

Article

Predicting Thermoelectric Power Plants Diesel/Heavy Fuel Oil Engine Fuel Consumption Using Univariate Forecasting and XGBoost Machine Learning Models

Elias Amancio Siqueira-Filho ^{1,2}, Maira Farias Andrade Lira ², Attilio Converti ³, Hugo Valadares Siqueira ^{4,*}
and Carmelo J. A. Bastos-Filho ¹

¹ Polytechnic School of Pernambuco, Pernambuco University, Benfica St., 455, Madalena, Recife 50720-001, Brazil

² Advanced Institute of Technology and Innovation (IATI), Potyra St., 31, Prado, Recife 50751-310, Brazil

³ Department of Civil, Chemical and Environmental Engineering, Pole of Chemical Engineering, University of Genoa, Via Opera Pia 15, 16145 Genoa, Italy

⁴ Department of Electronics, Federal University of Technology—Paraná (UTFPR), Dr. Washington Subtil Chueire St., 330, Jardim Carvalho, Ponta Grossa 84017-220, Brazil

* Correspondence: hugosiqueira@utfpr.edu.br

Abstract: Monitoring and controlling thermoelectric power plants (TPPs) operational parameters have become essential to ensure system reliability, especially in emergencies. Due to system complexity, operating parameters control is often performed based on technical know-how and simplified analytical models that can result in limited observations. An alternative to this task is using time series forecasting methods that seek to generalize system characteristics based on past information. However, the analysis of these techniques on large diesel/HFO engines used in Brazilian power plants under the dispatch regime has not yet been well-explored. Therefore, given the complex characteristics of engine fuel consumption during power generation, this work aimed to investigate patterns generalization abilities when linear and nonlinear univariate forecasting models are used on a representative database related to an engine-driven generator used in a TPP located in Pernambuco, Brazil. Fuel consumption predictions based on artificial neural networks were directly compared to XGBoost regressor adaptation to perform this task as an alternative with lower computational cost. AR and ARIMA linear models were applied as a benchmark, and the PSO optimizer was used as an alternative during model adjustment. In summary, it was possible to observe that AR and ARIMA-PSO had similar performances in operations and lower error distributions during full-load power output with normal error frequency distribution of -0.03 ± 3.55 and 0.03 ± 3.78 kg/h, respectively. Despite their similarities, ARIMA-PSO achieved better adherence in capturing load adjustment periods. On the other hand, the nonlinear approaches NAR and XGBoost showed significantly better performance, achieving mean absolute error reductions of 42.37% and 30.30%, respectively, when compared with the best linear model. XGBoost modeling was 8.7 times computationally faster than NAR during training. The nonlinear models were better at capturing disturbances related to fuel consumption ramp, shut-down, and sudden fluctuations steps, despite being inferior in forecasting at full-load, especially XGBoost due to its high sensitivity with slight fuel consumption variations.

Keywords: time series forecasting; power plants; fuel consumption; Box & Jenkins methodology; XGBoost; artificial neural networks



Citation: Siqueira-Filho, E.A.; Lira, M.F.A.; Converti, A.; Siqueira, H.V.; Bastos-Filho, C.J.A. Predicting Thermoelectric Power Plants Diesel/Heavy Fuel Oil Engine Fuel Consumption Using Univariate Forecasting and XGBoost Machine Learning Models. *Energies* **2023**, *16*, 2942. <https://doi.org/10.3390/en16072942>

Academic Editor: Alan Brent

Received: 23 February 2023

Revised: 17 March 2023

Accepted: 21 March 2023

Published: 23 March 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The Brazilian electrical matrix has good energy source diversity, particularly renewable sources. Hydroelectric energy is the primary energy source responsible for supplying the National Integrated System (SIN, in the Brazilian acronym), despite solar and wind generation growth in recent years [1]. As most of the Brazilian power grid is supplied by

energy sources impacted by rainfall regime and other climate variations, there is a need to activate thermoelectric power plants (TPPs) powered by fossil fuels to ensure SIN energy supply stabilization and flexibility [2].

The National System Operator (ONS, in the Brazilian acronym) and the Electric Energy Trading Chamber (CCEE, in the Brazilian acronym) implemented the Very Short-Term Hydrothermal Dispatch Model (DESSEM, in the Brazilian acronym). This model aims to monitor each power-generating unit individually in real-time and program each operation in semi-hourly granularity with a one-week forecast horizon to improve Brazilian hydrothermal system planning operation efficiency and optimal generation per plant [3,4]. Therefore, power generation agents are responsible for ensuring TPP availability on a daily dispatch schedule, which has made it increasingly challenging to guarantee emergency energy generation proposed by DESSEM [5,6].

The Brazilian electrical energy supply system often requires fast TPP power output generation changes and uninterrupted operation for long periods. Thus, the general reliability of TPP engines powered by diesel/HFO can be impacted by a lack of ideal operational maintenance [7]. One way to help the energy-generation monitoring process is to use intelligent tools based on data provided in real-time [8–10]. Although most TPPs have monitoring platforms that capture data from various sensors to monitor power generation system health, secondary data-based decision aid tools are yet to be fully explored [11,12]. In this sense, time series forecasting techniques prove to be an exciting alternative to analyze possible very short-term generation scenarios and assist TPP' operation and maintenance planning, especially on diesel/HFO TPPs, which usually operate on an availability basis and require quick actions to start the engine-driven generators and power generation.

Prediction models focused on observing energy generation patterns have been explored in several types of systems [13–18]. Typically, linear forecasting models are used as a benchmark during this analysis due to their ease of implementation and coefficient determination simplicity. Among those traditionally used are statistical models of the Box & Jenkins methodology [19,20]. However, despite being widely discussed in the literature, the coefficient calculation is often done using statistical tools such as maximum likelihood estimators that need to be improved to obtain models' optimal values. An alternative to this problem is bio-inspired computational intelligence techniques such as particle swarm optimization (PSO) [21–23].

Alternatively, given that these models can only assimilate linear series characteristics and real-world problems have both linear and nonlinear patterns that can be correlated with past values [24], several machine learning techniques are being used to predict more complex behaviors. Among them, artificial neural networks (ANN) such as multilayer perceptron networks (MLP) are pretty efficient in capturing the nonlinearity caused by sudden variations, as well as observing the influence of a plant's internal parameters on output parameters [11,13,14,25,26]. Similarly, reinforcement techniques based on decision trees (DT) have lately been explored in classification and regression problems such as eXtreme Gradient Boosting (XGBoost) that can also be applied to time series forecasting problems to capture complex patterns [27,28].

Different time series forecasting techniques have recently been developed to perform increasingly accurate observations. TPP energy generation, fuel consumption, and gas emissions prediction are discussed in a wide range of works that apply the Box & Jenkins methodology as the primary analysis tool for short- and very short-term horizons. Autoregressive integrated and moving average (ARIMA) and infinite impulse response filters (IIR) recursive models adjustment capacity were studied by Siqueira et al. [29]. Higher generalization was achieved using the genetic algorithm (GA), PSO, evolutionary differentiation (ED), CLONALG, and Opt-aiNet metaheuristics compared with non-recursive coefficients adjustment methods. Simulations proved superiority of recursive models in all scenarios when compared with the autoregressive (AR) model. This research did not determine which algorithm was the most suitable for the problem since ED, GA, and PSO achieved minimum mean-squared error values during the tests. This improvement can also

be seen in the study of Rusli et al. [30], who used ARIMA-PSO modeling to predict TPP coal consumption to carry out fuel stock planning. The ARIMA-PSO model application increased daily, weekly, and monthly coal need prediction precision by reducing the mean absolute percentage error (MAPE) by 4.34%, 4.91%, and 6.17%, respectively, when compared with the ARIMA model. However, despite the application of different optimizers for linear forecasting models adjustment, Siqueira et al. [29] and Rusli et al. [30] did not directly compare the metaheuristic application with nonlinear forecasting techniques that can achieve more accurate results depending on the analyzed series behavior.

PSO and GA bio-inspired algorithms can also be found in the study developed by Xu et al. [6], who analyzed coal-fired TPP load dispatch. From historical data, it was possible to: determine key performance indicators through the grey correlation method, forecast each unit-specific consumption based on PSO and support vector machine (SVM) hybrid modeling regression, and determine optimal dispatch distribution in each of the generators based on GA. The hybrid modeling proposed by Xu et al. [6] achieved a maximum coal saving capacity of 7.94 g/kWh when compared with power grid control through automatic generation control (AGC). Although this work used continuously measured supervision information system data from an actual coal-fired TPP, the analyzed model did not investigate each parameter's temporal influence on coal consumption prediction. For diesel/HFO TPPs unit electricity cost predictions, it is possible to observe that multivariate models based on auxiliary information, such as multiple linear regression (MLR) with each parameter temporal influence analysis, can achieve better performance when compared with univariate modelings if the most significant inputs are selected to avoid multicollinearity [31]. The developed time series regression model proposed by Weerasinghe and Jayasundara [31] achieved drastic error metrics reduction with at least 73.69% improvement in root mean square error (RMSE) and 78.47% in mean absolute error (MAE) when compared with the ARIMA univariate model. This significant improvement ensured by the multivariate model can be related to the strong influence of exogenous variables on the unit electricity cost values. However, given diesel/HFO engines' mechanical stationarity nature, univariate forecasting models application to investigate parameters such as fuel consumption can be a satisfactory alternative for maintenance management.

Nonlinear forecasting techniques are commonly used to capture more complex patterns than linear models [32]. The application of machine learning-based predictors can be seen in the work of Tuttle et al. [33] for NO_x gases prediction, whose ensemble modeling considers exchangeable weights ANN with hyper-parameters optimization through GA metaheuristic and PSO application to optimize system combustion parameters dynamically. Additionally, other well-established machine learning methods such as SVM, random forest (RF), and kernel partial least squares (KPLS) were directly compared with ANN hybrid modeling to determine the most suitable technique. The application of ANN in the closed-loop operation of combustion optimization systems for NO_x emission prediction proved to be superior due to the ability to accurately predict NO_x emission rates not evaluated in the training set. Additionally, Tuttle et al. [33] achieved a 22.5% reduction in the NO_x emission rate when applying the developed system in real-time during a year and a half in a coal-fired TPP, which reinforces the capacity of artificial neural networks in capturing typically non-linear patterns. Analysis of internal combustion engine monitoring parameters has lately been carried out [34]. Singh et al. [35] studied a direct injection engine powered by different proportions of biodiesel, injection times, and air/fuel ratio. This analysis used MLP to provide systems thermal efficiency and emission levels predictions. Harris Hawks (HH) and whale optimization algorithm (WOA) metaheuristics were used to determine the most significant ANN inputs. However, these techniques were not compared with well-established methods of determining ANN's hyper-parameters, such as grid-search or random-search, which can be computationally more efficient when working with a large amount of data and more robust models. Additionally, it is worth mentioning the work of Yıldırım et al. [36], in which the engine response to different types of enriched hydrogen gas fuel was analyzed using the ANN and support vector regression (SVR) techniques.

This work compared vibration, noise, CO, CO₂, and NO_x estimation performed by ANN and SVR models with experimental results collected from a small diesel engine. From this comparison, the authors concluded that the ANN allowed a lower overall MAPE, achieving an average difference of 53.70% compared with the SVR application. It was possible to observe the ANN's potential to generalize parameters inherent to diesel engine operation, which can be extended to investigate its performance on large-scale systems.

Machine learning algorithms based on DT have also been recently explored for heavy mechanical equipment regression problems as alternative models for capturing nonlinear patterns [27,28,37–39]. These models have architectures with better results interpretability and usually significantly reduced computational adjustment costs due to the parallel and distributed processing capacity [37]. Papandreou and Ziakopoulos applied regression models such as multivariate polynomial regression (MPR), ANN, and the XGBoost decision tree reinforcement method to determine the fuel consumption of large-scale crude oil conveyors based on information collected by sensors and path meteorological conditions [27]. XGBoost fuel consumption prediction exhibited better error metrics achieving 86.14% custom accuracy when compared to ANN (73.40%) and MPR (74.05%). Furthermore, ANN took approximately 40 times more than XGBoost to generate an equal number of models during training. This work developed a methodology to apply XGBoost's popular machine learning algorithm to predict fuel consumption achieving better results when compared to well-known techniques such as ANN.

Open-pit mine trucks fuel consumption regression based on external parameters was performed by Wang et al. [37] using different types of machine learning algorithms such as K-Nearest Neighbor (KNN), SVR, ANN, RF, and XGBoost. It was possible to determine the key features for predicting mining trucks' fuel consumption. Although SVR and XGBoost algorithms were the most effective models in pattern abstraction, the XGBoost approach was computationally faster with R² and MAPE by 0.93 and 8.78%, respectively. Additionally, Hu et al. [38] developed a hybrid GRU-XGB estimator to perform CO, CO₂, HC, and NO_x gas emission forecasts. This model initially used a double-layer gated recurrent unit (GRU) ANN responsible for analyzing historical emission data and transforming them into an encoded feature. This parameter was then fed into XGBoost estimator with other external factors, such as ambient temperature and humidity, speed, acceleration, and road conditions to generate gas predictions for two bus lines. GRU-XGB modeling captured more complex emission prediction patterns when compared to individually applied models, with an average MAPE of 3.84% for diesel-powered buses. Additionally, the XGBoost algorithm application ensured feature importance analysis for each external parameter. Nevertheless, the XGBoost modeling was applied as a regressor in this work that sought to relate the GRU encoded attribute (generated by historical emission past values) with real-time exogenous variables; it was then possible to explore the XGBoost estimator generalization potential for a variable of interest with temporal characteristics.

