If the TS exceeds the control limits, the ANN is performed again, otherwise, it returns to step 4.

The methodology also can be described by the pseudocode in Table 2 that is associated with the steps in Figure 2. It will be analyzed and validated in the next sections through a simulated and a case study.

Table 2. Pseudocode.

Step	Pseudocode
1	1: $y \leftarrow \text{dataset}$
	2: $t \leftarrow \text{time series size}$
	3: $\text{end} \leftarrow y(\text{lastPosition})$
	4: $n_{pred} \leftarrow \text{number of predictions}$
2	5: for $t \leftarrow 1$ to $50$
	6: $mlp_{net} \leftarrow \text{RNA\_MLP\_train}(t, y)$
	7: $y_{pred_1} \leftarrow \text{RNA\_MLP\_test}(mlp_{net}, y, n_{pred})$
3	8: $TS_1 \leftarrow \text{cts}(y_{pred_1}, y)$
	9: end for
	10: $l_s \leftarrow \text{upper limit}$
	11: $l_l \leftarrow \text{lower limit}$
4	12: for $t \leftarrow 51$ to $\text{end}$
	13: $y_{pred_t} \leftarrow \text{RNA\_MLP\_test}(mlp_{net}, y, 1)$
5	14: $TS_t \leftarrow \text{cts}(y_{pred_t}, y)$
	15: if $(\text{cts}(t) > l_s \text{ or } \text{cts}(t) < l_l)$
	16: $mlp_{net} \leftarrow \text{RNA\_MLP\_train}(t, y)$
	17: else
	18: end for
	19: // Functions
	20: function $c \leftarrow \text{cts}(y_{pred}, y)$
	21: $N \leftarrow y_{pred} \text{ length}$
	22: $\text{end} \leftarrow y(\text{lastPosition})$
	23: $y \leftarrow y(\text{end} - N + 1 : \text{end})$
	24: $\text{error} \leftarrow y_{pred} - y$
	25: $\text{error}_{abs} \leftarrow \text{absolute value of error}$
	26: $\text{mad}(1) \leftarrow \text{error}_{abs}(1)$
	27: $\text{cusum}(1) \leftarrow 0$
	28: $\text{tsb}(1) \leftarrow 0$
	29: $\text{ad}(1) \leftarrow \text{error}_{abs}(1)$
	30: for $t \leftarrow 1$ to $N$
	31: $\text{ad}(t) \leftarrow \text{error}_{abs}(t) + \text{ad}(t-1)$
	32: $\text{mad}(t) \leftarrow \text{ad}(t) / t$
	33: $\text{cusum}(t) \leftarrow \text{error}(t) + \text{cusum}(t-1)$
	34: $\text{cts}(t) \leftarrow \text{cusum}(t) / \text{mad}(t)$
	35: end for
	36: end function
	37: function $\text{net} \leftarrow \text{RNA\_MLP\_train}(N, y)$
	38: $t \leftarrow 1 : (N-1)$
	39: $x(t) \leftarrow y(t)$
	40: $y_p(t) \leftarrow y(t+1)$
	41: $\text{net}_{param_{train}} \leftarrow 80/100$
	42: $\text{net}_{param_{valid}} \leftarrow 10/100$
	43: $\text{net}_{param_{test}} \leftarrow 10/100$
	44: $\text{net}_{type} \leftarrow \text{RNA MLP Backpropagation}$
	45: $\text{hiddenLayer}_{size} \leftarrow 10$
	46: $\text{net} \leftarrow \text{feedForwardNet}(\text{hiddenLayer}_{size}, 'trainlm')$

Table 2. Continued...

Step	Pseudocode
47:	$net_{param}_{show} \leftarrow 50$
48:	$net_{param}_{lr} \leftarrow 0.01$
49:	$net_{param}_{epochs} \leftarrow 1000$
50:	$net_{param}_{goal} \leftarrow 1e-06$
51:	$[net, tr] \leftarrow train(net, x, y, p)$
52:	end function
53:	function $y_{pred} \leftarrow RNA\_MLP\_test(N, y, n_{pred})$
54:	$y_{pred} \leftarrow (y)_{net}$
55:	end function

#### 4. Simulation study

In this section, the simulation study for the nonlinear time series predictor is examined to test different scenarios and to generalize the results. First, it is presented the DOE applied in the simulation datasets. The datasets contain a part of the original time series and a part in which the series error ( $\epsilon_t$ ) is modified. Next, the time series forecasting and the tracking signals are applied. Finally, the ARL results of the tracking signals are analyzed by DOE and it is possible to determine how much the series error ( $\epsilon_t$ ) can be modified so that the tracking signals are still efficient in detecting the series shift.

##### 4.1. Design of experiments

DOE is a commonly used technique for processes where experiments are planned to find the optimal and robust solution through the combination of input variables at different levels (Lee et al., 2007; Dascalescu et al., 2008). However, the DOE technique can be used for simulation problems, Figure 3.

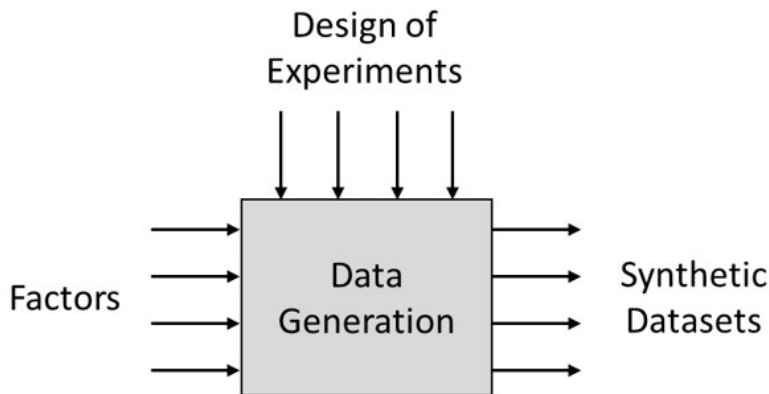


Figure 3. DOE for simulation. Source: Bianchesi et al. (2019).

The DOE applied in simulation problems increases the transparency of simulation model behavior and the effectiveness of reporting simulation results (Lorscheid et al., 2012). Furthermore, it allows controlling the factors that will be used in the simulation and present better and faster results than trial and error simulation. Therefore, DOE is a useful and necessary part of the analysis of simulation (Lee et al., 2007).

Concerning generating distinct time series datasets different from the original models (Table 1), the series error ( $\epsilon_t$ ) present in the time series models was modified by changing its mean and standard deviation in a controlled way by the DOE technique. The original series error ( $\epsilon_t$ ) assumes a normal distribution  $N(0;1)$ .

The levels assumed for the mean of the series error ( $\epsilon_t$ ) were defined considering the concept of effect size. Effect size is defined as the estimation of the magnitude of the relationship between variables or the difference between two samples (Rosenthal, 1994).

Cohen (1988) classified the effect size as small ( $d = 0.2$ ) when the difference between two samples is difficult to see with the naked eye, and as large ( $d = 0.8$ ) when the difference between two samples is evident to see with the naked eye.

According to Cohen (1988), for  $d = 0.2$ , 58% of the error of the original series will be above the mean of the error of the modified series; 92% of the two series of error will be overlapped, and there is a 56% of chance that a randomly chosen element from the error of the original series will be greater than a randomly chosen element from the error of modified series. For  $d = 0.8$ , 79% of the error of the original series will be above the mean of the error of the modified time series; 69% of the two series of error will be overlapped and there is a 71% of chance that a randomly chosen element from the error of the original series will be greater than a randomly chosen element from the error of modified series. That is depicted in Figure 4.

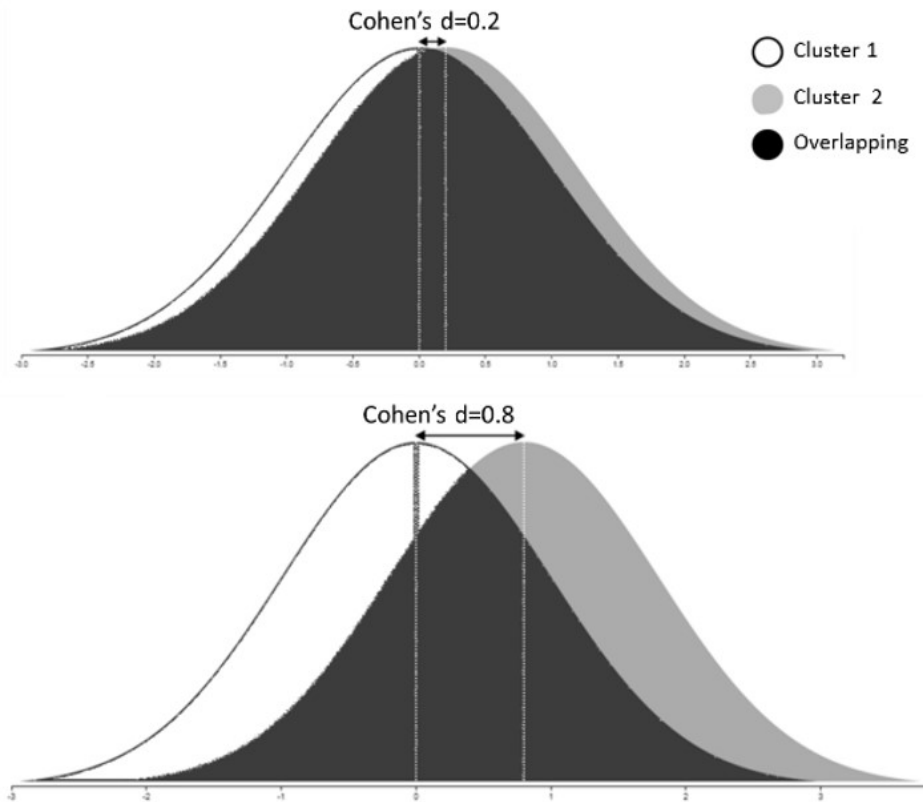


Figure 4. Cohen's d representation. Source: Bianchesi et al. (2019).

