

Article

Electrical Load Demand Forecasting Using Feed-Forward Neural Networks

Eduardo Machado ^{1,2}, Tiago Pinto ³ , Vanessa Guedes ² and Hugo Morais ^{1,4,*} 

¹ Instituto Superior Técnico-IST, Universidade de Lisboa, 1049-001 Lisbon, Portugal; eduardopraun@tecnico.ulisboa.pt

² Department of Materials, Energy Efficiency and Complementary Generation, Electrical Energy Research Center (Cepel), University City, Fundão Island, Rio de Janeiro 21941-911, Brazil; vanessag@cepel.br

³ GECAD—Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development, Rua DR. Antonio Bernardino de Almeida, 431, 4200-072 Porto, Portugal; tcp@isep.ipp.pt

⁴ INESC-ID, Department of Electrical and Computer Engineering, Instituto Superior Técnico-IST, Universidade de Lisboa, 1049-001 Lisbon, Portugal

* Correspondence: hugo.morais@tecnico.ulisboa.pt

Abstract: The higher share of renewable energy sources in the electrical grid and the electrification of significant sectors, such as transport and heating, are imposing a tremendous challenge on the operation of the energy system due to the increase in the complexity, variability and uncertainties associated with these changes. The recent advances of computational technologies and the ever-growing data availability allowed the development of sophisticated and efficient algorithms that can process information at a very fast pace. In this sense, the use of machine learning models has been gaining increased attention from the electricity sector as it can provide accurate forecasts of system behaviour from energy generation to consumption, helping all the stakeholders to optimize their activities. This work develops and proposes a methodology to enhance load demand forecasts using a machine learning model, namely a feed-forward neural network (FFNN), by incorporating an error correction step that involves the prediction of the initial forecast errors by another FFNN. The results showed that the proposed methodology was able to significantly improve the quality of load demand forecasts, demonstrating a better performance than the benchmark models.

Keywords: error correction; load demand forecast; feed-forward neural network



Citation: Machado, E.; Pinto, T.; Guedes, V.; Morais, H. Electrical Load Demand Forecasting Using Feed-Forward Neural Networks. *Energies* **2021**, *14*, 7644. <https://doi.org/10.3390/en14227644>

Academic Editor: Ahmed Abu-Siada

Received: 7 October 2021

Accepted: 6 November 2021

Published: 16 November 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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

According to the last International Panel on Climate Change (IPCC) report [1], there is strong evidence of human influence on global climate change, which is characterized by the global average temperature increase, higher ocean levels and the occurrence of catastrophic events, among others. The rise in greenhouse gas (GHG) concentration in the atmosphere is most likely the main driver of the changes in the Earth's climate. In this sense, the energy sector accounts for a quarter of the world's GHG emissions [1], being the one with the largest share. Therefore, it is not possible to discuss climate change without considering the energy sector.

Changes in climate conditions can affect physical, biological and human systems at global and regional scales and require adaptation and mitigation measures. In order to reduce GHG emissions, several national, regional and global agreements have been signed, such as the European Union (EU) 2009/28/EC directive [2], in which it was agreed that EU countries must fulfil at least 32% of their total energy needs with renewable energy by the year of 2030. More recently, as part of the European Green Deal, the European Commission proposed the first European Climate Law to enshrine the 2050 climate neutrality target into law [3]. This type of measure is in accordance with the findings presented in [4], which state that large-scale changes in the energy system are a must to achieve this goal and mitigate the global climate crisis. However, the energy transition towards cleaner

sources demands profound and challenging changes in the sector's infrastructure, policies, regulations, market design and operation.

The aging of conventional power plants, technological advances and cost reductions are allowing cleaner sources, mainly solar and wind-based systems, to boost their share in the electricity mix at the expense of fossil fuels [5]. Simultaneously, the electrification of significant sectors, such as transport and heating, is increasing the load demand while system decentralization is altering the load patterns and the energy flow, as consumers are changing their roles to become prosumers, i.e., someone who both produces and consumes energy. Moreover, the growth of digital and storage technologies also increases system complexity and can expose it to cyber threats [6]. All this transformation requires various adaptation measures, which involves technical, economical and political issues [6] that must be applied to ensure reliable, affordable, safe and high-quality electricity.

The uncertainty of energy supply and demand can cause grid instability issues, such as overvoltage and frequency deviations. To overcome this situation, the system must be flexible and resilient enough in order to cope with rapid generation and load changes and balance them at every moment. In this regard, recent technologies and approaches such as small-scale energy storage systems (e.g., batteries) and demand response programs have been gaining increased attention [6]. However, they are not always feasible as they can be extremely expensive and still need more improvement to be applied at a large scale. On the other hand, one common way for system operators to deal with this variability is by defining a certain amount of energy reserve that can be used to adjust the system frequency. In liberalized markets, such as the Iberian Electricity Market (MIBEL), the necessity and use of this reserve can configure an extra cost to the system operation and lead to an increase in the electricity price.