Although different time series forecasting techniques are applied in coal, natural gas TPPs systems, and various small/medium engine applications, their implementations on large diesel/HFO engines used for electric power generation have not yet been well-explored. Therefore, this work sought to investigate linear (AR and ARIMA-PSO) and nonlinear (MLP and XGBoost) forecasting methods and identify their ability to capture patterns during several operations of a large Brazilian TPP engine-driven generator. The identification and estimation of ARIMA model parameters were optimized using PSO modules, given the improvement seen in the literature with such a metaheuristic for this application. The AR model was used and estimated by Yule-Walker equations as a benchmark. As for the analysis of capturing nonlinear patterns, autoregressive neural networks (NAR) based on the MLP architecture were applied due to their high capacity to capture more complex nonlinear patterns. The application of the XGBoost algorithm adapted for forecasting problems was directly compared to the performance of other applied models as an alternative with a lower computational cost to perform this task.

This work is structured as follows: in Section 1, an introduction discussing the application of time series forecasting techniques and related works is provided; Section 2 presents the applied Box & Jenkins and PSO methods for linear predictions, and NAR and XGBoost fundamentals for nonlinear predictions; Section 3 covers data collection and pre-processing, AR and ARIMA-PSO models estimation, NAR and XGBoost models learning process, and the considered evaluation metrics; in Section 4, univariate model prediction is applied to the consumption of each fuel; and in Section 5, conclusions about model performance are presented.

2. Forecasting Methods

2.1. Box & Jenkins Methods

Autoregressive integrated moving average (ARIMA) models are a class of statistical models widely used for time series analysis developed by Box & Jenkins [19,20]. The first model class translates the regression of the variable of interest with its past values. This approach, called the autoregressive model AR(p), is expressed by Equation (1):

$$\hat{z}_t = \sum_{i=1}^p \phi_i z_{t-i} + a_t \quad (1)$$

where \hat{z}_t represents the forecasts of the variable of interest over time t as a function of the coefficients ϕ associated with each of the p time steps and a random term a_t which corresponds to a prediction error called white noise [40].

In order to use this class of models, the data must be stationary. Additionally, one advantage of using this type of model is that the optimal coefficients ϕ can be analytically estimated using the Yule-Walker equations. These equations are based on the relationship between the p autocorrelations ($\rho_1, \rho_2, \dots, \rho_p$) values and the p coefficients ($\phi_1, \phi_2, \dots, \phi_p$). With autocorrelations estimation (r_1, r_2, \dots, r_p), it is possible to determine AR model coefficients through Equation (2) [19]:

$$\begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_p \end{bmatrix} = \begin{bmatrix} 1 & r_1 & \cdots & r_{p-1} \\ r_1 & 1 & \cdots & r_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p-1} & r_{p-2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_p \end{bmatrix} \quad (2)$$

The moving average model MA(q) consists of the combination of q random shocks a_t related to their respective coefficients θ to determine the future values of the series \hat{z}_t , as shown in Equation (3). Unlike the AR(p) model, determining the q coefficients becomes a nonlinear estimation process.

$$\hat{z}_t = a_t + \sum_{j=1}^q \theta_j a_{t-j} \quad (3)$$

At last, the ARMA (p, q) class of models can be considered a combination approach between the previously mentioned models. Mathematically, it consists of a linear combination of p past values of z_t and q past random shocks. It is also associated with a random error a_t related to patterns not captured in each of the predictions, as shown in Equation (5). If the analyzed time series Y_t contains a clear trend pattern that can change the unconditional joint probability over time (mean and variance not constant throughout observations), it is necessary to carry out a differentiation process with order d shown in Equation (4) to transform the series into a stationary one, this being the integration term [19]. The used model in the differentiated and stationary time series is called ARIMA (p, d, q).

$$z_t = Y_t - Y_{t-1} \quad (4)$$

$$\hat{z}_t = \sum_{i=1}^p \phi_i z_{t-i} + \sum_{j=1}^q \theta_j a_{t-j} + a_t \tag{5}$$

The idea of applying this method is to consider that the past observations of a given stochastic time series contain linear information about its future. Finally, ARIMA models are flexible and can be used in a more complex time series, ideally focusing on capturing linear patterns.

2.2. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a stochastic optimization method proposed by Kennedy and Eberhart in 1995 [41] that seeks the best solution using agents, named particles. The position of each particle is a potential solution to a problem. Iteratively, each particle’s movement in the space of possibilities is updated based on its velocity at instant t . This update considers the best individual solution found by the particle $pbest_{id}$ in dimension d and according to the best position found by particles in its neighborhood $gbest_d$. The neighborhood, which is individually defined for each particle, represents a subset of the swarm with which the particle i can communicate. In general, the velocity v_{id} and position x_{id} of each particle can be described mathematically over the iterations for a continuous search space. A constraint factor χ is related to the velocity update. From values $\chi > 4$ in Equation (6), the risk of having premature convergences for those related to the local suboptimal is considerably reduced [42]. Then, continuous velocity and position updates can be carried out through Equations (7) and (8), respectively [43]:

$$\chi = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \text{ being } \varphi = c_1 + c_2 \tag{6}$$

$$v_{id}^{t+1} = \chi [v_{id}^t + c_1 \tau_1 (pbest_{id}^t - x_{id}^t) + c_2 \tau_2 (gbest_d^t - x_{id}^t)] \tag{7}$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}, d = 1, 2, \dots, n; i = 1, 2, \dots, m \tag{8}$$

where i and d represent the analyzed particle and the current dimension, respectively, t is the iteration counter, c_1 and c_2 are the acceleration constraints, τ_1 and τ_2 are randomly generated values between 0 and 1.

The continuous-space PSO algorithm must be adapted when the optimization problem produced in a discrete space of alternatives. Rapaic et al. [44] suggested the discrete particle swarm optimization algorithm (DPSO), in which the next iteration particle velocity v_{id}^{t+1} undergoes a transformation process after the previous calculation through a hyperbolic tangent function. This process aims to obtain a new saturated velocity that varies between -1 and 1 , as shown in Equation (9):

$$v_{id}^{t+1} = \frac{1 - \exp v_{id}^{t+1}}{1 + \exp v_{id}^{t+1}} = \tanh\left(\frac{v_{id}^{t+1}}{2}\right) \tag{9}$$

The particle’s new position x_{id}^{t+1} is then calculated through Equation (10):

$$x_{id}^{t+1} = x_{id}^t + \left[v_{id}^{t+1} \Delta x_{max} \right] \tag{10}$$

where Δx_{max} is a new model parameter that indicates the maximum allowed particle displacement.

2.3. Artificial Neural Networks (ANN)

The use of nonlinear models for time series forecasting problems presupposes that past series values must contain linear and nonlinear information, and complex information about its future. This class of discrete models is represented mathematically by

Equation (11), where $y_t \in \mathbb{Z}$ represents the model input at time t . The delay term h indicates how many steps are taken into account to make the predictions [45,46].

$$y_{t+1} = f[y_t, \dots, y_{t-h}] \quad (11)$$

The mapping function $f(\cdot)$ can be approximated using different approaches. It is possible to use artificial neural networks (ANN) to perform this generalization, named nonlinear autoregressive neural networks (NAR) for univariate cases. The biological inspiration comes from how the neurons exchange information in the nervous systems of superior organisms. In summary, the intelligent behavior of the ANN comes from the interactions among the several processing units (artificial neurons) capable of relating the observed data (inputs) with the actual outputs by weights and biases [36]. Mathematically, each neuron output y_k is given by a weighted sum of n inputs x_i with their respective weights w_{ki} and biases b_k as shown generically in Equation (12):

$$y_k = f(u_k) = f\left(\sum_{i=1}^n w_{ki}x_i + b_k\right) \quad (12)$$

The variable f is the activation function. Usually, a nonlinear function is responsible for inserting nonlinearity in the output response. A well-known fully connected feedforward ANN model class is the multilayer perceptron (MLP) due to training simplicity through the backpropagation algorithm. This model comprises an input layer, one or more hidden layers, and one or more neurons in the output layer. The hidden layers are used to determine the system characteristics, in which the weights are obtained by encoding the input characteristics. The output layer aims to receive all the information processed through the intermediate layers and build the calculated response pattern [25]. Several optimization algorithms can be used for the supervised learning process. For example, the Adam optimization algorithm can be used for stochastic gradient descent optimization purposes in large dataset applications where training speed can be considerably reduced [47,48].

2.4. Extreme Gradient Boosting (XGBoost)

Extreme gradient boosting (XGBoost) is a machine learning method implemented through an optimized library created by Chen and Guestrin [49] based on gradient boosting decision trees (GBDT) algorithm. This method aims to expand the knowledge behind GBDT techniques for regression and classification problems with scarcer data and scale the applicability of XGBoost for supervised problems with higher dimensionality. The efficiency and quick application of this method can be justified due to the parallel and distributed processing capacity, which allows more agile model explorations. The algorithm seeks to combine several subsequent secondary predictors (for example, decision trees (DT)) to make more robust predictions about the system under analysis. The relationship between each DT is given through residual predictions estimated by the immediately preceding DT. Iteratively, new trees are created, trained, and tuned to minimize the cumulative residual error. In summary, the final prediction made through the XGBoost algorithm consists of the sum of the first prediction with the residual predictions estimated by each created tree [50].

Despite being an algorithm initially created for classification and regression problems, it is possible to use it for time series forecasting analysis. For this task, the data must be transformed into a tabular input as previously presented in Section 2.3. Figure 1 illustrates this type of application, in which the dataset is used as input for the predictor and y_t represents the model input at time t . Mathematically, considering a nonlinear function capable of mapping the output \hat{y}_t based on a vector of past values Y , a set of DT models consists of the addition of several other weaker nonlinear DT predictors, $\phi(\cdot)$, to map more complex patterns after the union, as shown in Equation (13):

$$\hat{y}_t = f[Y] = \sum_{k=1}^K \phi_k[Y] \quad (13)$$

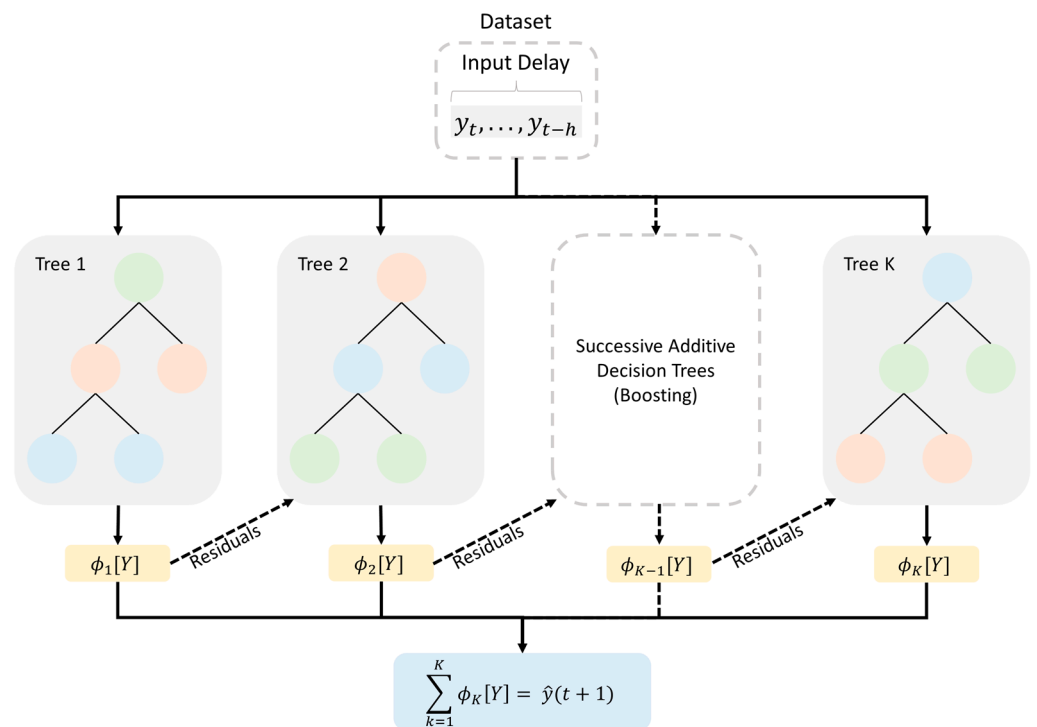


Figure 1. XGBoost modeling scheme applied to time series forecasting problems considering h input delay time steps, K additive residual decision trees $\phi(\cdot)$ to predict $\hat{y}(t + 1)$ values.