The standard deviation of the series error ( $\epsilon_j$ ) was simulated to be between 0.5 and 3.0, according to the control chart concept. Control rules take advantage of the normal curve in which 68.26% of all data is within plus or minus one standard deviation from the average, 95.44% of all data is within plus or minus two standard deviations from the average, and 99.73% of data will be within plus or minus three standard deviations from the average (Cohen, 1988), as shown in Figure 5.

Thus, a resume of the DOE factors and their levels is detailed in Table 3.

Considering the factors in Table 3 and the response surface method (RSM), the design matrix was constructed by using any statistical software with DOE routines.

The RMS is a class of DOE that is widely used because of its simplicity and effectiveness to design the experiments and it minimizes the number of experiments for a specific number of factors and its levels (Montgomery, 2009). The objective of RSM is to explore the relationship between the response and the studied factors involved in an experiment (Amdoun et al., 2018). The mathematical model of RSM is a second order polynomial equation, whose advantage is to be easy to estimate and then be applied in order to approximate the response (Cui et al., 2012).

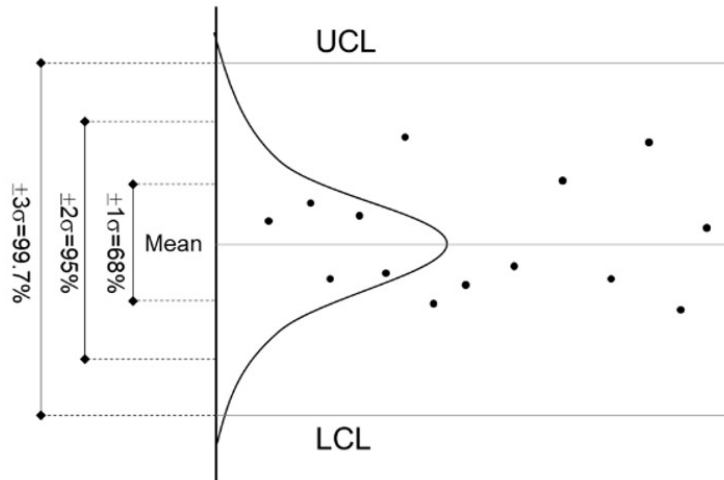


Figure 5. Relationship of Control Chart to Normal Curve. Source: Montgomery (2009).

Table 3. Experimental Factors.

	Mean (X)	Standard Deviation (SD)
(-1)	0.2	0.5
(+1)	0.8	3.0

The design matrix was obtained with RSM for 2 factors with 5 center points and 2 replications, resulting in 26 experiments presented in Table 4.

Table 4. Design Matrix.

Run Order	Replication	Mean	SD
1	1	0.20	0.50
2	1	0.80	0.50
3	1	0.20	3.00
4	1	0.80	3.00
5	1	0.08	1.75
6	1	0.92	1.75
7	1	0.50	0.02
8	1	0.50	3.52
9	1	0.50	1.75
10	1	0.50	1.75
11	1	0.50	1.75
12	1	0.50	1.75
13	1	0.50	1.75
14	2	0.20	0.50
15	2	0.80	0.50
16	2	0.20	3.00
17	2	0.80	3.00
18	2	0.08	1.75
19	2	0.92	1.75
20	2	0.50	0.02
21	2	0.50	3.52
22	2	0.50	1.75
23	2	0.50	1.75
24	2	0.50	1.75
25	2	0.50	1.75
26	2	0.50	1.75

The design matrix is a guide to indicate the combination of the mean and standard deviation to generate the modified series error ( $\varepsilon_t$ ), that will be used for the time series simulation. For each time series model, we generated 26 series containing the modified series errors ( $\varepsilon_t$ ), according to the design matrix. Afterward, it was generated 100 samples of each 26 datasets, which contains 100 samples of the original series, in order to get a false alarm of 0.01 (Bischak & Trietsch, 2007), and 50 samples with the modified series error ( $\varepsilon_t$ ).

## 4.2. Time-series forecasting

After getting the datasets, MLP was implemented in each time series model by using Matlab® software. For the original time series, 50 samples were trained by using 80% random observations from the original dataset, and 10% was the validation set. So, the remaining 10% of observations were allocated to the testing set. In the training process, we used the seed for sampling equal to a constant number. The neural network used was the three-layer perceptron due to presents excellent performance when dealing with nonlinear data sets (Kialashaki & Reisel, 2013), and theoretical results prescribe that an MLP with one hidden layer is capable of approximating any continuous function (Mo et al., 2017). The number of hidden units was defined as 10, the training function used was the Bayesian Regularization, and the learning rate was set to 0.01. The steps used to predict are set as 1 or 2 according to the lag seasonality of the model. The steps ahead to predict, or forecasting horizon, represents the number of steps ahead of the lagged input values that the predicted output lies. In this case, due to the small synthetic time series and considering the error propagation throughout the steps prediction, just one step ahead was used. The output of the network can be combined with previous input values, shifted one-time steps, and repeated predictions made. Since the runtime is mainly dependent on the minimum error to be reached and this error is not linear, it is not correct to say that predicting two steps ahead doubles the runtime of predicting one step ahead (Balestrassi et al., 2009).

After MLP was implemented, other 50 samples of the original time series were predicted, and the tracking signals were calculated by Equation 3. The initial values of CUSUM (Equation 4) were set to zero as suggested by Gardner Junior (1985) and McClain (1988). Then, the control limits (CL) for the tracking signals were determined based on simulation to detect the probability of getting a false alarm, defined as Type I error (Gardner Junior, 1985). Type I error is the probability of a tracking signal exceeding the control limits when the time series does not have changes. In this paper, the simulation was performed assuming a desired false alarm rate between 0.01 and 0.0105 considering a confidence interval (CI) of 99%, which represents an ARL0 between 100 and 95 samples of the original series. Next, the modified part of the series was implemented in the forecasting model and 50 samples were predicted.

Figure 6 shows an example of the dataset generated for the STAR1 model and the time series forecast using MLP (ANN) and Figure 7 shows an example of the Bilinear model. It can be noted in both cases that from sample 100 onwards, the forecasting model does not fit the data well and the forecast error ( $e_t$ ) increases once the time series error ( $\varepsilon_t$ ) has changed.

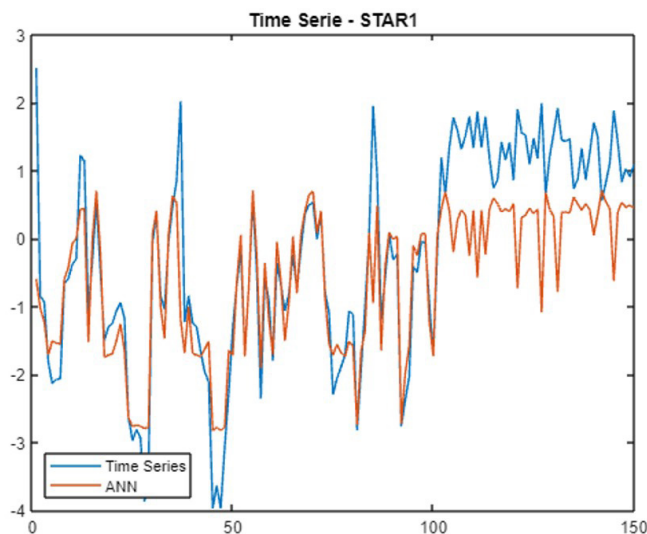


Figure 6. STAR1 - Time Series Forecasting. Source: Authors.