The above-mentioned solutions have in common the fact that they try to deal with the system variability by giving immediate responses to instantaneous deviations, by discharging batteries, turning off electrical appliances or increasing the power of a generator, for example. Nonetheless, they all typically have their behavior planned based on power generation and load demand predictions. Therefore, accurate forecasts are essential for these tools to optimize their performances and, consequently, the whole system operation.

The interest in energy demand and supply forecasts has significantly increased since the oil crisis in the 1970s [7]. At that time, most of the employed models were statistics-based such as linear regression, ordinary least squares and stepwise regression, among others. However, the growing complexity of electrical systems and the unknown relationships between multiple variables impose great difficulties that these simpler models cannot handle.

On the other hand, a boost in the application of computational methods has been observed in recent years. The technological developments in computing allowed the creation of sophisticated and efficient computational methods that typically use advanced statistical concepts. They can be combined with statistical models, used to estimate their parameters or to make forecasts with reduced computational time and improved performance [7].

In the present paper, a machine learning model based on a feed-forward neural network (FFNN) for a load demand forecast is proposed. The main novelty of this method is that the FFNN is first applied to the historical data of load measurements, and, in a second step, the same method is applied to the results of the load forecast to estimate the errors of the initial load forecast. Finally, the initial load forecast is adjusted considering the errors forecast, providing a very accurate load demand forecast. The consumption forecast can be used by several stakeholders for different purposes such as transmission and distribution system management, support to market participation or energy management in energy communities.

The increased need for electricity and the change in the power generation mix and load patterns are some of the already observed transformations. With all this, electricity grid management becomes more unpredictable and its operation and control more complex,

which can lead to greater supply instability. Therefore, techniques and methods capable of increasing system reliability are extremely necessary.

One way to ensure more trustworthiness and better management of the system is by anticipating the load demand. When accurate forecasts are made, the decisions regarding the power system operation, maintenance and planning become more efficient [8,9]. Furthermore, improvements on energy policies and tariffs can be achieved. In recent years, much of the research has been focused on the development of models to forecast the electrical load in different time horizons. These periods are often classified as short term, which goes up to 1 week ahead; medium term, from weeks to 1 year; and long term, for future years [10]. Additionally, each of these timeframes have different applications, with the first being more important for daily operation and cost minimization [11] and the others for fuel reserves estimation, maintenance and capacity expansion planning [12].

Several approaches have been employed recently to make these forecasts. These approaches can be separated into three categories: statistics-based, computational intelligence-based and hybrid approaches [7]. Statistical models usually embrace uni or multivariate time-series models and regression techniques, such as Autoregressive Integrated Moving Average (ARIMA) and Linear Regression, while computational intelligence models are mainly related to ML approaches. Commonly, statistics-based methods are less memory intensive due to their simplicity and, thus, faster to execute. On the other hand, ML models are capable of identifying nonlinear relationships between inputs and outputs and can be extremely time consuming. However, this level of complexity can be necessary to achieve better results [10]. Finally, hybrid models combine features from statistical and computational intelligence models. They generally use the former to preprocess and/or select the input data that will be fed to the latter.

These forecast models can be implemented using a large range of inputs that can be divided into four major categories: socio-economic, such as the regional average income and GDP; environmental, such as mean temperature; building and occupancy, which is related to building sizes and dwelling types; and time index, which is related to the date stamps used as inputs [13]. Additionally, electricity demand historical data are generally taken into consideration. However, the choice of these inputs will depend on the time scale and type of the region of the study. Usually, historical, environmental and time index data are more common for short-term forecasts in a region scale [13].

Recently, several neural network structures have been employed to develop for short-term load forecasting. In [14], a model composed of a Convolutional Neural Network (CNN), a bidirectional Gated Recurrent Unit (GRU) and a Long Short-Term Memory (LSTM) recurrent neural network (RNN) was presented. First, the authors computed hourly autocorrelation coefficients of hourly loads and temperature, which were used to calculate the kernel size of convolutional layers. Later, two-dimensional convolutional layers with the load and temperature time-series were used to extract features from these data [14,15].