The quadratic error is commonly used to adjust the model between each learning iteration. Additionally, other parameters were considered during the creation of the regularized objective function $\mathcal{L}(f)$, such as a penalty term $\Omega(\phi)$ that controls the complexity of the model and also avoids overfitting. This term considers γ and λ penalty coefficients, as well as the maximum number of leaves allowed in the DT model, T , and a weight associated with each leaf, c , as shown in Equation (14). Therefore, it is possible to consider the regularized objective function shown in Equation (15) [27].

$$\Omega(\phi) = \gamma T + \frac{\lambda c^2}{2} \tag{14}$$

$$\mathcal{L}(f) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_k \Omega(\phi_k) \tag{15}$$

The XGBoost training process is similar to the methodology followed by GBDT, in which the model adjustment is additively taken into account [51]. However, to reduce the error associated with the predictions, the objective function $\mathcal{L}(f)$ is updated for each one of t iterations by adding a decision tree $\phi_t[Y]$. The final model after the last tree additions can be seen in Equation (16), in which $\hat{y}_i^{(t-1)}$ is the adjusted model until $t - 1$ iteration and \hat{y}^t is the updated model.

$$\hat{y}^t = \hat{y}_i^{t-1} + \phi_t[Y] \tag{16}$$

The modification made in the model structure directly implies an alteration of the objective function $\mathcal{L}(f)$ that must be controlled for each t iteration. Thus, the training process can go on using the new cost function described in Equation (17), which considers the penalty described in Equation (14) and previously mentioned new model.

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + \phi_t[Y]) + \Omega(\phi_t) \tag{17}$$

Chen and Guestrin [49] argued that, due to the composition of the model proposed in Equation (15) with different functions and penalty parameters, Euclidean space traditional

optimization methods cannot be used directly. Therefore, for the minimization of the objective function $\mathcal{L}(f)$, it is necessary to apply a second-order approximation of Taylor’s theorem so that it is possible to transform the non-Euclidean domain into a real vector space. Hence, considering $g_i = \partial_{(\hat{y}_{t-1})} l(y_i, \hat{y}^{(t-1)})$ the first derivative of the error between predicted observations and reals in the iteration $t - 1$ and $h_i = \partial_{(\hat{y}_{t-1})}^2 l(y_i, \hat{y}^{(t-1)})$ the second-order derivative, the objective function to be minimized for iteration t can be approximated through Equation (18):

$$\mathcal{L}^{(t)} = \sum_{i=1}^n \left(l(y_i, \hat{y}_i^{(t-1)}) + g_i \phi_t[Y] + \frac{1}{2} h_i \phi_t^2[Y] \right) + \Omega(\phi_t) \tag{18}$$

The training stage goal will consist of finding an additional $\phi_t([Y])$ decision tree in the t iteration that is capable of minimizing the term shown in Equation (19), where I_j represents the instances that belong to leaf j of the analyzed decision tree:

$$\operatorname{argmin}_{\phi_t[Y]} - \frac{1}{2} \sum_{j=1}^T \left(\frac{(\sum_{i \in I_j} g_i)^2}{\sum_{i \in I_j} h_i + \lambda} \right) + \gamma T \tag{19}$$

The construction process of each tree takes into account a greedy approach. Initially, the analyzed decision tree has a structure with a depth equal to zero. In other words, all residuals are grouped into a single set called the root. After the first stage, for each extra added node, the algorithm tries to find a division that guarantees the analyzed error minimization. The cost function value before the division is compared with the magnitude of the error after division through Equation (19), and the gain can be calculated, as described by Equation (20), from the difference between the previously determined values and consideration of the regularizer γ to avoid divisions that deteriorate the generalization capacity of the final model.

$$\text{Gain} = \frac{1}{2} \left[\frac{G_D^2}{H_D + \lambda} + \frac{G_E^2}{H_E + \lambda} - \frac{(G_D + G_E)^2}{H_D + H_E + \lambda} \right] - \gamma \tag{20}$$

This formula consists of a term relative to the score on the new left leaf (G_E, H_E) , a score that considers the new right leaf (G_D, H_D) , the previous score on the original leaf $(G_D + G_E, H_D + H_E)$ if there was no division and in the regularization term in the additional leaf (γ) previously mentioned. Suppose the gain is less than zero with the addition of the analyzed branch. In that case, there is no benefit in further increasing the complexity of the model with this modification, being an algorithm with supervised training with weights adjustment that considers a learning rate ϵ . The calculation of the final forecast based on a given set of inputs will consist of the sum of the initial prediction \hat{y}_0 with each tree DT residuals output multiplied by the learning rate ϵ and with the last prediction based on the first order gradient g_i and the hessian h_i also associated with the learning rate. If the number of iterations has yet to be reached, a new DT_k tree is adjusted and added to the predictions.

$$\hat{y}^t(t + 1) = \hat{y}_0 + \epsilon DT_1 + \epsilon DT_2 + \dots + \epsilon \frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \tag{21}$$

Several hyper-parameters seek to regulate tree growth to guarantee a greater model abstraction capacity and avoid overfitting during the learning phase. In addition to the penalty coefficients λ, γ , L1 type regularization α and learning rate ϵ [49], the following hyper-parameters can be applied to the model:

- Maximum depth limits the maximum allowed size for each tree added to the model. It is noteworthy that the larger the tree size, the greater abstraction capacity the model will have, being more prone to overfitting;
- Minimum child weight needed to split the tree in the next iteration;
- Subsample, which represents the proportion of randomly collected instances selected as model input before starting the tree-building process;
- Columns Sample by Tree, which also represents the percentage of features randomly collected to be used to build decision trees;
- Number of estimators that represents the number of decision trees to be fitted and, in other words, the number of iterations of the algorithm;
- Maximum delta step allowed by each tree that avoids the case where weights become infinitely large, mainly in unbalanced data, where the learning rate ϵ would not be enough to control the magnitude of the correction.

These hyper-parameters will be considered during the adjustment of the XGBoost model to carry out fuel consumption predictions. The parameter determination procedure and model performance analysis will be described in Section 3.3.

3. Material and Methods

This research evaluated univariate prediction models applied to fuel consumption series related to one of the 17 Wärtsilä 20V46F engines available in Energética Suape II S.A. TPP powered by diesel oil/HFO, with total installed capacity of 381.2 MW, which is located in Pernambuco, Brazil. The analysis goal was to investigate univariate forecast models' generalization capacity when applied to fuel consumption series based on training and data collection referring to 6 months of operation.

3.1. Data Acquisition, Cleaning and Normalization

The data collection step captured fuel consumption information read by a flowmeter sensor placed before the engine-driven generator intake branch. Figure 2 shows the physical data acquisition and control system used in this analysis, consisting of a flowmeter (Figure 2a) and local control panel (Figure 2b). From the monitoring platform, it was possible to obtain fuel consumption data in kilograms per hour every 7 to 10 s of operation related to the plant engine, with the highest number of accumulated hours during the analyzed period. Additionally, it is essential to consider that during the selected timeframe (31 May 2021 to 4 December 2021), this TPP had a higher dispatch level due to the then-current Brazilian energy and pluvial scenario.

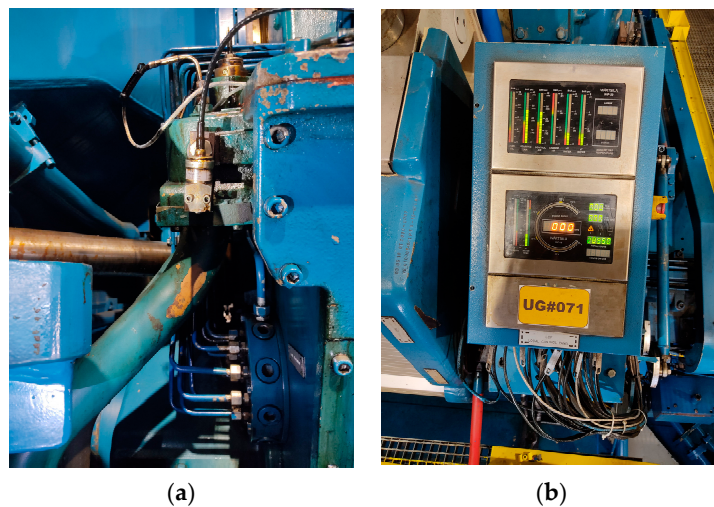


Figure 2. Energética SUAPE II S.A. TPP data acquisition system available for each engine-driven generator: (a) fuel consumption flowmeter placed before the engine-driven generator intake branch; (b) local control panel responsible for collecting various engine-related signals.

A total of 2,053,542 million fuel consumption observations were collected for this study. After data collection, each non-captured value was identified due to possible errors in the data acquisition step and outliers. We identified 35,815 nonconforming samples and used linear interpolations between available nearest-neighbor values to replace them. To guarantee temporal uniformity in data, a fuel consumption series resizing was carried out in groups of signals considering the average of the minute under analysis since no considerable changes were observed when smaller temporal granularities were analyzed.

After data cleaning, normalization was carried out to guarantee pattern recognition and forecasting model convergence. It is noteworthy that, thanks to the decision tree architecture, XGBoost predictors did not need data normalization before learning, and the same applies to statistical models based on the Box & Jenkins methodology. The normalization process used the z – score and was based on fuel consumption series values y_t taken every minute t , with its mean μ and standard deviation σ .

At the end of this stage, a set of 190,391 fuel consumption samples referring to 188 days was considered, totaling 54 complete operations. As machine learning-based nonlinear predictor techniques require sets of supervised signals during the model's learning period, the collected observations were divided into three distinct sets of operations. The first and most extensive set, called the training set, consisted of 36 operations used for forecast model refinement. The second set, called the validation set, consisted of 8 unknown operations during the adjustment of the models to monitor possible overfitting. This set was also used to select nonlinear model hyper-parameters. The last set, called the testing set, consisted of 10 operations in which linear and nonlinear prediction models were compared. It is worth mentioning that the statistical models do not require any validation step during model adjustment, since this procedure is based on the first 44 operations.

Figure 3 shows the fuel consumption dataset split. It is possible to see that the operations carried out during this period varied considerably depending on the requested order of dispatch, ranging from very-short operations (277 min) to operations over days (19,803 min).

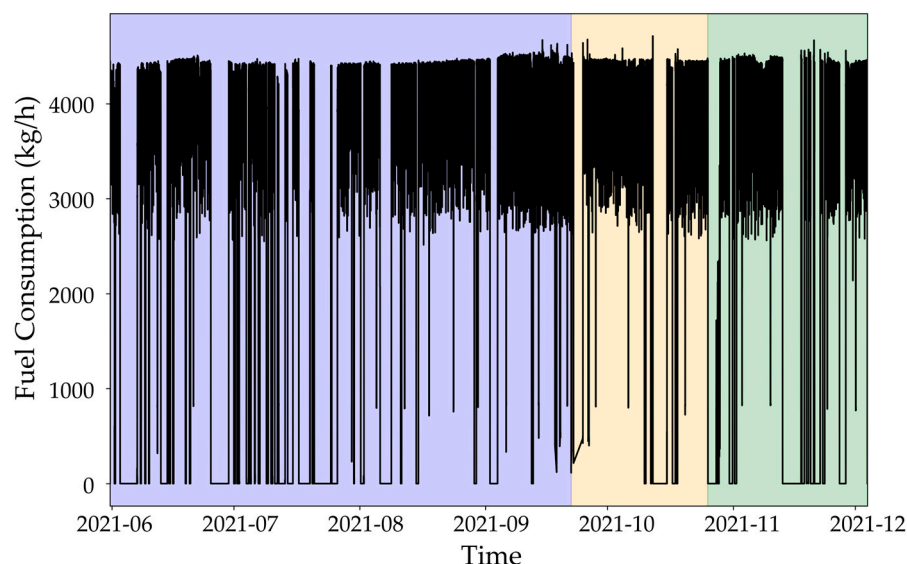


Figure 3. Data set division between training set (purple) with 36 operations ($\approx 60\%$ of data), validation set (orange) with 8 operations ($\approx 20\%$ of data), and testing set (green) with 10 operations ($\approx 20\%$ of data).

3.2. Linear Modeling

For AR and ARIMA linear modeling, it was first necessary to perform a fuel consumption series stationarity analysis. For this task, autocorrelation function (ACF) and partial autocorrelation function (PACF) were used to verify correlations between lags and detect possible patterns as an initial analysis. ADF and KPSS unit root and stationarity statistical

tests were also used [52,53]. These tests were used simultaneously although the ADF test is commonly used individually to verify series stationarity related to a deterministic trend potentially seen at the beginning and end of each operation. For nonstationarity patterns suggested by statistical tests at a 5% significance level, a new series was created by taking the analyzed time series difference d . The input of this subsystem was the preprocessed fuel consumption time series Z_i , and the output was a stationary time series and its statistical results. Statistical tests available by the *statsmodels* library in Python were used for this analysis.