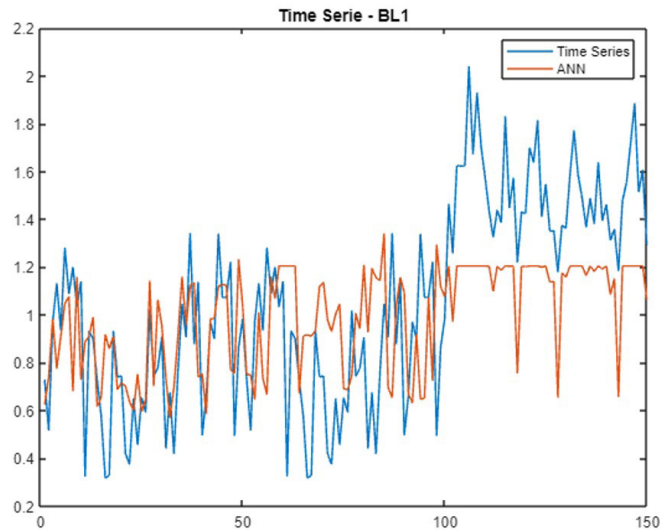


Figure 7. BL1 - Time Series Forecasting. Source: Authors.

After that, the tracking signals for the modified time series were also calculated. Figures 8, 9, and 10 show an example of the methodology applied for the STAR1 series for the training samples, the original series, and the modified series, respectively.

The forecast error ( $e_t$ ) for the original time series is normally distributed whereas from the modified series it is not normally distributed and presents larger values. Also, it is possible to note that for the modified series the tracking signals exceed de control limits early.

From the tracking signals, the ARL were obtained for each dataset and the average results of STAR1, BL1, and NMA models are presented in Table 5.

Given the ARL results, the DOE statistics were analyzed. Thus, it was possible to establish mathematical relationships between the responses and the input parameters by using the Ordinary Least Squares Method (OLS) and Analysis of Variance (ANOVA). Furthermore, the residual was analyzed to ensure that it was uncorrelated, normally, and randomly distributed (Priestley, 1980).

Therefore, we could obtain the general equation of each model that are described in Equations 8, 9, and 10, which present an uncoded coefficient. All terms are significant with a significance level of 5%, and the

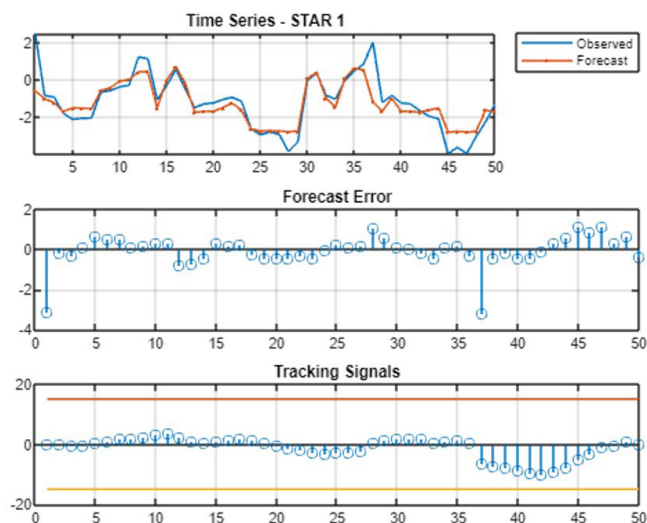


Figure 8. STAR1 - Training samples. Source: Authors.

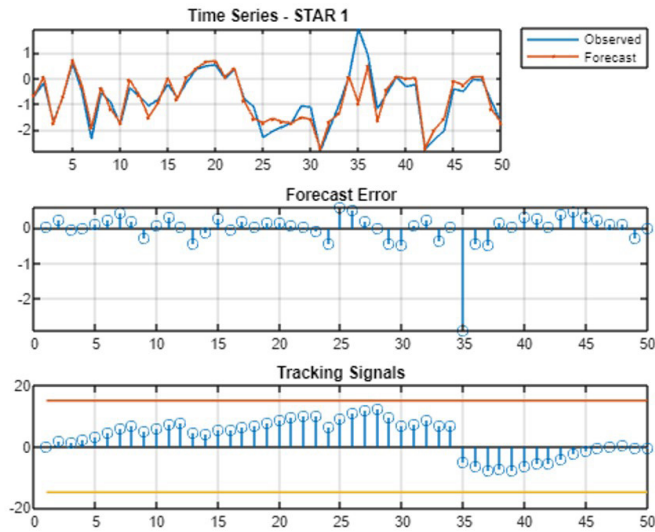


Figure 9. STAR1 - Original series. Source: Authors.

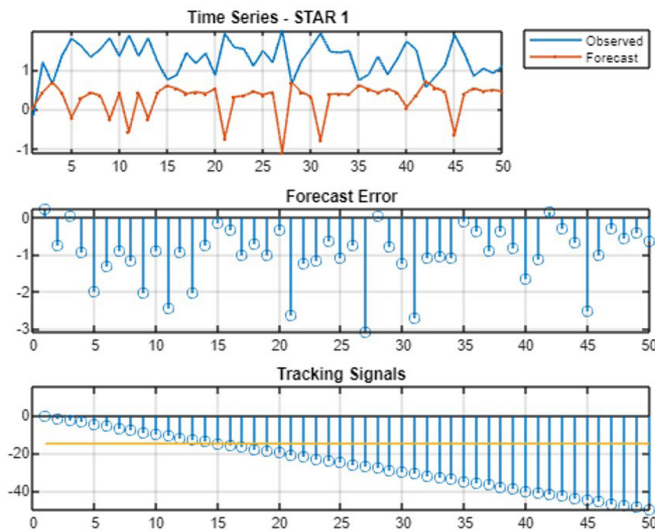


Figure 10. STAR1 - Modified series. Source: Authors.

values of  $R^2$  (adj.) are equal to 81.25%, 97.11%, and 92.65% for STAR1, Bilinear (BL1), and NMA, respectively, indicating good reliability and predictability of the model.

$$ARL(STAR1) = 56.10 - 99.70 * X - 7.90 * SD + 55.98 * X^2 + 12.53 * X * SD \quad (8)$$

$$ARL(BL1) = 22.90 - 11.10 * X - 3.25 * SD + 4.61 * X^2 + 0.39 * SD^2 + 2.28 * X * SD \quad (9)$$

$$ARL(NMA) = 30.08 - 36.71 * X + 2.79 * SD + 27.93 * X^2 - 0.53 * SD^2 - 2.29 * X * SD \quad (10)$$

Afterward, it is possible to analyze which factors are more significant and have a greater influence on the final result. For better visualization of the factor significance, Figures 11, 12, and 13 present the Pareto Chart for each method. STAR1 and NMA show that the mean presents an effect relatively larger than the standard deviation, while for BL1 the standard deviation presents more influence.

Table 5. ARL Results.

Run Order	ARL		
	STAR1	BL1	NMA
1	35.76	32.26	25.59
2	18.15	16.11	20.56
3	17.76	16.00	17.12
4	16.94	16.00	29.65
5	23.84	16.95	18.21
6	17.04	16.00	17.17
7	17.66	18.74	17.83
8	17.07	16.00	19.90
9	17.60	16.01	19.90
10	17.60	16.01	19.90
11	17.60	16.01	19.90
12	17.60	16.01	19.90
13	17.60	16.01	19.90
14	47.94	31.25	26.38
15	28.55	16.20	20.63
16	18.63	16.00	17.46
17	16.94	16.00	30.34
18	41.83	17.82	18.49
19	17.61	16.00	16.97
20	16.23	18.53	18.03
21	16.83	16.00	20.05
22	18.27	16.00	20.05
23	18.27	16.00	20.05
24	18.27	16.00	20.05
25	18.27	16.00	20.05
26	18.27	16.00	20.05

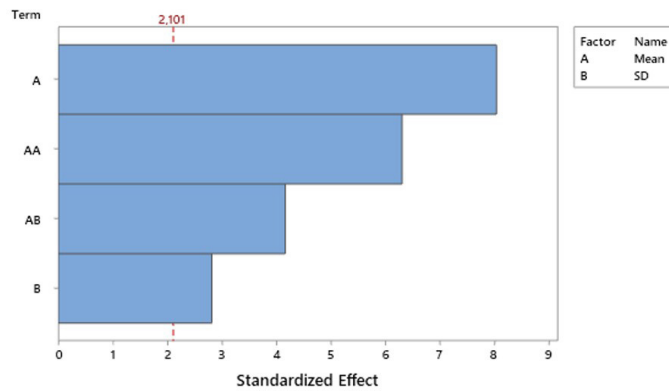


Figure 11. Pareto Chart of the Standardized Effects for ARL,  $\alpha=0.05$  – STAR1. Source: Authors.

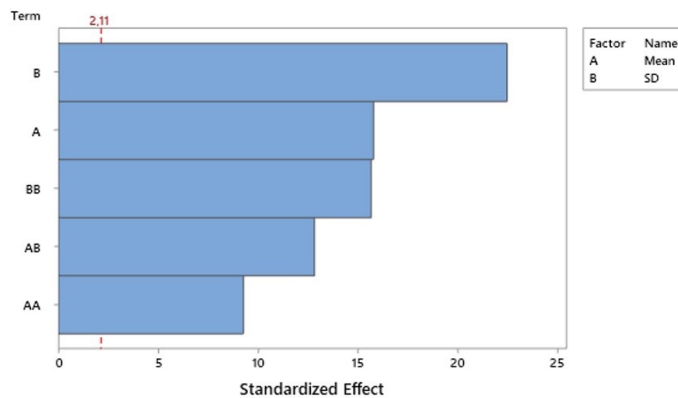


Figure 12. Pareto Chart of the Standardized Effects for ARL,  $\alpha=0.05$  – BL1. Source: Authors.