In [16], a hybrid Artificial Neural Network (ANN) to forecast the day-ahead load in a smart grid was proposed. The strategy was divided into three modules: pre-processing, forecasting and optimization. The goal of the first module was to remove irrelevant and redundant features from the dataset by applying a mutual information-based technique. Mutual information (MI) represents the uncertainty reduction about one variable as a consequence of observing another one and is a concept related to the information theory [17]. In [18], the authors introduced an Advanced Wavelet Neural Network (AWNN) to forecast one-step-ahead load demand. The proposed approach was composed of four stages: load decomposition, feature selection, prediction model creation and forecasting. The main idea of data decomposition is to break the data into their constituent parts to find universal or functional properties that are not observed in their usual representation [15].

In [19], an LSTM network using a cross-correlation-based transfer learning approach was proposed to forecast 15-min-ahead load demand. Transfer learning is a methodology that identifies similarities in different datasets and allows the use of knowledge from other

tasks on related ones [15]. This approach can be extremely useful when the available data are scarce. In this work, energy demand data from several randomly selected buildings over approximately one year were collected for the transfer learning step, while the load demand to be estimated came from data collected from a university building in Turkey over one month. Additionally, both of them were sampled at every 15 min. The results showed that the proposed model was able to outperform the benchmark models in terms of the Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Moreover, the contribution of the transfer learning approach was evident, as a significant improvement in the LSTM model could be verified. The authors also observed that the best proposed models came from those weights obtained with the data from the buildings with the highest cross-correlation. However, the MAPE was higher than the one observed in other works (about 8%), and details about the performance of these models for each day and time were not provided.

In [12], ML approaches including ANN, Multiple Linear Regression (MLR), Adaptive Neuro-Fuzzy Inference System (ANFIS) and Support Vector Regression (SVR) were employed to forecast week and year-ahead electricity demand in Cyprus. The inputs for the models were time, environmental data such as temperature, humidity and solar irradiation and socio-economic data such as population, gross national income per capita and electricity price. The results showed that the ANN and the SVR models performed significantly better than the other two for both short-term and long-term forecasts, with the MAPE around 2% in the first scenario and 5% in the second.

In [20], a hybrid model that combined ARIMA with SVM was applied to forecast hourly load demand. This work used historical data from a state in the south of India to estimate the ARIMA parameters and generate initial load forecasts. Then, the outliers of the initial predictions were detected by means of the percentage error method and corrected using the deviation method. The forecast errors data of the corrected ARIMA model output and the other two variables, namely day and average temperature of the week, were given as inputs for the SVM model to estimate the initial forecast error. Finally, the initial forecasts and the expected errors were added to obtain the final load prediction. The authors found that the proposed ARIMA–SVM model was able to outperform single ARIMA and SVM models in terms of MAPE (4.15% versus 5.16% and 4.97%, respectively). Furthermore, the performance of the proposed model without the outlier detection approach was worse (6.23% versus 4.15%). In [21], SVM was proposed for fault prediction of specific loads.

Another hybrid model was proposed in [22], which combined an SVR model and a two-step parameter optimization algorithm using the Grid Traverse Algorithm (GTA) and Particle Swarm Optimization (PSO) to forecast the load demand in several short-term scales (from 5 min to 16 h ahead). The tested data were comprised of 80 days of load from a distribution feeder. In the first moment, these data were pre-processed to eliminate excessively deviating samples using a mapping algorithm. Then, a GTA designed with cross-validation was used to narrow the SVR parameters' search area. After, the PSO searched for the best parameters of the SVR model in the GTA solution space also using cross-validation.

In [23], three ensemble learning algorithms, namely Random Forest (RF), Gradient Boosted Regression Tree (GBRT) and Adaboost (AR2), were employed to forecast one-hour-ahead electricity load. Ensemble models are those that combine several models, which are trained separately in order to reduce the generalization error [15]. The main advantage of this type of model is that, on average, they perform at least as well as any of its members and, if the errors of its members are independent, they will perform significantly better. In this work [23], historical electricity demand from an office building was collected at every 10 s and averaged to a one-hour basis. Different training strategies merging various features such as historical, time index and environmental data were tested to forecast every hour of a single day. The authors found that using the time index data with the most recent temperature measurements and a few past load observations provided the best forecasts in general. The results of this work were compared to the ones in [24], which

generated forecasts for the same building using SVM and three fuzzy rule-based models. It was observed that the AR2 model outperformed the SVM model in terms of MAPE (5.34% vs. 5.82%). Moreover, the other two ensemble models performed better than the fuzzy rule-based models.