After the stationarity analysis, 15 AR-type models were adjusted to identify the best model order p considering the analyzed error metrics. Model coefficients $\phi_1, \phi_2, \dots, \phi_p$ were estimated on the learning set, and Yule-Walker equations shown in Equation (2) were used. The input of this sub-step was the obtained stationary series, while the output 15 fitted AR models and their predictions on learning subset partition \hat{L}_{AR} .

To identify the influence of adding moving average components in the linear model approach, a univariate ARMA (p,q) model was adjusted by using two PSO modules as previously shown [29,54]. These modules are responsible for identifying the order (p,q) of the model through a PSO search (module 1) and estimating their respective coefficients (ϕ, θ) through another PSO search (module 2), minimizing the total prediction error on the learning subset. The first module was a discrete search mentioned in Equation (10) since the orders (p,q) are discrete values in the search space. For the second module, the search for coefficients was done in a continuous space through Equation (8). Table 1 shows the considered ARMA and PSO search parameter configuration ranges. Each search module was done after reaching the total number of iterations or if there was no prediction error improvement for several consecutive iterations. The input of this subsystem was the stationary series. The outputs were the identification of optimal order and coefficients (p,q) and (ϕ, θ) , respectively, swarms learning curve, an adjusted ARMA model and its predictions in the learning subset \hat{L}_{ARMA} .

Table 1. Parameter configuration ranges and options for PSO search and ARMA model.

Model	Parameter	Options
ARMA	(p,q)	1~15
	(ϕ, θ)	-1~1
PSO	χ	4.1
	$n_{particle}$	20
	$n_{iteration}$	250
	Δx_{max}	2

3.3. Nonlinear Modeling

For the nonlinear approach, it was first necessary to transform the fuel consumption dataset into a tabular input, as the nonlinear machine learning models required. Once transformed, NAR and XGBoost models were used to perform nonlinear predictions. To optimize the model performance, a hyper-parameter adjustment step was carried out, including determining the k input delay to make predictions. This step, seeking the best model performance considering the RMSE metric, applied a randomized search with 50 hyper-parameter combinations for each model. Each combination used different values of hyper-parameters within a pre-established limit (Table 2).

Models were evaluated through the cross-validation methodology according to the division of data presented in Section 3.1. Furthermore, to avoid problems related to the stochasticity of machine learning methods, each combination of hyper-parameters was repeated 10 times, totalizing 500 adjusted models during this analysis.

To define the best-randomized search combinations, we analyzed each metric set's average and standard deviation. NAR and XGBoost models with the best performance in the randomized search were selected and retrained 30 times to select the model with the smallest validation subset error metrics. The input of nonlinear modeling was the

preprocessed fuel consumption time series Z_i , while the output two adjusted models (NAR and XGBoost) and their respective predictions in the learning set \hat{N}_{NAR} and $\hat{N}_{XGBoost}$, respectively. Figure 4 describes the univariate modeling step scheme.

Table 2. Parameter configuration ranges and options for linear and machine learning modeling.

Model	Parameter	Option
AR	p	Yule-Walker Equations
ARMA	p, q	PSO
MLP	Input Delay	1, 3, 5, 7, 9, 12
	Learning Algorithm	Adam
	β_1	0.850~0.950
	β_2	0.900~0.999
	Learning Rate	$1 \times 10^{-10} \sim 1 \times 10^{-6}$
	Weight Decay L2	$0 \sim 1 \times 10^{-4}$
	Number of Hidden Layers	1, 3, 5, 7, 9
	Number of Neurons per Layer	50~1000
	Activation Function	ReLU, tanh, Sigmoid
	Batch Size	32, 64, 128, 254, 512, 1024
XGBoost	Input Delay	1, 3, 5, 7, 9, 12
	Learning Rate	0.01~0.51
	Maximum Depth	3~15
	Number of Estimators	1000~10,000
	Columns Sample by Tree	0.1~1
	Subsample	0.5~1
	λ	1~4.5
	α	0~1
	Maximum Delta Step	1~10
	γ	0.5~5
Minimum Child Weight	1~10	

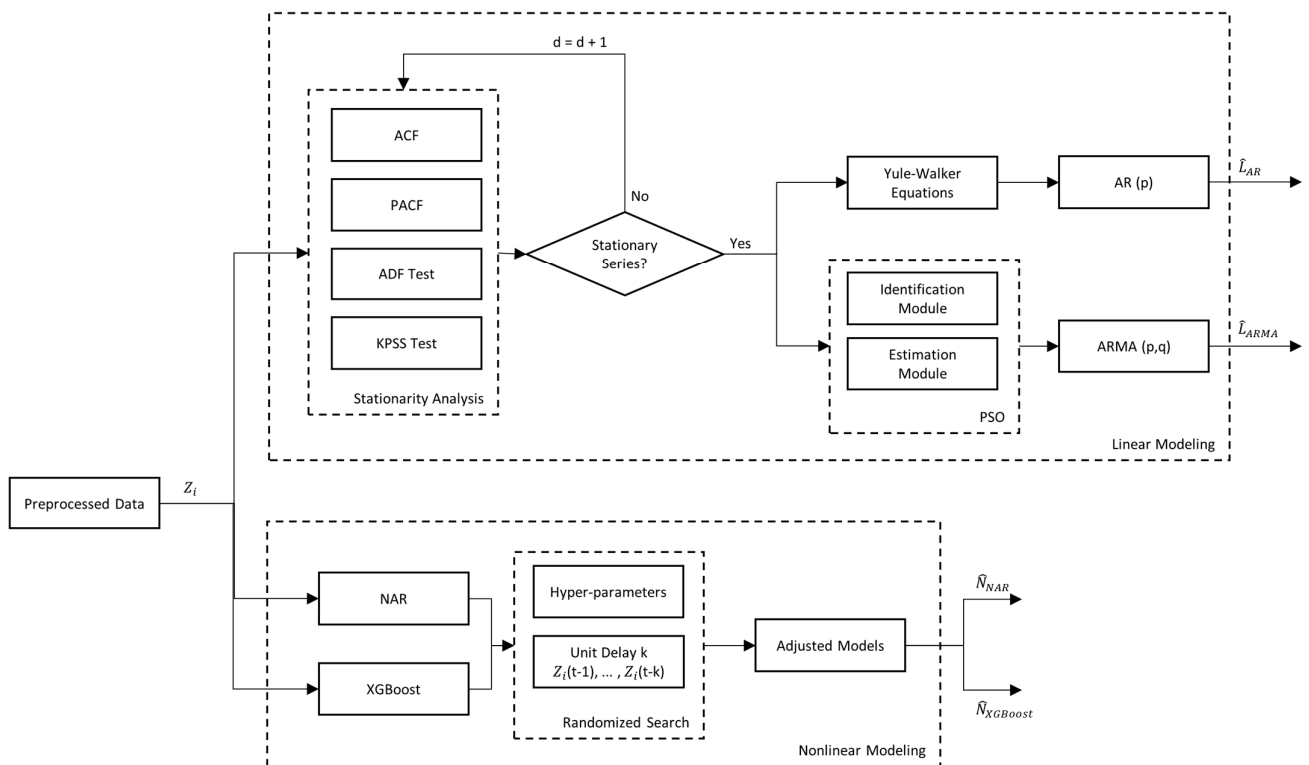


Figure 4. Univariate modeling approach in which linear and nonlinear modeling are optimized based on pre-processed data during training set generating \hat{L}_{AR} , \hat{L}_{ARMA} , \hat{N}_{NAR} , and $\hat{N}_{XGBoost}$.

3.4. Model Evaluation Metrics

Regarding error metrics, we used the root mean squared error (RMSE) (Equation (22)) and mean absolute error (MAE) (Equation (23)) to evaluate the performances:

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (z_i(t) - \hat{z}_i(t))^2} \quad (22)$$

$$MAE = \frac{1}{n} \sum_{t=1}^n |z_i(t) - \hat{z}_i(t)| \quad (23)$$

where n represents the total number of predicted values, $\hat{z}_i(t)$ are the output values of the prediction model and $z_i(t)$ are the actual values of the evaluated time series.

Both metrics are widely used in forecasting tasks, the former being more sensitive to outliers, while the latter providing the distances between predicted and actual values.

The traditional coefficient of determination (R^2) defined in Equation (24) was also used to indicate the prediction adherence compared to an adjusted regression. This metric varies between 0 and 1, and higher coefficients indicate better overall model performance.

$$R^2 = 1 - \frac{\sum_{t=1}^n (z_i(t) - \hat{z}_i(t))^2}{\sum_{t=1}^n (z_i(t) - \mu_i)^2} \quad (24)$$

The last metric utilized in this work, which was presented by Papandreou and Ziakopoulos [27], is a simple one used for regression and classification evaluation called custom accuracy (CA). This performance metric, defined in Equation (25), is calculated considering an acceptable error margin to distinguish whether predictions are accurate. For this work, we also considered a 5% custom threshold acceptable for performance evaluation.

$$N_{\hat{z}_i, 5\%} = \begin{cases} \text{accurate, if } \frac{|z_i - \hat{z}_i|}{z_i} \\ \text{not accurate, otherwise} \end{cases} \quad (25)$$

In summary, this metric counted how many values were considered accurate ($N_{\hat{z}_i, 5\%}$) within the custom threshold. It also considered the total number of predictions generated (N_{total}) for this analysis as indicated in Equation (26):

$$CA_{5\%} = \frac{N_{\hat{z}_i, 5\%}[\text{accurate}]}{N_{total}} \quad (26)$$

Even though its value varies between 0 and 1, like that of R^2 , it has a different performance evaluation principle.

For the final model evaluation analysis, we used the Wilcoxon Signed-Rank statistical test to evaluate statistical differences between linear and nonlinear models.

4. Results

This section will analyze and discuss adjustment procedure results of selected models, their prediction behavior in the test set, and their main error metrics.

4.1. Stationarity Analysis and AR and ARIMA Models Adjustment

Initially, original fuel consumption ACF and PACF series were analyzed as shown in Figure 5 to understand patterns correlation.

It is possible to see that more distant lags have a smaller magnitude than closer values, which indicates the stationarity nature of the time series. Since there is a smooth decay profile in the ACF as delays are considered, it is suggested that the series is stationary with p autoregressive orders indicated by the truncation after significantly relevant partial autocorrelations values presented in the PACF graph. The low influence of the moving averages for linear model construction considering the original series can also be observed

since there is no drastic drop in ACF after a specific delay q . Furthermore, evaluating the original PACF series, it is possible to visualize the presence mainly of the first three orders p with a purely autoregressive behavior $AR(p)$. However, there may be a significant influence of higher orders for model building as there is an increase in PACFs values as long delays are observed.

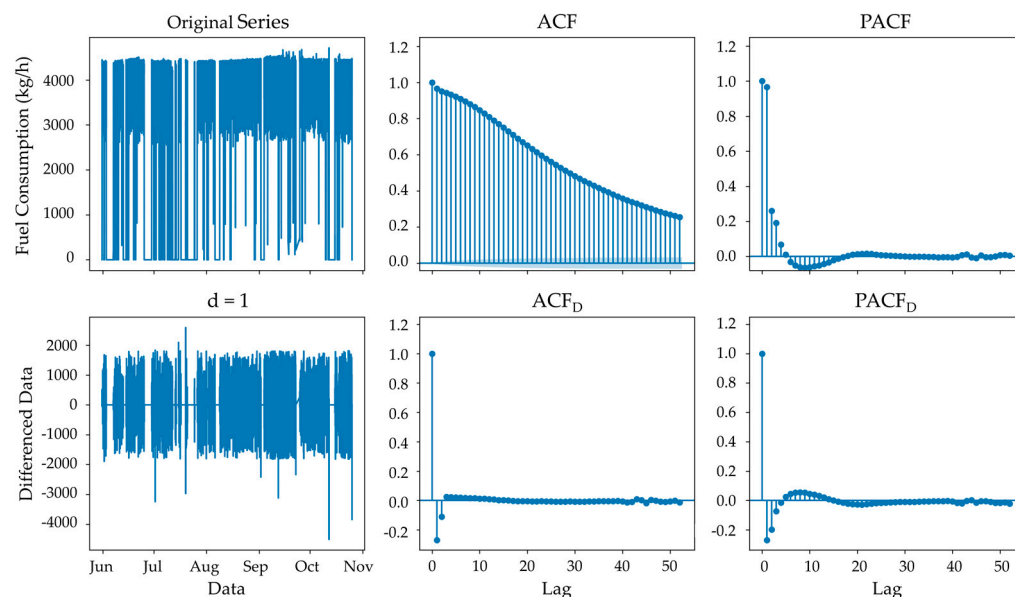


Figure 5. ACF and PACF functions behavior for the original fuel consumption series and its first difference ($d = 1$).