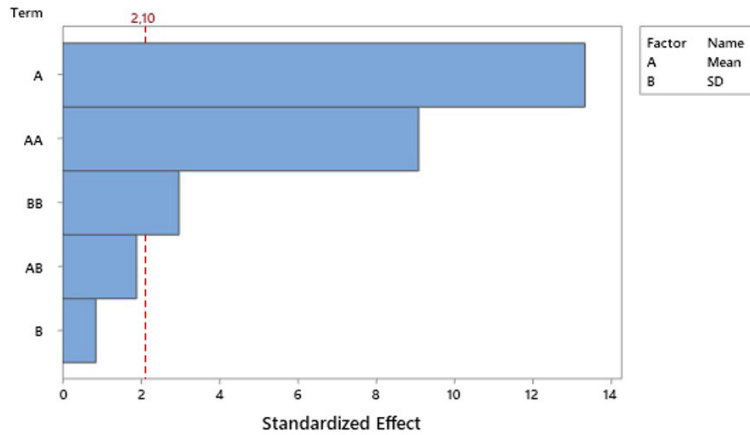


Figure 13. Pareto Chart of the Standardized Effects for ARL,  $\alpha=0.05$  – NMA. Source: Authors.

At this moment, it is necessary to understand how each significant factor influences the model. Then, an analysis of the main effects, Figures 14, 15, and 16, and the interaction plots, Figures 17, 18, and 19, was developed for STAR1, BL1, and NMA, respectively.

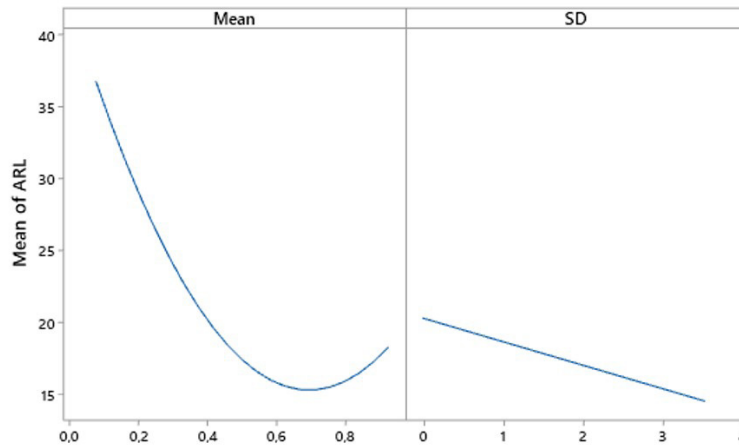


Figure 14. Main Effects Plot for ARL – STAR1. Source: Authors.

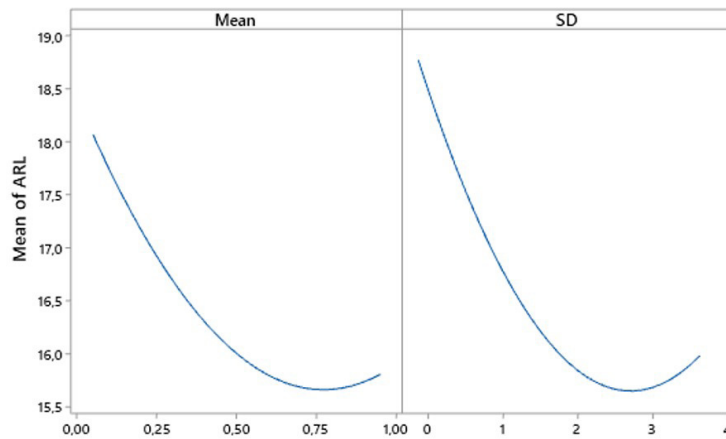


Figure 15. Main Effects Plot for ARL – BL1. Source: Authors.

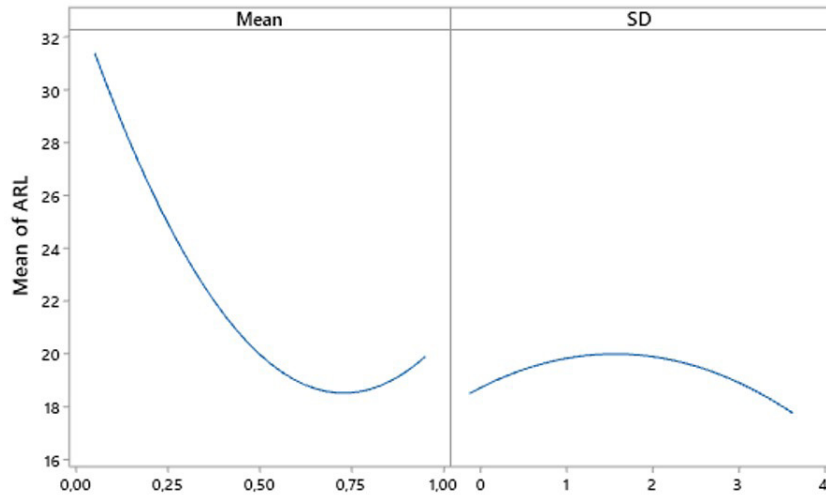


Figure 16. Main Effects Plot for ARL – NMA. Source: Authors.

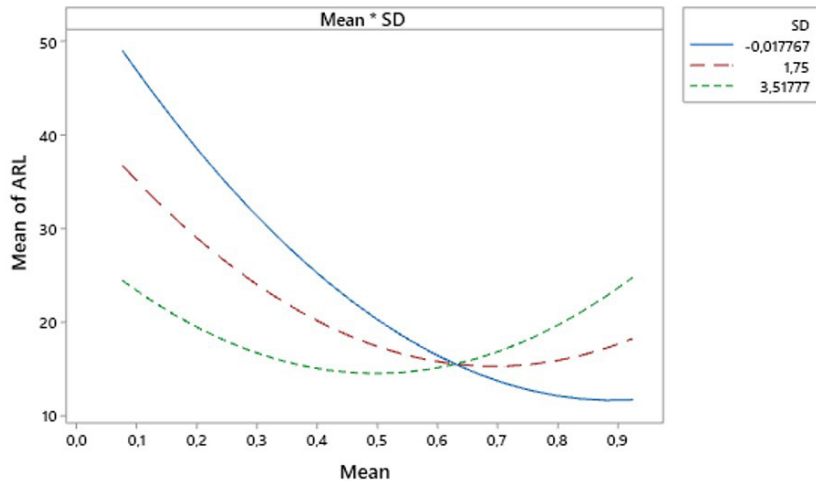


Figure 17. Interaction Plot for ARL – STAR1. Source: Authors.

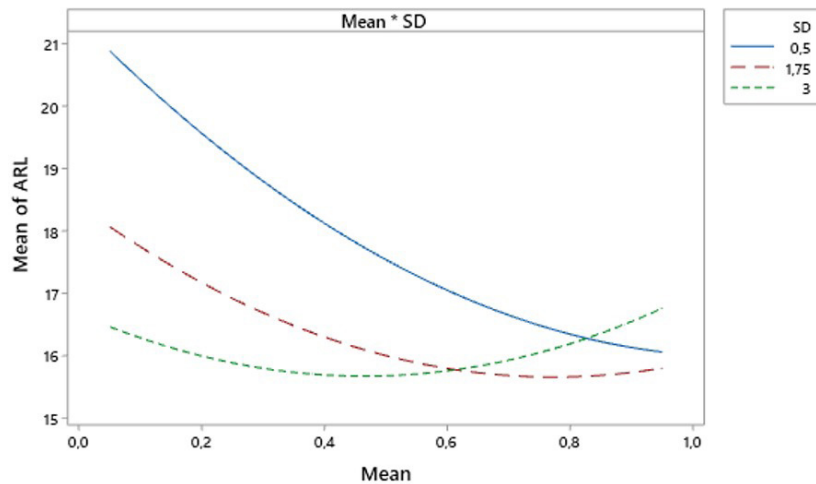


Figure 18. Interaction Plot for ARL – BL1. Source: Authors.