In [25], a short-term load forecasting model based on error correction using dynamic mode decomposition (DMD) was proposed. Using two years of electricity demand data from a city in China, the authors built several load forecasting models including ANN, SVR and ARIMA. They used historical data, namely previous day, same day in the previous week and similar day loads, as inputs. The latter was obtained by grey relational analysis, which is a method that aims to find highly correlated data. With the errors achieved by these models, the DMD was applied to forecast the errors. The DMD is a data-driven method that can extract complex spatiotemporal features from data. Extensive experiments on small and large geographical scales were conducted to evaluate the proposed methodology. Nearly in all cases, combining the DMD for the error correction with the load forecasting model resulted in better results. Additionally, different decomposition techniques including Wavelet Transform were tested and generally had a worse performance. Finally, it was possible to notice a smaller prediction accuracy for the small area forecasts, which was probably caused by its higher load variability. In Table 1, a summary of previous works can be seen.

Table 1. Summary of previous works.

Reference	Year	Forecasting Model	Dataset Size	Sample Rate	Input Data	Forecast Horizon	Metrics
[12]	2020	ANN, MLR, ANFIS, SVR	2 years	15 min	Time index, socio-economic, environmental	1 year, 1 week	MAPE, RMSE
[14]	2021	CNN, GRU, LSTM	7 and 11 years	1 h	Historical, environmental	1 h	MAPE, MAE, RMSE
[16]	2019	ANN	2 years	1 h	Historical, environmental	24 h	MAPE, Variance
[18]	2016	AWNN	2 years	5 and 60 min	Historical	5 and 60 min	MAPE, MAE
[19]	2021	LSTM	14 months	15 min	Historical, time index	15 min	MAPE, MAE, RMSE
[20]	2017	ARIMA, SVM	2 years	1 h	Historical, time index, environmental	24 h	MAPE
[22]	2018	SVR	80 days	1 s	Historical	From minutes to hours	MAPE
[23]	2020	RF, GBRT, AR2	10 days	1 h	Historical, time index, environmental	1 h	MAPE
[24]	2018	SVM	11 weeks	10 s	Historical, building, environmental	24 h	MAPE
[25]	2020	DMD	2 years	15 min, 1 h	Historical, time index, environmental	24 h	MAPE, MAE, RMSE, Variance, Direction Accuracy

Several works have been published in the field of consumption forecasts. Nevertheless, most of the methods are very simple and not accurate and others are very complex and not useful in many applications where computational resources or time are limited. The present paper has the following main contributions:

- Propose a consumption forecast method assuring a balance between the accuracy and the computational effort/costs;
- Propose a new methodology comprising the forecast of consumption based on historical data and the forecast of the error of the initial forecast. Combining the two forecasts, we can improve the accuracy of the final results.

2. Proposed Methodology

In the present work, the FFNN was selected to develop the proposed methodology. The term neural network comes from the fact that these models were inspired by biological brains in the sense of how they process information. A neural network model is typically composed of nodes (or neurons) that are distributed across different layers, namely input, hidden and output layers. Each node in a layer is linked to the ones in the next by means of a weight parameter that measures the strength of that connection, forming a fully connected network structure that resembles the nervous system. In Figure 1, a general neural network is illustrated.