ADF and KPSS tests were performed to confirm the fuel consumption series stationarity hypothesis. The ADF test did not detect a unit root presence in model construction, suggesting the series stationarity during the learning set and considering a confidence level of 95% ($p - value_{ADF} \ll 0.01$). However, the KPSS stationarity test concerning a deterministic trend suggested the series non-stationarity around the mean μ considering the reliability of 95% ($p - value_{KPSS} = 0.01$), possibly due to fuel consumption behavior changes during engine start-up and shutdown, as well as sudden drops in fuel consumption during regular operation. It is noteworthy that while the null hypothesis of the ADF indicates the series non-stationarity, the null hypothesis used in the KPSS test suggests a possible stationarity of the time series around the mean. Due to contradictory conclusions, we decided to carry out a differentiation ($d = 1$) and repeat the stationarity analysis in the differentiated series also presented in Figure 5.

Unlike the original series, ACF and PACF autocorrelations for the differentiated series suggested a presence of autoregressive and moving average behaviors since there was a drastic drop in both after a set of orders p and q . ADF and KPSS statistical tests under the differentiated series indicated a stationarity nature ($p - value_{ADF} \ll 0.01$ and $p - value_{KPSS} = 0.10$) also considering the reliability of 95%, making it clear that any deterministic trends were removed along the differentiated series mean. ACF truncation referring to the orders $q = 1$ or $q = 2$ together with a sharp decay in PACF can also suggest a purely moving average characteristic for the model final adjustment, which will consider the search for identification and estimation parameters through the previously presented PSO modules. Two observations can then be made on the analysis of both functions: visually identifying orders can be a complex task; it is possible to perform the adjustment of more than one model that guarantees white noise presence, which is an indication of linear patterns generalization when using this type of approach.

Error metrics referring to the adjustments of the 15 different $AR(p)$ models during the learning set considering the first 15 orders p estimated through Yule-Walker equations are shown in Figure 6.

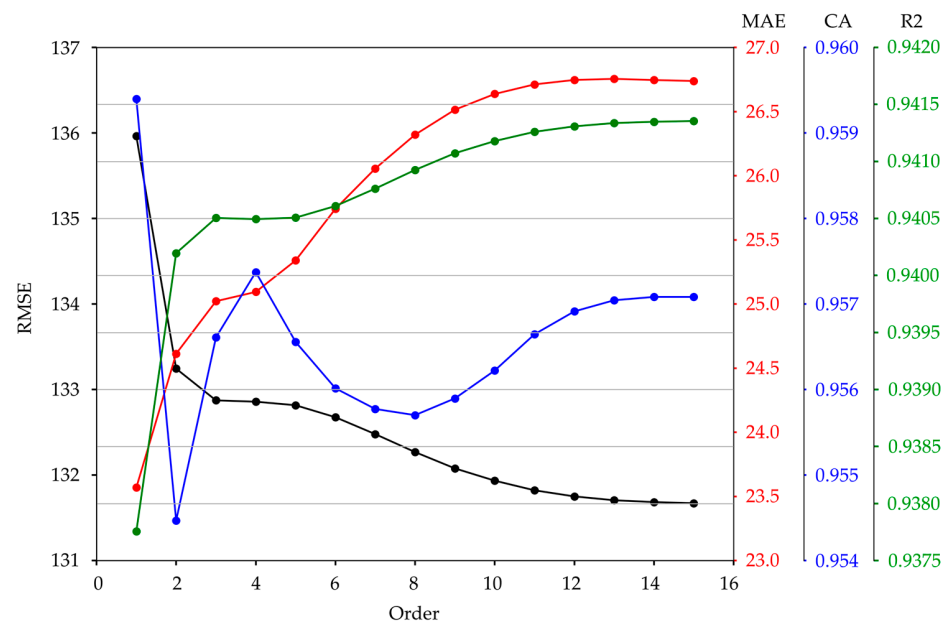


Figure 6. RMSE, MAE, R^2 , and CA error metrics for each one of the 15 AR models adjusted by Yule-Walker equations.

These model adjustments were also evaluated in the differentiated series that obtained relatively worse metrics, indicating a better predictive capacity of this family of models for the original fuel consumption series. Considering the set of used error metrics, a $p = 3$ order was chosen for the final AR model since this order generates a balance between performance and complexity, being considered the most parsimonious model among those generated. It is noteworthy that the model can better predict perturbations in the time series, directly reducing the RMSE metric with higher orders. However, it becomes more sensitive to small changes that affect MAE evaluation. The increase does not influence R^2 and CA behaviors in orders.

Regarding ARIMA model family application (p,d,q) , orders identification (p,q) and coefficients estimation (ϕ,θ) from the PSO modules found an ARIMA model $(4,1,6)$ with convergence curves shown in Figure 7. It is possible to observe that the order selection module (1st module) achieved a significantly faster convergence than the coefficient estimation (2nd module). This process lasted about 50 iterations without any decrease in the objective function. However, we decided to present fewer iterations to facilitate learning curve visualization. This behavior is predictable since the orders discrete search space is limited to few possibilities (orders 1~15 in both cases), while coefficients' continuous search space makes optimization a more complex procedure. Additionally, despite only six iterations required to model the adjustment process in the first module to find the optimal solution, there was a computational cost of 13.73 h to carry out this task, which is considerably longer than the AR models estimation, which had a processing period of a few minutes to calculate the 15 combinations. It is worth mentioning that depending on the orders (p,q) , the estimation of coefficients (ϕ,θ) varied with duration ranging from 30 min to 2 h for orders higher than both cases.

Despite providing a more complex model than the AR model with more coefficients to be identified and estimated, the addition of moving averages did not show a significant performance difference in error metrics in the learning set (RMSE = 132.72; MAE = 28.30; $R^2 = 0.940$; CA = 0.959). However, since engine-driven operations had severe fluctuations in the fuel consumption series that do not present linear patterns, the addition of moving average orders may have caused extra errors by considering q previous errors to make prediction corrections in samples right after sudden changes. Despite error metrics deterioration due to this ARIMA model corrective behavior, this sensitivity helped predict other patterns that improved the metrics set in the training set.

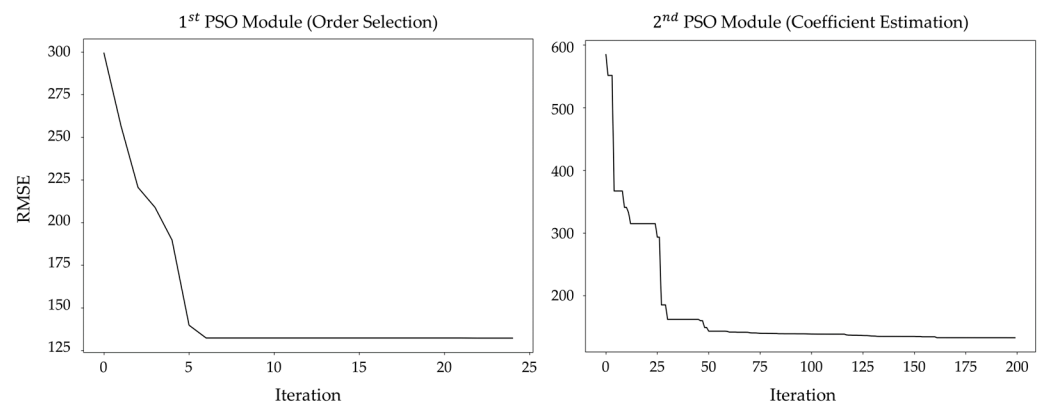


Figure 7. ARIMA model optimization search for orders (p,q) and coefficients (ϕ,θ) through PSO modules over iterations.

4.2. NAR and XGBoost Models Adjustment

For nonlinear univariate prediction NAR and XGBoost models, the RMSE set obtained in the randomized search for the validation set depends considerably on the combination of hyper-parameters used (Figure 8). The five best adjustments were highlighted in color to facilitate the performance visualization of each combination. Combinations 10 (XGBoost) and 36 (NAR), highlighted in green, were selected for model final fit because they had the lowest RMSE means as the lowest limits.

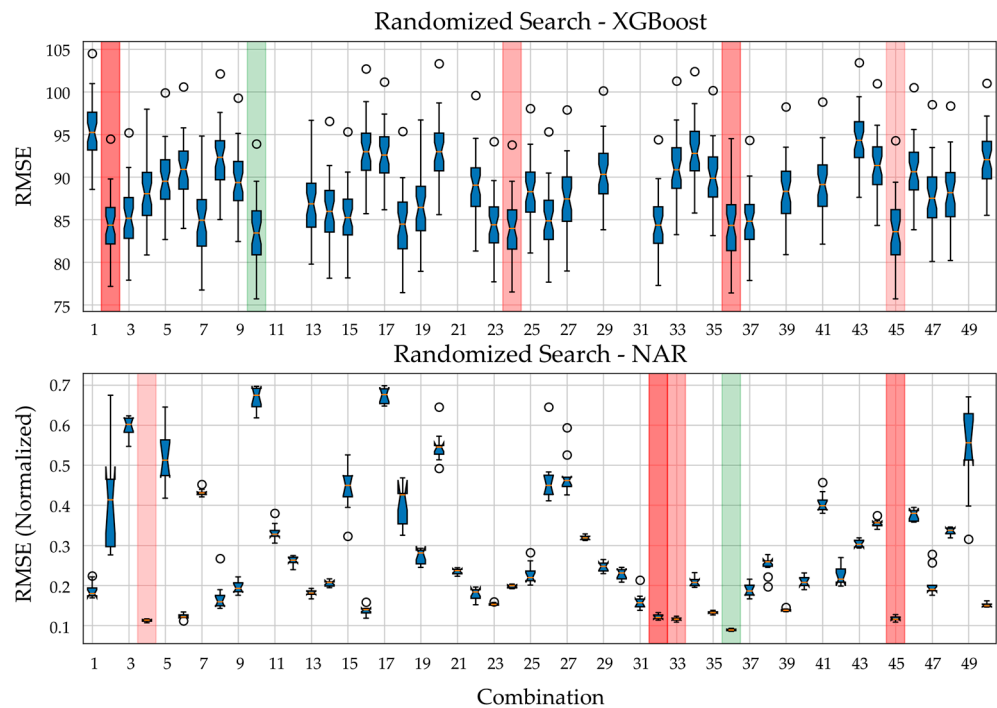


Figure 8. Error metrics for each NAR and XGBoost generated model with different hyper-parameters selected through the random search technique.

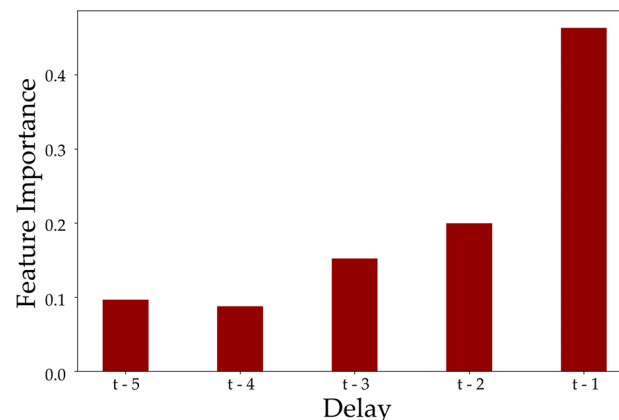
Table 3 lists the selected hyper-parameters for each analyzed nonlinear model.

It is worth mentioning that combinations 11, 12, 21, 28, 30, 31, 38, 40, 42, and 49 of the univariate XGBoost model randomized search exhibited much higher error metrics than the others, but they are not shown to facilitate the visualization of the other combinations.

Table 3. Selected NAR and XGBoost hyper-parameters set found through randomized search with the lowest RMSE metric in the validation set.

Model	Parameter	Option
NAR	Input Delay	9
	β_1	0.923
	β_2	0.928
	Learning Rate	6.742×10^{-7}
	Weight Decay L2	8.144×10^{-6}
	Number of Hidden Layers	9
	Number of Neurons per Layer	156
	Activation Function	Sigmoid
	Batch Size	32
XGBoost	Input Delay	5
	Learning Rate	0.298
	Maximum Depth	7
	Number of Estimators	2025
	Columns Sample by Tree	0.645
	Subsample	0.815
	λ	4.09
	α	0.544
	Maximum Delta Step	9.17
	γ	3.25
Minimum Child Weight	8	

XGBoost error metrics have the same scale as the fuel consumption series since the input data normalization process is unnecessary due to the models' structure construction based on residuals. Additionally, for the analyzed validation set, XGBoost error metrics were much lower than the previously mentioned AR and ARIMA model metrics, suggesting that there was better model adherence during training. The relevance of each temporal step after the final model adjustment is shown in Figure 9, which indicates each attribute proportion usage for DT building. It is possible to notice that the fuel consumption immediately preceding minute $t - 1$ has an important composing role in the next-minute prediction. We can also observe decreasing lag importance on the XGBoost model trees with further delays. This behavior is close to the previously presented ACF profile behavior. Furthermore, $t - 5$ and $t - 4$ steps showed similar importance, suggesting that new model entries had no benefits. The 50 randomized search combinations execution time was equal to 26.62 h, being much higher than the AR model adjustment but only two times longer than ARIMA-PSO adjustment through order identification and coefficients estimation modules.