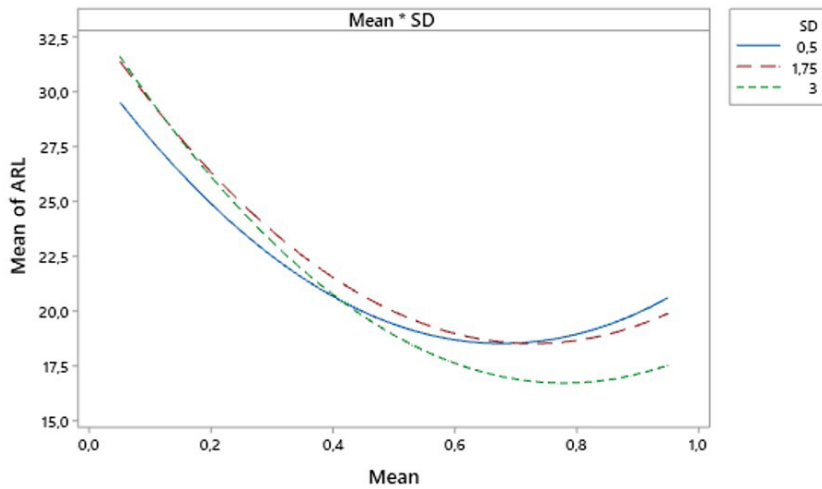


Figure 19. Interaction Plot for ARL – NMA. Source: Authors.

As inferred from Figure 14, for the STAR1 model the ARL decreases until the mean reaches the value of 0.7, then increases again until the mean reaches 0.8. In addition, the increase in the standard deviation implies a decrease in the ARL results, being easier to detect the change in the time series. Besides, there is a significant interaction between the mean and the standard deviation, as shown in Figure 17. The smallest ARL occurs when the mean is greater than 0.65 and the standard deviation is equal to 0.5, but if the mean is smaller than 0.65, the best ARL is obtained with the standard deviation equal to 3.

Analyzing Figures 15 and 16, the effect of the mean in the ARL for BL1 and NMA follows the same trend as for STAR1. For the BL1 model, the ARL results present a better performance when the standard deviation reaches 2.8 and then increases again until it reaches 3.5. Moreover, Figure 18 also shows an interaction between the mean and the standard deviation. The smallest ARL occurs when the mean is about 0.5 and the standard deviation is equal to 3, but if the mean is greater than 0.5, the best ARL is obtained with the standard deviation equal to 1.75.

The NMA model presents a small variation in the ARL values as the standard deviation changes, showing a better ARL with a standard deviation of 3.5. Figure 19 also shows an interaction between the mean and the standard deviation for the NMA, which presents a smaller ARL with a mean greater than 0.5 and a standard deviation of 3.5. For mean values smaller than 0.5, the best ARL is reached with a standard deviation of 0.5.

The effect in the ARL results concerning the change in the mean and standard deviation of the series error can also be seen in Figures 20, 21, and 22. The STAR1 model is the one that presents the greatest variation of ARL results according to changes in the series error.

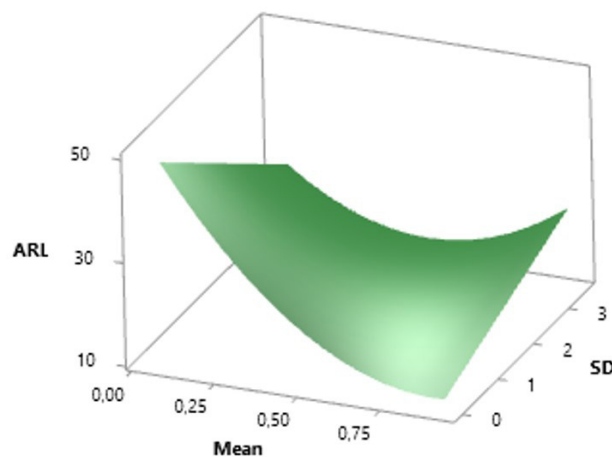


Figure 20. Surface Graph for ARL – STAR1. Source: Authors.

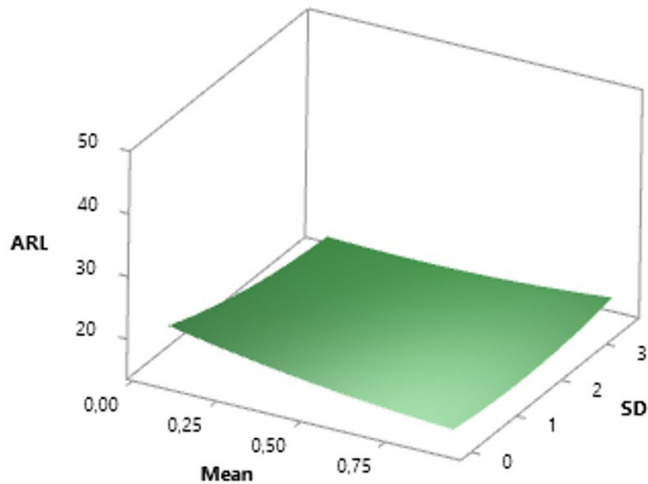


Figure 21. Surface Graph for ARL – BL1. Source: Authors.

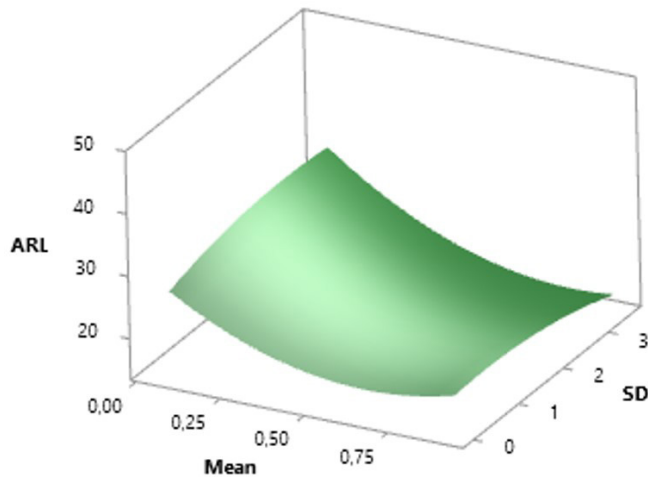


Figure 22. Surface Graph for ARL – NMA. Source: Authors.

Then, the desirability analysis was performed to find the optimal solution to minimize the ARL response. The desirability function, defined by Harrington (1965) and Derringer & Suich (1980), is one of the approaches used for factor optimization (Candiotti et al., 2014). It is based on the transformation of all the obtained responses from different scales into a scale-free value. The values of desirability (D) functions lie between 0 and 1. The value 0 is attributed when the factors give an undesirable response, while the value 1 corresponds to the optimal performance (Derringer & Suich, 1980).

Table 6 presents the results of the desirability analysis for the three nonlinear time series. They were obtained according to a similar procedure as shown for the STAR1 model, Figure 23. For STAR1, the smallest ARL = 11.79 is got with a mean equal to 0.89 and a standard deviation equal to 0.02, with D=1. For BL1, the smallest ARL=15.62 is got with a mean equal to 0.59 and a standard deviation equal to 2.41, with D=1. For NMA, the smallest ARL=15.45 is got with a mean equal to 0.80 and a standard deviation equal to 3.52, with D=1.

Table 6. Desirability Results.

Factors	STAR1	BL1	NMA
Mean	0.89	0.59	0.80
SD	0.02	2.41	3.52
ARL	11.79	15.62	15.45

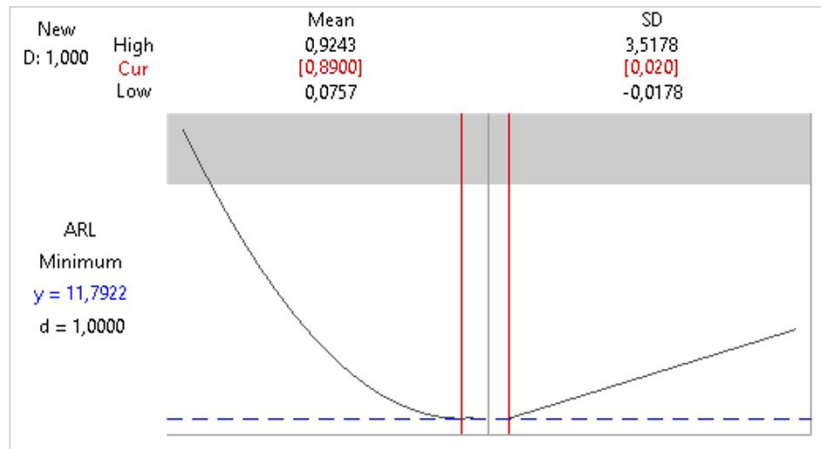


Figure 23. Response Optimization for ARL (a). Source: Authors.

The main findings of the experimental results are following summarized.

- The prediction methodology presents better performance applied in the STAR1 model once the change in the time series can be detected early than in other models.
- For STAR1 and NMA, the change in the mean of the error has a significant effect on the ARL results being the smallest ARL when the mean has a high change, while the BL1 presents a smaller ARL with a moderate change in the mean.
- The results also show that the standard deviation has a significant effect on the ARL results, mainly concerning the BL1 and NMA models, which have the best ARL result when the standard deviation has a moderate to high change while the smallest ARL for STAR1 is obtained with small standard deviation change. This could be theoretically expected since BL1 and NMA have a moving average (MA) correlation and a change in the standard deviation of the error can imply a greater variability in the y values.

## 5. Case study

### 5.1. Total oil and grease

The primary processing of petroleum consists of the separation of oil, gas, and water obtained from the extraction of crude oil. In this process, the produced water is a complex mixture that can be characterized by the total oil and grease (TOG) (Yang, 2011; Ray & Engelhardt, 1992). TOG is also widely used for environmental surveillance purposes (Brasil, 2007).

The TOG value obtained from a chemical analysis of produced water is extremely method dependent. According to Yang (2011), the main reference methods for measuring TOG are infrared absorption, gravimetric, and gas chromatography. Among these, gravimetric is considered in the present study. Some advantages of gravimetric analysis are related to its simplicity and low cost. However, it has a disadvantage in terms of sensitivity, since its lower detection limit varies from 5 to 10 mg/L (Iggunnu & Chen, 2014).

The TOG value obtained by the gravimetric method considers both the dispersed oil fraction, which represents the oil in the produced water in the form of small droplets and the dissolved oil fraction, which is defined as the amount of oil in the produced water in a soluble form (Ray & Engelhardt, 1992).

### 5.2. Application of the methodology

It used 100 samples of TOG in the case study, obtained from an oil platform in Brazil. First, 50 samples were trained by using MLP with the parameters set as shown in the simulation study. Afterward, the data were predicted, and the tracking signals were calculated using the forecast errors. So, it was possible to determine the

control limits for the tracking signals, assuming a desired false alarm rate of 0.01. The control limits calculated were equal to  $-4$  and  $4$ .

Then, new data were input into the predictor and new forecasting and tracking signals were calculated. This process was done until a tracking signal exceeded the control limit in the 54<sup>th</sup> sample. Figure 24 shows the real TOG series and the forecasting, as well as the forecast error and the tracking signal.

Therefore, it was necessary to apply again the MLP and update the forecasting model. So, the tracking signals and the control limits were recalculated, and another 30 samples were predicted using this forecasting model until the tracking signal transposed the upper limit, as shown in Figure 25.

So, the forecasting model was updated and it was used in the rest of the samples, being all of them inside the control limits of the tracking signals chart, Figure 26.

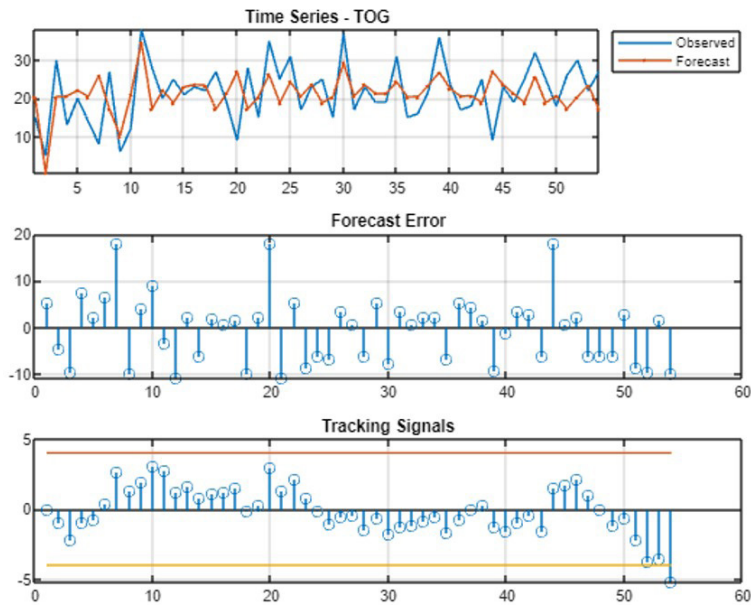


Figure 24. TOG Time Series Forecasting – Samples 1 to 54. Source: Authors.

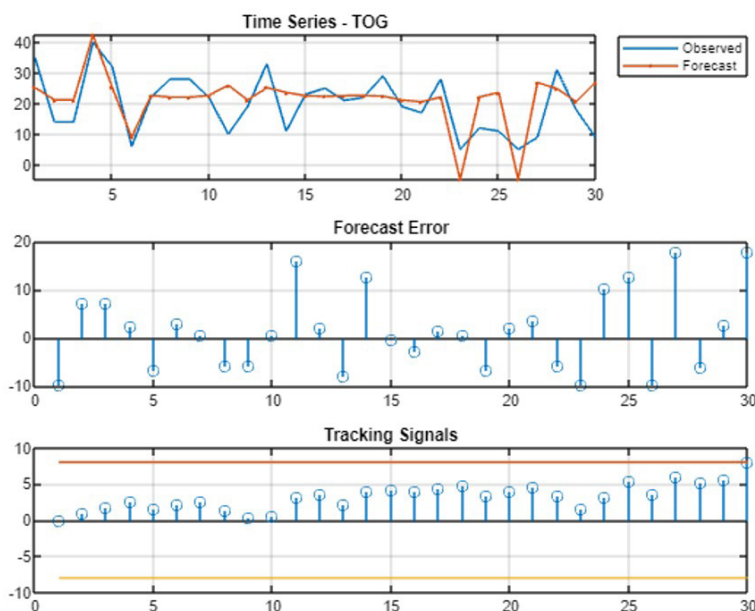


Figure 25. TOG Time Series Forecasting – Samples 55 to 84. Source: Authors.

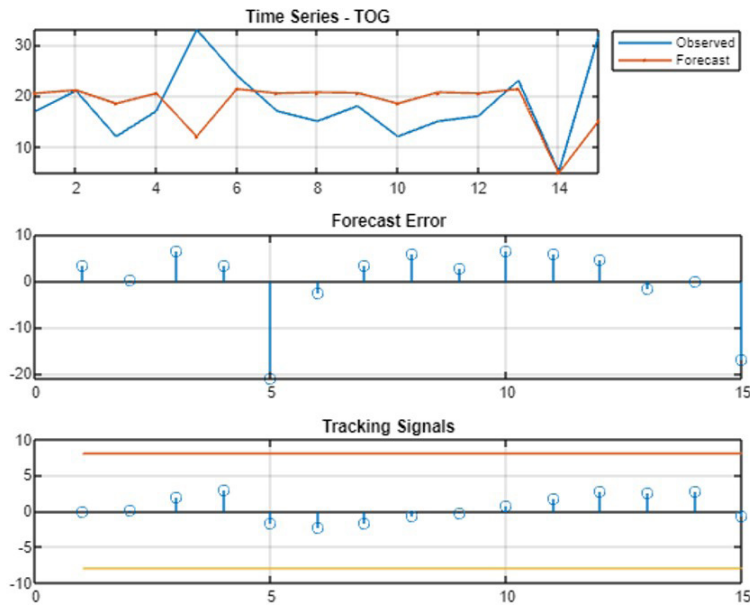


Figure 26. TOG Time Series Forecasting – Samples 85 to 100. Source: Authors.

Different from many studies that update the forecasting model for each sample or use the same model for all samples, the case study showed that the predictor was able to detect changes in the TOG data and it was necessary to update the forecasting model only twice.

## 6. Conclusion

Time series forecasting is widely used in several areas to make reasonable inferences about the future and the monitoring of forecast errors is essential to ensure forecasting accuracy. Although intelligent tools such as neural networks have been applied in time series forecasting for some time, the problem of monitoring the forecasting process has not been widely addressed. Sometimes there might not be enough current data available. Or, in some cases, as time goes by, more recent data become available while some old historical observations might distort the current structure of the system, or the existing pattern or relationships obtained from the past might not continue in the future (Deng et al., 2004). Hence, detection of any changes in the system and taking the process brought back under control is very important to ensure certain predictions and correct decisions making.

Motivated by that, this paper aims to evaluate a nonlinear time series prediction methodology using the tracking signal method to detect bias and their responsiveness to non-random changes in the time series, because the forecasting model is static once the training is over. Moreover, this paper presented a case study using TOG data to exemplify the application of the time series prediction methodology.

The study consisted of analysis based on 100 samples of 26 synthetic datasets generated to simulate different situations by changing the mean and the standard deviation of the series error for STAR1, BL1, and NMA models. Each dataset contained 100 samples of the original series and 50 samples with the error modified. The MLP was implemented in the original time series samples with the training parameters carefully set. The CUSUM tracking signals were applied in the forecast errors, and the results were plotted in control charts. Thus, the modified time series were inserted in the forecast model, and the ARL results were obtained and compared by using ANOVA with a significance level of 5%.

The results presented in this paper show that the prediction methodology presented better performance applied in the STAR1 because it was able to detect the change in the series before the other models. For STAR1 and NMA, the smallest ARL results are obtained when the mean has a high change, while the BL1 presents a smaller ARL with a moderate change in the mean. The results also show that, concerning the BL1 and NMA models, the best ARL occurs when the standard deviation has a moderate to high change while the smallest ARL for STAR1 is obtained with a small standard deviation change.