**Figure 9.** Temporal steps importance taken into account for the residual tree's construction in the selected XGBoost model during the randomized search step.

NAR randomized search RMSE metrics refer to normalized fuel consumption prediction values since this practice is necessary for good model convergence. All neural network simulations considered 500 epochs supported by an early stopping technique. This procedure interrupts weights adjustment after not detecting RMSE improvement in the validation set after 150 consecutive iterations to reduce the analyzed combinations' computational effort. NAR modeling lasted 231.51 h, resulting in a computational cost 8.7 times greater than XGBoost modeling and the slowest univariate model optimized through the followed methodology.

The final NAR model convergence profile on training and validation sets is depicted in Figure 10. Initially, it is possible to observe that both learning curves presented very noisy convergences that tended to decrease their variations during the learning phase. Noisy patterns' presence in small randomly selected training batches justifies this behavior and contributes to better model generalizability. Adam batches stochasticity forces neural network destabilization in local optima during training epochs. The decrease in curve perturbation at the end of epochs may have been related to learning rate smoothing due to Adam β coefficients. Despite this behavior, training and validation sets converged without overfitting characteristics during learning.

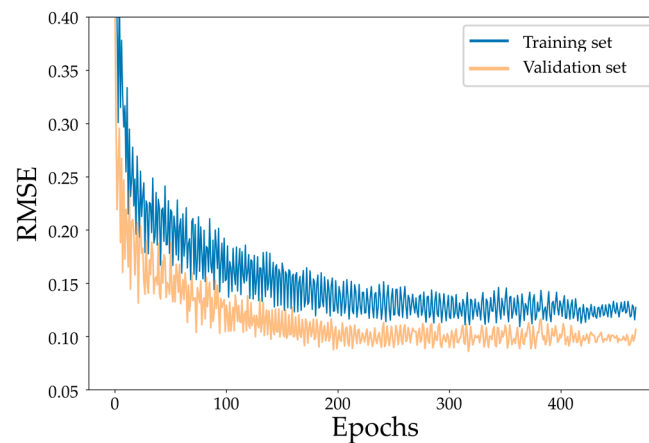


Figure 10. Best NAR model convergence over epochs during training and validation set.

Notably, the magnitude of RMSE calculated on the validation set (orange) during the learning period is smaller than the one of the training error itself (blue). There are several assumptions related to this type of behavior, the main ones being:

- The eight validation set operations have simpler patterns to be captured by the neural network. However, validation predictions were not affected by ANN's higher generalization capability due to more complex patterns during training;
- Among 36 training operations, there may be random noises that are not related to the engine's operation and were not observed during the model validation;
- The L2 regularization process is applied only to the training set and increases the cost function value on this occasion to avoid overfitting. The cost function applied to the validation set is calculated from an unregularized RMSE, which may have resulted in lower error magnitudes;
- The dropout regularization technique seeks to penalize the model variance by randomly freezing neurons in a specific layer during the training set and unfreezing them during validation. However, this technique was not applied in the present study.

After the hyper-parameters selection phase, fuel consumption testing set predictions were made to better understand each model's characteristics during the final 10 operations. NAR and XGBoost models were retrained 30 times to avoid misadjustments due to the algorithms' stochasticity.

4.3. Univariate Prediction Analysis by Engine Operation

Initially, it is useful to consider that the duration of operations in the test set varies drastically, ranging from 277 to 16,333 min, which directly reflects on the performance of the forecast models and fuel consumption predictions profile. Error metrics obtained during this analysis are illustrated in Figure 11. The linear models' (AR and ARIMA-PSO) performance in the test set had similar error metrics behavior. Although the AR model exhibited a slightly higher RMSE in operations 5, 6, 7, and 8, it had a lower MAE in 9 of the 10 operations. However, NAR and XGBoost models' performance was better for all test operations except for operation 4. There is a general behavior similarity among the machine learning models. However, the NAR model provided closer predictions to the actual fuel consumption series, which directly implied a decrease in MAE among almost all models. The short-duration characteristics of operations 4 and 7 are directly reflected in the peaks of RMSE and MAE metrics and the reductions of R^2 and CA metrics.

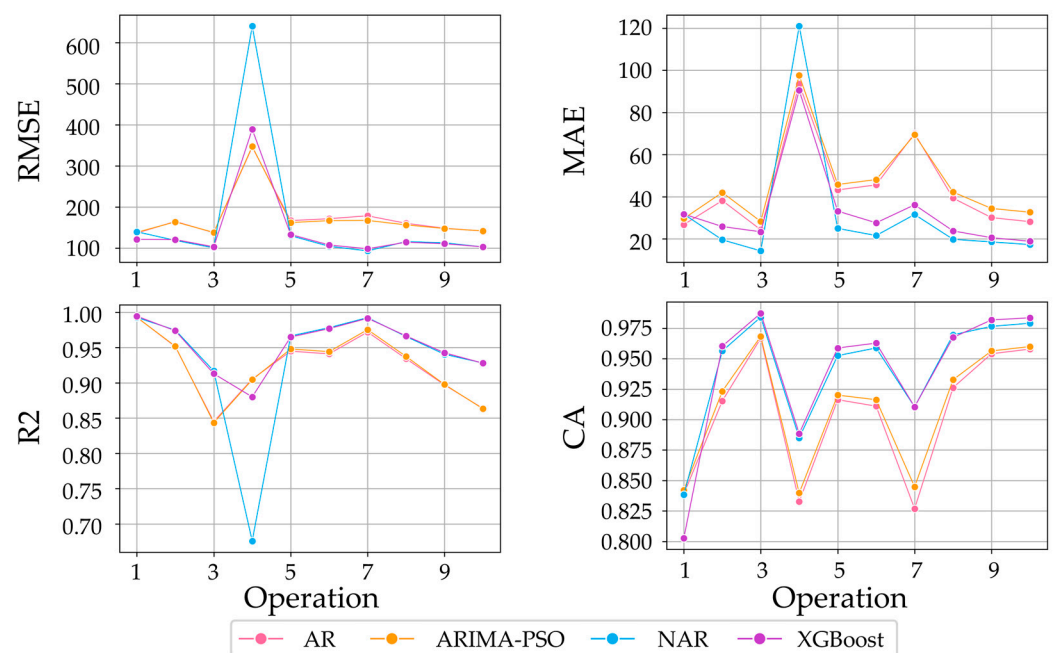


Figure 11. RMSE, MAE, CA, and R^2 of univariate models in all 10 test set operations. Performances of AR (pink) and ARIMA-PSO (orange) linear models are directly compared to those of NAR (blue) and XGBoost (purple) nonlinear models.

Despite relatively close error metrics, operation prediction profiles differ from each other. It was possible to observe that AR and ARIMA-PSO models were better at capturing full-load operation patterns when there were no large fuel consumption fluctuations. However, as shown in Figure 12, NAR and XGBoost models were better at predicting sudden variations due to process stochasticity. Initially, it was possible to notice that no model captured sudden drop patterns when prediction vs. actual fuel consumption curves were analyzed. These patterns are related to fuel valve closure that guarantees micro fuel leak usage. Leaked fuel is stored in an external reservoir positioned right after the flowmeter. Due to this positioning, reused fuel flow is not captured by the sensors for 1 to 3 min until all total leaked fuel is consumed. This sharp drop pattern is unpredictable by univariate models as they only consider past fuel consumption observations to predict its future.

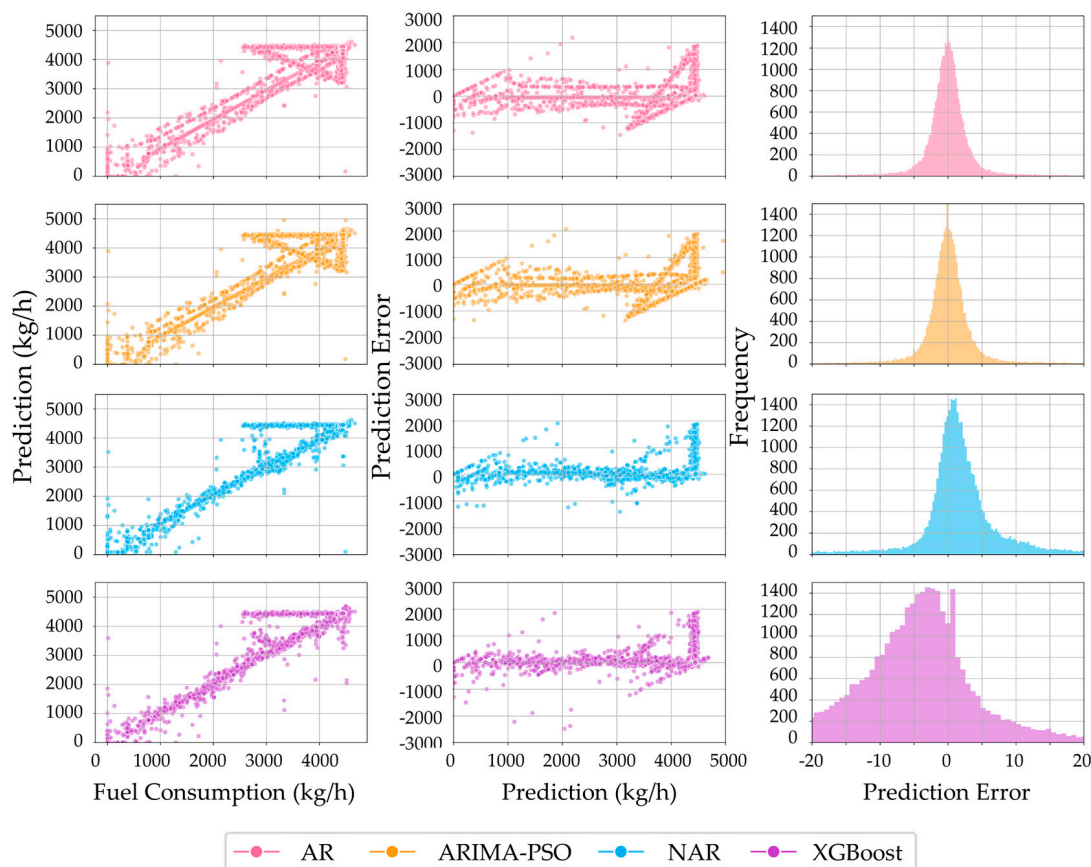


Figure 12. Univariate models prediction vs. fuel consumption (**left**), prediction error vs. prediction (**middle**), and frequency vs. prediction error (**right**) in all 10 test set operations. Performance of AR (pink) and ARIMA-PSO (orange) linear models are directly compared to those of NAR (blue) and XGBoost (purple) nonlinear models.

Each model's sensitivity to readjusting after sudden drop episodes is another exciting pattern to be analyzed. As AR and ARIMA-PSO models make predictions based on a linear combination of the past values, the immediately subsequent predictions are affected by abrupt oscillations. This characteristic generates error propagation for a few minutes, which increases error prediction, mainly in longer and more complex operations. This error propagation can be observed in Figure 12 through the arrow in the prediction vs. actual consumption graphs and prediction error vs. prediction graphs. Furthermore, these characteristics generate punctual errors up to 2000 kg/h. However, when frequency and prediction errors magnitude are analyzed during regular operation in which fuel consumption varies only between 4350 to 4650 kg/h, it is possible to observe that the AR and ARIMA-PSO linear models generate predictions with errors that have normal distributions close to zero (-0.03 ± 3.55 and 0.03 ± 3.78 kg/h, respectively).

Different patterns were observed when NAR and XGBoost models were applied. Although these models were not able to capture the sudden drops related to fuel supply valve closure, they were able to accurately predict most of the time fuel consumption in the minutes immediately after these events, reducing error propagation. Error decrease after sudden drops is evidenced in Figure 12 by the disappearance of the previously mentioned arrow, mainly in the predictions made by the NAR model. Furthermore, it is possible to verify that the NAR model generates relatively worse predictions than the AR and ARIMA-PSO linear models during regular engine operation at full load. Additionally, NAR predictions tended to overestimate the fuel consumption series with a displaced normal distribution of 1.66 ± 5.20 kg/h, while XGBoost conventional operation predictions resulted in a fuel consumption underestimation with a normal distribution of -4.03 ± 7.42 kg/h.

This characteristic reinforces the assumption that the model has a higher sensitivity to slight variations despite the use of optimized regulators determined through the randomized search.

Operations and model prediction characteristics can be seen in Figure 13. These behaviors are related to different operation phases such as engine start-up (ramp) and shutdown, sudden peaks and drops, and nonlinear oscillations inherent to engine operation. While AR and ARIMA-PSO models underestimated fuel consumption during the start-up process and overestimated future values at engine shutdown, NAR and XGBoost models provided predictions closer to the actual series values. It is noteworthy that among the linear models, ARIMA-PSO made closer predictions when compared to the real fuel consumption series, benefiting from the use of moving averages for this operation stage. The higher XGBoost sensitivity was confirmed when small fluctuations happened at full load, which resulted in an MAE penalty. This sensitivity may have occurred due to the learning ability of the model to predict sudden variations that affected its capacity to predict simpler patterns inherent to the engine's operation. Linear model behavior in predicting sudden drops confirmed prediction error propagation and greater ARIMA-PSO instability due to moving averages addition. Nonlinear models correctly predicted fuel valve opening and achieved faster prediction stability in these scenarios.