In general, the proposed prediction methodology is an effective way to detect bias in the process when an error is introduced in the nonlinear time series because the mean and the standard deviation of the series error have a significant impact on the ARL. By using this methodology, the forecasting model can be kept up-to-date and thereby improve forecast accuracy

This simulation study differs from others also concerning the method used to compare the ARL results. The DOE technique was applied to simulate the datasets by using an arrangement of experiments with controlled levels of the factors and to identify in which conditions the tracking signals are efficient. Nevertheless, other papers presented a usual approach of ARL in which a larger number of different datasets is tested and can lead to restricted conclusions without a formal and statistical analysis, which can be achieved using DOE. Another difference is the calculation of the control limits that were determined by using the false alarm rate concept, different from other papers that use a fixed control limit.

Many other studies still can be performed, for example (1) analysis of other nonlinear time series models, (2) evaluation of time series prediction methodology by using other forecasting method than MLP, (3) or by using another tracking signals method than CUSUM.

Further, this paper may be a reference for researchers looking to improve the accuracy of nonlinear time series forecasting.

## References

- Aizenberg, I., Sheremetov, L., Villa-Vargas, L., & Martinez-Muñoz, J. (2016). Multilayer Neural Network with Multi-Valued Neurons in time series forecasting of oil production. *Neurocomputing*, 175, 980-989. <http://dx.doi.org/10.1016/j.neucom.2015.06.092>.
- Amdoun, R., Khelifi, L., Khelifi-Slaoui, M., Amroune, S., Asch, M., Assaf-Ducrocq, C., & Gontier, E. (2018). The Desirability optimization methodology; a tool to predict two antagonist responses in biotechnological systems: case of biomass growth and hyoscyamine content in elicited datura starmonium hairy roots. *Iranian Journal of Biotechnology*, 16(1), e1339. <http://dx.doi.org/10.21859/ijb.1339>. PMID:30555836.
- Amiri, E. (2015). Forecasting daily river flows using nonlinear time series models. *Journal of Hydrology (Amsterdam)*, 527, 1054-1072. <http://dx.doi.org/10.1016/j.jhydrol.2015.05.048>.
- Armstrong, J. (2001). *Principles of forecasting: a handbook for researchers and practitioners*, New York: Springer Science. <http://dx.doi.org/10.1007/978-0-306-47630-3>.
- Balestrassi, P., Popova, E., Paiva, A., & Lima, J. (2009). Design of experiments on neural network's training for nonlinear time series forecasting. *Neurocomputing*, 72(4-6), 1160-1178. <http://dx.doi.org/10.1016/j.neucom.2008.02.002>.
- Bandeira, S. G., Alcalá, S. G. S., Vita, R. O., & Barbosa, T. A. (2020). Comparison of selection and combination strategies for demand forecasting methods. *Production*, 30, e20200009. <http://dx.doi.org/10.1590/0103-6513.20200009>.
- Berry, M., & Linoff, G. (1997). *Data mining techniques*. New York: John Wiley & Sons.
- Bianchesi, N., Romão, E., Lopes, M., Balestrassi, P., & Paiva, A. (2019). A design of experiments comparative study on clustering methods. *IEEE Access: Practical Innovations, Open Solutions*, 7, 167726-167738. <http://dx.doi.org/10.1109/ACCESS.2019.2953528>.
- Bischak, D. P., & Trietsch, D. (2007). The rate of false signals in  $\bar{X}$  control charts with estimated limits. *Journal of Quality Technology*, 39(1), 55-65. <http://dx.doi.org/10.1080/00224065.2007.11917673>.
- Box, G. E. P., & Jenkins, G. M. (1970). *Time series analysis: forecasting and control*. San Francisco: Holden Day.
- Brasil, Conselho Nacional do Meio Ambiente – CONAMA. (2007, August 9). *Resolution nº 393/2007*. Diário Oficial da União. Retrieved in 2022, May 13, from <http://www.brazilianr.com/brazilian-environmentallegislation/conama-resolution-39307/>
- Brence, J., & Mastrangelo, C. (2006). Parameter selection for a robust tracking signal. *Quality and Reliability Engineering International*, 22(4), 493-502. <http://dx.doi.org/10.1002/qre.724>.
- Brown, G. (1959). *Statistical forecasting for inventory control*. New York: McGraw-Hill.
- Candiotti, L. V., De Zan, M., Cámara, M., & Goicoechea, H. (2014). Experimental design and multiple response optimization: using the desirability function in analytical methods development. *Talanta*, 124, 123-138. <http://dx.doi.org/10.1016/j.talanta.2014.01.034>. PMID:24767454.
- Chan, K. S., & Tong, H. (1986). On estimating thresholds in autoregressive models. *Journal of Time Series Analysis*, 7(3), 179-190. <http://dx.doi.org/10.1111/j.1467-9892.1986.tb00501.x>.
- Chang, J., & Tseng, C. (2017). Analysis of correlation between secondary PM2.5 and factory pollution sources by using ANN and the correlation coefficient. *IEEE Access: Practical Innovations, Open Solutions*, 5, 22812-22822. <http://dx.doi.org/10.1109/ACCESS.2017.2765337>.
- Chen, T., & Chen, H. (1995). Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Transactions on Neural Networks*, 6(4), 911-917. <http://dx.doi.org/10.1109/72.392253>. PMID:18263379.
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Hillsdale: Lawrence Erlbaum Associates.
- Corzo, G., & Solomatine, D. (2007). Knowledge-based modularization and global optimization of artificial neural network models in hydrological forecasting. *Neural Networks*, 20(4), 528-536. <http://dx.doi.org/10.1016/j.neunet.2007.04.019>. PMID:17532609.
- Cui, L., Wang, Z., & Zhou, X. (2012). Optimization of elicitors and precursors to enhance valtrate production in adventitious roots of *Valeriana amurensis* Smir. ex Kom. *Plant Cell, Tissue and Organ Culture*, 108(3), 411-420. <http://dx.doi.org/10.1007/s11240-011-0052-2>.

- Dascalescu, L., Medles, K., Das, S., Younes, M., Caliap, L., & Mihalcioiu, A. (2008). Using design of experiments and virtual instrumentation to evaluate the tribocharging of pulverulent materials in compressed air devices. *IEEE Transactions on Industry Applications*, 44(1), 3-8. <http://dx.doi.org/10.1109/TIA.2007.912801>.
- De Gooijer, J. G., & Hyndman, R. J. (2006). 25 years of time series forecasting. *International Journal of Forecasting*, 22(3), 443-473. <http://dx.doi.org/10.1016/j.ijforecast.2006.01.001>.
- Deboeck, G. J. (1994). *Trading on the edge: neural, genetic, and fuzzy systems for chaotic financial markets*. New York: John Wiley & Sons.
- Deng, Y., Jaraiedi, M., & Iskander, W. (2004). Tracking signal test to monitor an intelligent time series forecasting model. *Intelligent Manufacturing*, 5263, 149-160. <http://dx.doi.org/10.1117/12.517225>.
- Derringer, G., & Suich, R. (1980). Simultaneous optimization of several response variables. *Journal of Quality Technology*, 12(4), 214-219. <http://dx.doi.org/10.1080/00224065.1980.11980968>.
- Gardner Junior, E. (1985). CUSUM vs. smoothed error forecast monitoring schemes: some simulation results. *The Journal of the Operational Research Society*, 36(1), 43-47. <http://dx.doi.org/10.1057/jors.1985.6>.
- Granger, C. W. J., & Anderson, A. P. (1978). *An introduction to bilinear time series models*. Gottingen: Vandenhoeck & Ruprecht.
- Hamilton, J. (1989). A new approach to the economic analysis of nonstationary time series and the business cycle. *Econometrica Journal of Economic Society*, 57(2), 357-384. <http://dx.doi.org/10.2307/1912559>.
- Harrington, E. (1965). The desirability function. *Ind Quality Control*, 21, 494-498.
- Haykin, S. (2009). *Neural networks and learning machines* (3rd ed.). New Jersey: Prentice Hall.
- Hippert, H. S., & Taylor, J. (2010). An evaluation of Bayesian techniques for controlling model complexity and selecting inputs in a neural network for short-term load forecasting. *Neural Networks*, 23(3), 386-395. <http://dx.doi.org/10.1016/j.neunet.2009.11.016>. PMID:20022462.
- Hornik, K. (1993). Some new results on neural network approximation. *Neural Networks*, 6(8), 1069-1072. [http://dx.doi.org/10.1016/S0893-6080\(93\)80018-X](http://dx.doi.org/10.1016/S0893-6080(93)80018-X).
- Hu, J., Wang, X., Zhang, Y., Zhang, D., Zhang, M., & Xue, J. (2020). Time series prediction method based on variant LSTM recurrent neural network. *Neural Processing Letters*, 52(2), 2. <http://dx.doi.org/10.1007/s11063-020-10319-3>.
- Igunnu, E. T., & Chen, G. Z. (2014). Produced water treatment technologies. *International Journal of Low-Carbon Technologies*, 9(3), 157-177. <http://dx.doi.org/10.1093/ijlct/cts049>.
- Kialashaki, A., & Reisel, J. (2013). Modeling of the energy demand of the residential sector in the United States using regression models and artificial neural networks. *Applied Energy*, 108, 271-280. <http://dx.doi.org/10.1016/j.apenergy.2013.03.034>.
- Krishnamurthy, B. (2006). *A comparison of the relative efficiency of tracking signals in forecast control* (Thesis). West Virginia University, West Virginia.
- Kumar, D., & Murugan, S. (2017). A novel fuzzy time series model for stock market index analysis using neural network with tracking signal approach. *Indian Journal of Science and Technology*, 10(16), 10. <http://dx.doi.org/10.17485/ijst/2017/v10i16/104994>.
- Lee, J. Y., Chang, J. H., Kang, D. H., Kim, S. I., & Hong, J. P. (2007). Tooth shape optimization for cogging torque reduction of transverse flux rotary motor using design of experiment and response surface methodology. *IEEE Transactions on Magnetics*, 43(4), 1817-1820. <http://dx.doi.org/10.1109/TMAG.2007.892611>.
- Liu, Z., Zhu, Z., Gao, J., & Xu, C. (2021). Forecast methods for time series data: a survey. *IEEE Access: Practical Innovations, Open Solutions*, 9, 91896-91912. <http://dx.doi.org/10.1109/ACCESS.2021.3091162>.
- Lorscheid, I., Heine, B. O., & Meyer, M. (2012). Opening the 'black box' of simulations: Increased transparency and effective communication through the systematic design of experiments. *Computational & Mathematical Organization Theory*, 18(1), 22-62. <http://dx.doi.org/10.1007/s10588-011-9097-3>.
- Makridakis, S., & Wheelwright, S. (1989). *Forecasting methods for management* (5th ed.). New York: John Wiley.
- Mao, S., & Xiao, F. (2019). Time series forecasting based on complex network analysis. *IEEE Access: Practical Innovations, Open Solutions*, 7, 40220-40229. <http://dx.doi.org/10.1109/ACCESS.2019.2906268>.
- Matta, C., Bianchesi, N., Oliveira, M., Balestrassi, P., & Leal, F. (2021). A comparative study of forecasting methods using real-life econometric series data. *Production*, 31, e20210043. <http://dx.doi.org/10.1590/0103-6513.20210043>.
- McClain, J. (1988). Dominant tracking signals. *International Journal of Forecasting*, 4(4), 563-572. [http://dx.doi.org/10.1016/0169-2070\(88\)90133-1](http://dx.doi.org/10.1016/0169-2070(88)90133-1).
- Mircetic, D., Rostami-Tabar, B., Nikolicic, S., & Maslaric, M. (2022). Forecasting hierarchical time series in supply chains: an empirical investigation. *International Journal of Production Research*, 60(8), 2514-2533. <http://dx.doi.org/10.1080/00207543.2021.1896817>.
- Mo, F., Shen, C., Zhou, J., & Khonsari, M. (2017). Statistical analysis of the influence of imperfect texture shape and dimensional uncertainty on surface texture performance. *IEEE Access: Practical Innovations, Open Solutions*, 5, 27023-27035. <http://dx.doi.org/10.1109/ACCESS.2017.2769880>.
- Montgomery, C. D. (2009). *Introduction to statistical quality control* (6th ed.). New York: John Wiley & Sons.
- Olson, O., Delen, D., & Meng, Y. (2012). Comparative analysis of data mining methods for bankruptcy prediction. *Decision Support Systems*, 52(2), 464-473. <http://dx.doi.org/10.1016/j.dss.2011.10.007>.
- Pant, M., & Kumar, S. (2022). Particle swarm optimization and intuitionistic fuzzy set-based novel method for fuzzy time series forecasting. *Granular Computing*, 7(2), 285-303. <http://dx.doi.org/10.1007/s41066-021-00265-3>.
- Priestley, M. (1980). State-dependent models: a general approach to nonlinear time series analysis. *Journal of Time Series Analysis*, 1(1), 47-71. <http://dx.doi.org/10.1111/j.1467-9892.1980.tb00300.x>.
- Qian, B., Xiao, Y., Zheng, Z., Zhou, M., Zhuang, W., Li, S., & Ma, Q. (2020). Dynamic multi-scale convolutional neural network for time series classification. *IEEE Access: Practical Innovations, Open Solutions*, 8, 8. <http://dx.doi.org/10.1109/ACCESS.2020.3002095>.
- Ravi, P. (2014). An analysis of a widely used version of the CUSUM tracking signal. *The Journal of the Operational Research Society*, 65(8), 1189-1192. <http://dx.doi.org/10.1057/jors.2013.50>.

- Ray, J., & Engelhardt, F. R. (1992). *Produced water: technological environmental issues and solutions*. New York: Plenum Press. <http://dx.doi.org/10.1007/978-1-4615-2902-6>.
- Rosenthal, R. (1994). Parametric measures of effect size. In H. Cooper and L. V. Hedges (Eds.), *The Handbook of research synthesis* (pp. 231–244). New York: Russell Sage Foundation.
- Sabeti, H., Al-Shebeeb, O., & Jaridi, M. (2016). Forecasting system monitoring under non-normal input noise distributions. *Journal of Industrial Engineering and Management*, 5(2), 1000194.
- Santos, C. H., Lima, R. D. C., Leal, F., Queiroz, J. A., Balestrassi, P. P., & Montevechi, J. A. B. (2020). A decision support tool for operational planning: a Digital Twin using simulation and forecasting methods. *Production*, 30, e20200018. <http://dx.doi.org/10.1590/0103-6513.20200018>.
- Sun, K., Huang, S., Wong, D., & Jang, S. (2017). Design and application of a variable selection method for multilayer perceptron neural network with LASSO. *IEEE Transactions on Neural Networks and Learning Systems*, 28(6), 1386–1396. <http://dx.doi.org/10.1109/TNNLS.2016.2542866>. PMID:28113826.
- Superville, C. (2019). Tracking signal performance in monitoring manufacturing processes. *Journal of Business and Management*, 21, 23–28.
- Tong, H. (1978). On a threshold model. In C. H. Chen (Ed.), *Pattern recognition and signal processing*. Amsterdam: Sijhoff & Noordhoff. [http://dx.doi.org/10.1007/978-94-009-9941-1\\_24](http://dx.doi.org/10.1007/978-94-009-9941-1_24).
- Trigg, W. (1964). Monitoring a forecasting system. *Operational Research Quarterly*, 15(3), 271–274. <http://dx.doi.org/10.1057/jors.1964.48>.
- Tsay, R. (2005). *Analysis of financial time series* (2nd ed.). Hoboken: Wiley. <http://dx.doi.org/10.1002/0471746193>.
- Verma, P., Reddy, S. V., Ragha, L., & Datta, D. (2021). Comparison of time-series forecasting models. In *2021 International Conference on Intelligent Technologies (CONIT)*. New York: IEEE. <http://dx.doi.org/10.1109/CONIT51480.2021.9498451>.
- Wang, Z., & Lou, Y. (2019). Hydrological time series forecast model based on wavelet de-noising and ARIMA-LSTM. In *3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (pp. 1697–1701). New York: IEEE.
- Wong, W. K., Xia, M., & Chu, W. C. (2010). Adaptive neural network model for time-series forecasting. *European Journal of Operational Research*, 207(2), 207. <http://dx.doi.org/10.1016/j.ejor.2010.05.022>.
- Xiao, D., Shi, H., & Wu, D. (2012). Short-term load forecasting using bayesian neural networks learned by Hybrid Monte Carlo algorithm. *Applied Soft Computing*, 12(6), 1822–1827. <http://dx.doi.org/10.1016/j.asoc.2011.07.001>.
- Xiao, H., Jiang, X., Chen, C., Wang, W., Wang, C., Ali, A., Berthe, A., Moussa, R., & Diaby, V. (2020). Using time series analysis to forecast the health-related quality of life of post-menopausal women with non-metastatic ER+ breast cancer: a tutorial and case study. *Research in Social & Administrative Pharmacy*, 16(8), 1095–1099. <http://dx.doi.org/10.1016/j.sapharm.2019.11.009>. PMID:31753693.
- Yang, M. (2011). Measurement of oil in Produced Water. In K. Lee & J. Neff (Eds.), *Produced water* (pp. 57–88). New York: Springer. [http://dx.doi.org/10.1007/978-1-4614-0046-2\\_2](http://dx.doi.org/10.1007/978-1-4614-0046-2_2).
- Yu, L., & Lai, K. (2005). Adaptive smoothing neural networks in foreign exchange rate forecasting. In *International Conference on Computational Science* (pp. 523–530). Berlin: Springer. [http://dx.doi.org/10.1007/11428862\\_72](http://dx.doi.org/10.1007/11428862_72).
- Zhai, X., Ali, A., Amira, A., & Bensaali, F. (2016). MLP neural network based gas classification system on Zynq SoC. *IEEE Access: Practical Innovations, Open Solutions*, 4, 8138–8146. <http://dx.doi.org/10.1109/ACCESS.2016.2619181>.