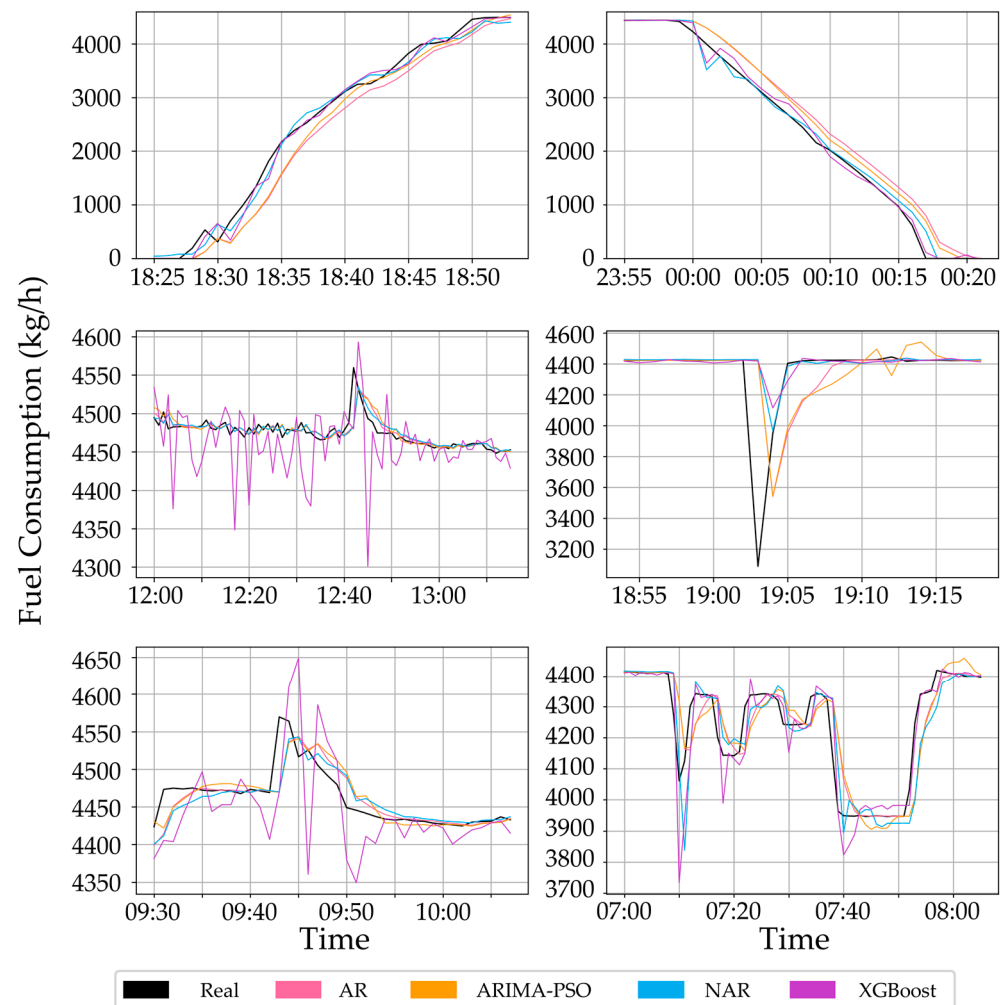


Figure 13. Univariate model prediction characteristics during start-up, shutdown, full-load, and sudden drop episodes in the test set operations. Real fuel consumption series (black) is directly compared to AR (pink) and ARIMA-PSO (orange) linear modeling, and NAR (blue) and XGBoost (purple) nonlinear modeling.

We performed statistical analysis to evaluate possible differences between the applied models on test metrics. For this purpose, we used the Wilcoxon Signed-Rank Test to verify whether there was any statistically significant difference between error median values considering a confidence level of 95%. It was not possible to reject the null hypothesis since there was no difference in central tendency between AR and ARIMA-PSO models ($p - value_{RMSE} = 0.232$ and $p - value_{R^2} = 0.193$) and NAR and XGBoost models ($p - value_{RMSE} = 0.770$, $p - value_{MAE} = 0.105$, $p - value_{R^2} = 1.00$ and $p - value_{CA} = 0.139$). However, we observed a statistically significant difference between AR and ARIMA-PSO models regarding MAE and CA error metrics ($p - value_{MAE} = 0.004$ and $p - value_{CA} = 0.002$). This difference can be explained by the addition of moving averages to the ARIMA-PSO model, which reduced its stability to sudden oscillations during operations, directly affecting MAE and CA metrics.

5. Conclusions

Univariate linear AR models adjusted by the Yule-Walker equations and ARIMA identified and estimated by the PSO module's ability to capture patterns in engine fuel consumption were investigated. The presented results indicated that, despite the computational time spent increase related to the PSO modules application to adjust ARIMA models of 13.73 h, there were no improvements in the training set when compared to the AR(3) model. There was a significant MAE degradation of 13.11% due to the moving averages addition to the ARIMA(4,1,6) model, which caused prediction instabilities right after sudden fluctuations during the operation. The test set metrics behavior and error profile were statistically similar to the analytically adjusted AR model application, even with the same performance behavior observed during training with a 1.24% RMSE improvement and 6.45% MAE degradation. This characteristic suggests that the diesel/HFO engine fuel consumption series may be purely autoregressive. Overall AR and ARIMA-PSO performance during full-load operation presented normal error distributions of -0.03 ± 3.55 and 0.03 ± 3.78 kg/h, respectively. Additionally, although these models did not efficiently capture consumption variations related to disturbances and periods of power output modification (being the ARIMA-PSO model is more accurate for the latter task), they were superior to machine learning models to assimilate linear patterns during regular operation.

Simultaneously, the performance of nonlinear NAR and XGBoost models adjusted through randomized searches was investigated. As expected, the machine learning model computational costs were considerably higher than the linear prediction model adjustments. During the learning stage, while training the 50 randomized XGBoost models took 26.62 h (approximately twice as long as the ARIMA-PSO fitting), the NAR model training took 231.51 h (8.7 times longer than the XGBoost training). XGBoost feature importance for each temporal step considered during the DTs construction was evaluated. It was found that there was a recent input predominance of 81.5% ($t - 1$, $t - 2$, and $t - 3$) compared to the later temporal steps ($t - 4$ and $t - 5$). These models ensured considerably better error metrics in test operations when compared to the applied linear models. The NAR and XGBoost models had statistically similar RMSE performances with an improvement of 26.59% on average when compared with the ARIMA-PSO model, which was the best linear model considering this error metric. Furthermore, the NAR model had a 42.37% MAE improvement over the AR model, while the XGBoost model achieved a 30.30% improvement over the same statistical model. Predictions based on ANN showed RMSE, R^2 , and CA metrics similar to the XGBoost technique. However, there was a clear difference in MAE behavior between these models, with the ANN-based model having a MAE 21.96% lower than the boosting model. It is worth mentioning that these results did not consider the operation 4 performance due to the error metrics' high variability related to the very-short duration of this operation. Additionally, it was possible to notice that although NAR and XGBoost models can assimilate nonlinear patterns related to engine startup, shutdown, and sudden fluctuations in fuel consumption, the application of the XGBoost predictor resulted in a higher prediction sensitivity during the full load operation, despite regularizers usage

during model construction, which reinforces the MAE metric reduction in relation to the NAR model. Overall NAR and XGBoost performance during full-load operation presented normal error distributions of 1.66 ± 5.20 and -4.03 ± 7.42 kg/h, respectively. Therefore, it was possible to observe better adherence to NAR and ARIMA-PSO univariate models for engines' fuel consumption series.

Finally, predictions for the very short-term range fuel consumption series related to a large TPP diesel/HFO engine can be accomplished with univariate forecasting models. The machine learning methods application can improve real-time data-driven system monitoring despite the high computational cost inherent to model training. However, statistical methods proposed by Box & Jenkins also present satisfactory performances for the analyzed fuel consumption series. The methodology applied in this work can be adapted for other parameters inherent to the engine operation analysis, such as SO_x and NO_x emission episodes at TPPs. Since these series typically have emission peak patterns that are not effectively predicted with linear models, NAR and XGBoost can be investigated for pollution episode prediction due to their capacity to capture nonlinear patterns and sudden oscillations. However, each emission series behavior must be analyzed, and hybrid machine learning models can be developed to obtain more accurate predictions. Overall, nonlinear forecasting techniques have already been applied to improve emissions rates in coal-fired TPPs over long periods and can also be explored in diesel/HFO engine emission series. Additionally, it is necessary to verify whether exogenous variables addition related to the engine's operating state and combustion quality can benefit emission levels pattern recognition as a multivariate forecasting approach.

Therefore, it is possible to notice that no analyzed univariate forecasting model could better generalize all the described fuel consumption series patterns. Given the model's high-performance specificity in capturing different operation patterns, such as full-load, start-up, shut-down, and sudden oscillations, we propose the development of new adaptive prediction algorithms that can identify momentary operation regions and, based on this information, select the historically best-suited forecasting model for the analyzed behavior. Machine learning techniques based on temporal window clustering and forecasting models classification based on training errors could be applied. Additionally, different time series forecasting techniques, such as recurrent neural networks (RNN) and support vector machine (SVM), can be analyzed. Beyond that, in future works, the multivariate time series forecasting techniques application can be analyzed for fuel consumption prediction. For that matter, the addition of a new feature selection step can be investigated through linear and nonlinear causality analysis. Additionally, it is worth mentioning the investigation of each model's performance through different fuel consumption series temporal granularities and input delays during the data pre-processing step.

Author Contributions: Conceptualization, E.A.S.-F., H.V.S. and C.J.A.B.-F.; methodology, E.A.S.-F., H.V.S. and C.J.A.B.-F.; software, E.A.S.-F.; validation, E.A.S.-F., H.V.S. and C.J.A.B.-F.; formal analysis, E.A.S.-F. and M.F.A.L.; investigation, E.A.S.-F. and M.F.A.L.; writing—original draft preparation, E.A.S.-F., M.F.A.L., A.C., H.V.S. and C.J.A.B.-F.; writing—review and editing, E.A.S.-F., M.F.A.L., A.C., H.V.S. and C.J.A.B.-F.; visualization, E.A.S.-F. and M.F.A.L.; supervision, A.C., H.V.S. and C.J.A.B.-F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by: the Programa de Pesquisa e Desenvolvimento da Agência Nacional de Energia Elétrica (ANEEL) and TPP Energética Suape II S.A. as the project "Desenvolvimento de dispositivo utilizando hidrogênio e oxigênio para a redução dos gases poluentes no ciclo de combustão carburante" grant number PD-06599-0007/2019, the Brazilian National Council for Scientific and Technological Development (CNPq) grant number 315298/2020-0, and Araucaria Foundation grant number 51497.

Data Availability Statement: Restrictions apply to the availability of these data. Data were obtained from TPP Energética Suape II S.A. and are unavailable due to confidentiality agreements.

Acknowledgments: The authors are grateful to the Advanced Institute of Technology and Innovation (IATI) team, especially to Robson do Carmelo Santos Barreiros and Flávio Wilson Barreiros de

Oliveira for their efforts in data acquisition. The authors are also grateful to TPP Energética Suape II S.A. and its team for supplying the underlying data, especially to Luzivan da Cruz Moura for technical enlightments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Energy Research Company. *Brazilian Energy Balance—Year 2021*; Technical Report; Ministry of Mines and Energy: Brasilia, Brazil, 2022.
2. National Electric Systems Operator. Plano da Operação Energética 2021/2025. Technical Report RT—ONS DPL 0492/2021. 2021. Available online: <https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/revisoes-quadrimestrais-da-carga#:~:text=Planejamento%20Anual%20da%20Opera%C3%A7%C3%A3o%20Energ%C3%A9tica%202021%2D2025&text=Para%20o%20per%C3%ADodo%202021%2D2025,pelo%20carregamento%20estat%C3%ADstico%20de%202020> (accessed on 1 July 2022).
3. Santos, T.N.; Diniz, A.L.; Sabóia, C.H.; Vilas Boas, C.E.; Ferreira, J.M.F.; Mourão, F.; Cabral, R.; Cerqueira, L.F.; Araújo Junior, C.A. Incorporação de restrições operativas detalhadas utilizadas na elaboração do Programa Diário de Operação do Operador Nacional do Sistema (ONS) no modelo de Despacho Hidrotérmico de Curto Prazo. In Proceedings of the XXV SNPTEE—Seminário Nacional de Produção e Transmissão de Energia Elétrica, Belo Horizonte, Brazil, 10–13 November 2019; pp. 1–10.
4. Santos, T.N.; Diniz, A.L.; Saboia, C.H.; Cabral, R.N.; Cerqueira, L.F. Hourly pricing and day-ahead dispatch setting in Brazil: The dessem model. *Electr. Power Syst. Res.* **2020**, *189*, 106709. [[CrossRef](#)]
5. KPMG. Energética Suape II S.A.: Demonstrações Financeiras em 31 de Dezembro de 2020. Tech. Rep. KPDS 758824, KPMG Auditores Independentes. 2021. Available online: https://suapeenergia.com.br/df/DF_2020.pdf (accessed on 1 July 2022).
6. Xu, J.; Gu, Y.; Chen, D.; Li, Q. Data mining based plant-level load dispatching strategy for the coal-fired power plant coal-saving: A case study. *Appl. Therm. Eng.* **2017**, *119*, 553–559. [[CrossRef](#)]
7. Rao, X.; Sheng, C.; Guo, Z.; Yuan, C. A review of online condition monitoring and maintenance strategy for cylinder liner-piston rings of diesel engines. *Mech. Syst. Signal Process.* **2022**, *165*, 108385. [[CrossRef](#)]
8. Hajdarevic, A.; Džananovic, I.; Banjanovic-Mehmedovic, L.; Mehmedovic, F. Anomaly Detection in Thermal Power Plant using Probabilistic Neural Network. In Proceedings of the 2015 38th IEEE International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), Opatija, Croatia, 25–29 May 2015; pp. 1118–1123.
9. Xu, R.; Yan, W. A Comparison of GANs-Based Approaches for Combustor System Fault Detection. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; IEEE: Glasgow, UK; pp. 1–8.
10. Siqueira, H.; Macedo, M.; Tadano, Y.d.S.; Alves, T.A.; Stevan, S.L., Jr.; Oliveira, D.S., Jr.; Marinho, M.H.; Neto, P.S.d.M.; Oliveira, J.F.d.; Luna, I.; et al. Selection of temporal lags for predicting riverflow series from hydroelectric plants using variable selection methods. *Energies* **2020**, *13*, 4236. [[CrossRef](#)]
11. Smrekar, J.; Pandit, D.; Fast, M.; Assadi, M.; De, S. Prediction of power output of a coal-fired power plant by artificial neural network. *Neural. Comput. Appl.* **2010**, *19*, 725–740. [[CrossRef](#)]
12. Zhang, W.; Yang, D.; Wang, H. Data-driven methods for predictive maintenance of industrial equipment: A survey. *IEEE Syst. J.* **2019**, *13*, 2213–2227. [[CrossRef](#)]
13. Tunckaya, Y.; Koklukaya, E. Comparative prediction analysis of 600 MWe coal-fired power plant production rate using statistical and neural-based models. *J. Energy Inst.* **2015**, *88*, 11–18. [[CrossRef](#)]
14. Lu, Y.S.; Lai, K.Y. Deep-learning-based power generation forecasting of thermal energy conversion. *Entropy* **2020**, *22*, 1161. [[CrossRef](#)] [[PubMed](#)]
15. Louzazni, M.; Mosalam, H.; Khouya, A.; Amechnoue, K. A non-linear auto-regressive exogenous method to forecast the photovoltaic power output. *Sustain. Energy Technol. Assess.* **2020**, *38*, 100670. [[CrossRef](#)]
16. Kerem, A.; Kirbaş, İ. Multi-step forward forecasting of electrical power generation in lignite-fired thermal power plant. *Mühendislik Bilim. Ve Tasarım Derg.* **2021**, *9*, 1–13. [[CrossRef](#)]
17. Neto, P.S.D.M.; Firmino, P.R.A.; Siqueira, H.; Tadano, Y.D.S.; Alves, T.A.; De Oliveira, J.F.L.; Marinho, M.H.D.N.; Madeiro, F. Neural-based ensembles for particulate matter forecasting. *IEEE Access* **2021**, *9*, 14470–14490. [[CrossRef](#)]
18. Siqueira, H.; Boccato, L.; Attux, R.; Filho, C.L. Echo state networks for seasonal streamflow series forecasting. In Proceedings of the Intelligent Data Engineering and Automated Learning (IDEAL), Natal, Brazil, 29–31 August 2012; Springer: Berlin/Heidelberg, Germany; pp. 226–236.
19. Box, G.E.P.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*, 5th ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2016; pp. 1–709.
20. Hyndman, R.J.; Khandakar, Y. Automatic time series forecasting: The forecast package for R. *J. Stat. Softw.* **2008**, *27*, 1–22. [[CrossRef](#)]
21. Wang, H.; Zhao, W. ARIMA Model Estimated by particle swarm optimization algorithm for consumer price index forecasting. In Proceedings of the International Conference on Artificial Intelligence and Computer Science, Shanghai, China, 7–8 November 2009; Springer: Berlin, Germany; pp. 48–58.
22. Ervural, B.C.; Beyca, O.F.; Zaim, S. Model estimation of ARMA using genetic algorithms: A case study of forecasting natural gas consumption. *Procedia Soc. Behav. Sci.* **2016**, *235*, 537–545. [[CrossRef](#)]

23. Siqueira, H.; Santana, C.; Macedo, M.; Figueiredo, E.; Gokhale, A.; Bastos-Filho, C. Simplified binary cat swarm optimization. *Integr. Comput. Aided Eng.* **2021**, *28*, 35–50. [\[CrossRef\]](#)
24. Bento, P.M.; Pombo, J.A.; Calado, M.R.; Mariano, S.J. Stacking ensemble methodology using deep learning and ARIMA models for short-term load forecasting. *Energies* **2021**, *14*, 7378. [\[CrossRef\]](#)
25. Zhang, G.; Patuwo, B.E.; Hu, M.Y. Forecasting with artificial neural networks: The state of the art. *Int. J. Forecast.* **1998**, *14*, 35–62. [\[CrossRef\]](#)
26. de Souza Tadano, Y.; Siqueira, H.V.; Alves, T.A. Unorganized machines to predict hospital admissions for respiratory diseases. In Proceedings of the IEEE Latin American Conference on Computational Intelligence (LA-CCI), Cartagena, Colombia, 2–4 November 2016; pp. 1–6.
27. Papandreou, C.; Ziakopoulos, A. Predicting VLCC fuel consumption with machine learning using operationally available sensor data. *Ocean Eng.* **2022**, *243*, 110321. [\[CrossRef\]](#)
28. Sharma, P.; Sahoo, B.B. Precise prediction of performance and emission of a waste derived biogas–biodiesel powered dual–fuel engine using modern ensemble boosted regression tree: A critique to artificial neural network. *Fuel* **2022**, *321*, 124131. [\[CrossRef\]](#)
29. Siqueira, H.; Belotti, J.T.; Boccato, L.; Luna, I.; Attux, R.; Lyra, C. Recursive linear models optimized by bioinspired metaheuristics to streamflow time series prediction. *Int. Trans. Oper. Res.* **2020**, *30*, 742–773. [\[CrossRef\]](#)
30. Rusli, R.; Hidayanto, A.N.; Ruldeviyani, Y. Consumption prediction on steam power plant using data mining hybrid Particle Swarm Optimization (PSO) and Auto Regressive Integrated Moving Average (ARIMA). In Proceedings of the IEEE International Workshop on Big Data and Information Security (IW BIS), Bali, Indonesia, 11 October 2019; pp. 15–20.
31. Weerasinghe, W.P.M.C.N.; Jayasundara, D.D.M. Modelling and forecasting the unit cost of electricity generated by fossil fuel power plants in Sri Lanka. *Int. J. Sci. Basic Appl. Res.* **2020**, *51*, 163–178.
32. Zhang, M.; Liu, Y.; Li, D.; Cui, X.; Wang, L.; Li, L.; Wang, K. Electrochemical impedance spectroscopy: A new chapter in the fast and accurate estimation of the state of health for lithium-ion batteries. *Energies* **2023**, *16*, 1599. [\[CrossRef\]](#)
33. Tuttle, J.F.; Vesel, R.; Alagarsamy, S.; Blackburn, L.D.; Powell, K. Sustainable NOx emission reduction at a coal-fired power station through the use of online neural network modeling and particle swarm optimization. *Control Eng. Pract.* **2019**, *93*, 104167. [\[CrossRef\]](#)
34. Al-Rawashdeh, H.; Hasan, A.O.; Gomaa, M.R.; Abu-jrai, A.; Shalby, M. Determination of carbonyls compound, ketones and aldehydes emissions from CI diesel engines fueled with pure diesel/diesel methanol blends. *Energies* **2022**, *15*, 7933. [\[CrossRef\]](#)
35. Singh, S.; Jain, A.; Mahla, S.K. Sampled-data model validation: An algorithm and experimental setup of dual fuel IC engine. *Fuel* **2020**, *279*, 118517. [\[CrossRef\]](#)
36. Yıldırım, S.; Tosun, E.; Çalık, A.; Uluocak, İ.; Avşar, E. Artificial intelligence techniques for the vibration, noise, and emission characteristics of a hydrogen-enriched diesel engine. *Energy Sources A Recovery Util. Environ. Eff.* **2019**, *41*, 2194–2206. [\[CrossRef\]](#)
37. Wang, Q.; Zhang, R.; Lv, S.; Wang, Y. Open-pit mine truck fuel consumption pattern and application based on multi-dimensional features and XGBoost. *Sustain. Energy Technol. Assess.* **2021**, *43*, 100977. [\[CrossRef\]](#)
38. Hu, L.; Wang, C.; Ye, Z.; Wang, S. Estimating gaseous pollutants from bus emissions: A hybrid model based on GRU and XGBoost. *Sci. Total Environ.* **2021**, *783*, 146870. [\[CrossRef\]](#) [\[PubMed\]](#)
39. Park, M.H.; Hur, J.J.; Lee, W.J. Prediction of oil-fired boiler emissions with ensemble methods considering variable combustion air conditions. *J. Clean. Prod.* **2022**, *375*, 134094. [\[CrossRef\]](#)
40. Siqueira, H.; Luna, I.; Alves, T.A.; de Souza Tadano, Y. The direct connection between box & Jenkins methodology and adaptive filtering theory. *Math. Eng. Sci. Aerosp.* **2019**, *10*, 27–40.
41. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the International Conference on Neural Networks, Perth, Australia, 27 November—1 December 1995; Service Center: Perth, Australia; pp. 1942–1948.
42. Bratton, D.; Kennedy, J. Defining a Standard for Particle Swarm Optimization. In Proceedings of the IEEE Swarm Intelligence Symposium (SIS), Washington, DC, USA, 1–5 April 2007; pp. 120–127.
43. Monteiro, R.P.; Lima, G.A.; Oliveira, J.P.; Cunha, D.S.; Bastos-Filho, C.J. Improving adaptive filters for active noise control using particle swarm optimization. *Int. J. Swarm Intell. Res.* **2018**, *9*, 47–64. [\[CrossRef\]](#)
44. Rapaić, M.R.; Kanović, Ž.; Jeličić, Z.D. Discrete particle swarm optimization algorithm for solving optimal sensor deployment problem. *J. Automat. Contr.* **2008**, *18*, 9–14. [\[CrossRef\]](#)
45. Xie, H.; Tang, H.; Liao, Y.H. Time series prediction based on NARX Neural Networks: An advanced approach. In Proceedings of the International Conference on Machine Learning and Cybernetics, Baoding, China, 12–15 July 2009; pp. 1275–1279.
46. Siqueira, H.; Luna, I. Performance comparison of feedforward neural networks applied to streamflow series forecasting. *Math. Eng. Sci. Aerosp.* **2019**, *10*, 41–53.
47. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
48. Kachba, Y.; Chirolì, D.M.d.G.; Belotti, J.T.; Antonini Alves, T.; de Souza Tadano, Y.; Siqueira, H. Artificial neural networks to estimate the influence of vehicular emission variables on morbidity and mortality in the largest metropolis in South America. *Sustainability* **2020**, *12*, 2621. [\[CrossRef\]](#)
49. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; ACM: San Francisco, CA, USA; pp. 785–794. [\[CrossRef\]](#)

50. Guo, R.; Zhao, Z.; Wang, T.; Liu, G.; Zhao, J.; Gao, D. Degradation state recognition of piston pump based on ICEEMDAN and XGBoost. *Appl. Sci.* **2020**, *10*, 6593. [[CrossRef](#)]
51. Si, S.; Zhang, H.; Keerthi, S.S.; Mahajan, D.; Dhillon, I.S.; Hsieh, C.J. Gradient boosted decision trees for high dimensional sparse output. In Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; PMLR: Sydney, Australia; pp. 3182–3190.
52. Dickey, D.A.; Fuller, W.A. Distribution of the estimators for autoregressive time series with a unit root. *J. Am. Stat. Assoc.* **1979**, *74*, 427–431. [[CrossRef](#)]
53. Kwiatkowski, D.; Phillips, P.C.B.; Schmidt, P.; Shin, Y. How sure are we that economic time series have a unit root? *J. Econom.* **1992**, *54*, 159–178. [[CrossRef](#)]
54. Teixeira, L.L.; Siqueira, P.H. Series streamflow forecasting with meta-heuristic PSO. *Rev. Ciências Exatas E Nat.* **2015**, *17*, 207–224. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.