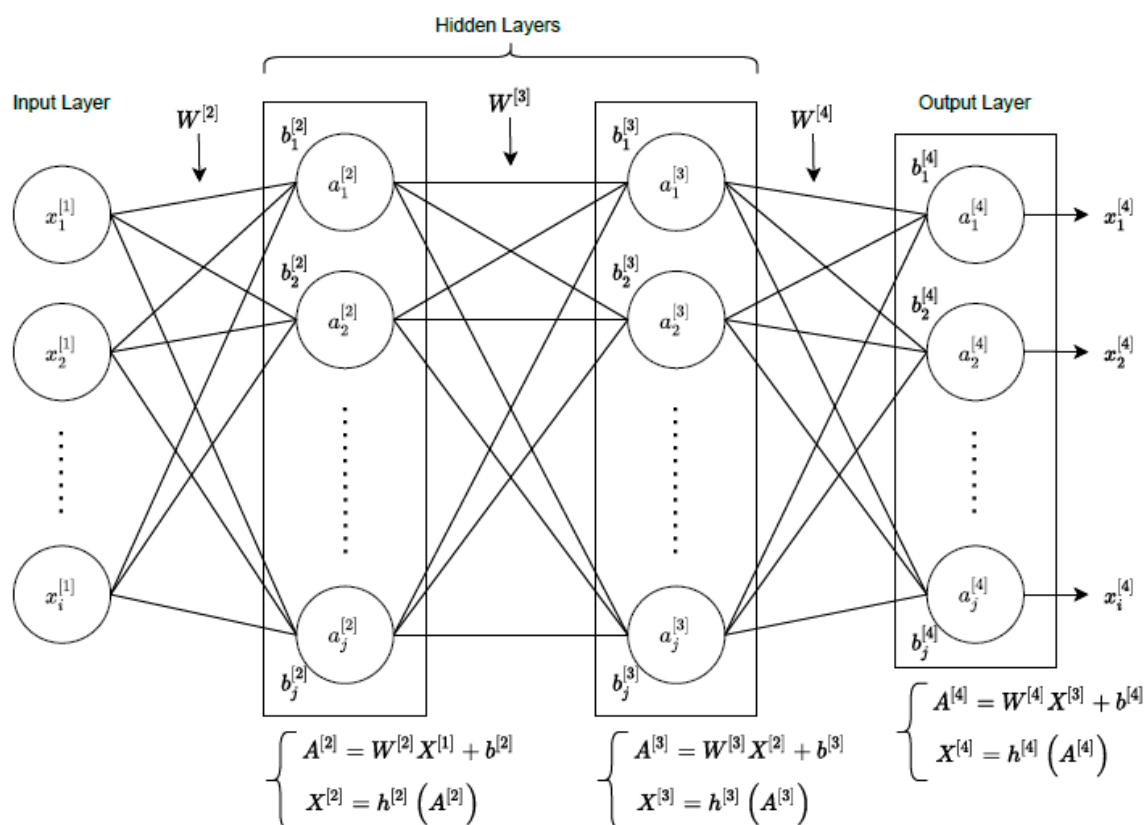


Figure 1. General neural network model structure.

The operating principle of neural networks can be described as a sequence of functional transformations [17]. For a given layer $l \in \{1, \dots, L\}$, where L is the number of layers, quantity, called the activation value $A^{[l]} = (a_1^{[l]}, \dots, a_j^{[l]})^T$, can be calculated as a linear combination of inputs $X^{[l-1]} = (x_1^{[l-1]}, \dots, x_i^{[l-1]})$ and weights $W^{[l]}$ in the form

$$A^{[l]} = W^{[l]}x^{[l-1]} + b^{[l]} \quad (1)$$

where

$$W^{[l]} = \begin{pmatrix} \omega_{1,1}^{[l]} & \omega_{1,2}^{[l]} & \cdots & \omega_{1,j}^{[l]} \\ \omega_{2,1}^{[l]} & \omega_{2,2}^{[l]} & \cdots & \omega_{2,j}^{[l]} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_{i,1}^{[l]} & \omega_{i,2}^{[l]} & \cdots & \omega_{i,j}^{[l]} \end{pmatrix} \quad (2)$$

and $b^{[l]} = (b_1^{[l]}, \dots, b_j^{[l]})^T$ is a parameter known as bias, which is used to adjust the output. The subscripts i and j represent the number of nodes or dimension of layers $l - 1$ and l , respectively.

Then, the activation value $A^{[l]}$ is transformed by a nonlinear, differentiable function $h^{[l]}(.)$ that is named activation function as in Equation (3), resulting in the next layer input vector $X^{[l]}$. For hidden layers, the activation function is a hyperbolic tangent function (tanh) or a rectified linear unit (ReLU), while for the output layer it is the identity function.

$$X^{[l]} = h^{[l]}(A^{[l]}) \quad (3)$$

Equations (1) and (3) present recursive calculations that constitute a process known as forward propagation [17]. This name comes from the fact that the information is flowing forward through the network, and this is the reason why this type of model is called FFNN. There are some particularities about these equations that should be mentioned: the first input vector $X^{[0]}$ comes from the features selected from the dataset, while the remaining are the result of the calculations. Additionally, the final result, observed in $X^{[L]}$, is the output of the model.

The parameter optimization is performed with gradient descent-based calculations (Equation (4)). With this approach, the required partial derivatives are related to the two parameters of an FFNN model: the weights and biases. Applying Equation (4) to the last layer of this model results in Equations (5) and (6).

$$p_{[i+1]} = p_{[i]} - \eta \nabla E(p) \quad (4)$$

where p represents the model parameters, η is the learning rate (LR) and $E(p)$ is the loss function in Equation (5), which is the Mean Squared Error (MSE), also called Euclidean or L2 norm. In Equation (5), x_i is the forecasted value, t_i is the target value and n is the number of points in the dataset. The search for the loss function minimum is commonly performed by computing its gradient ($\nabla E(p)$), which is the vector containing the partial derivatives of $E(p)$ [15]. The partial derivative $\frac{\partial}{\partial p} E(p)$ indicates how the function changes with a small change in one of the parameters. Therefore, the gradient vector points to the direction of the steepest increase in the function. As the learning algorithm goal is to minimize the error, with this approach the parameters can be updated at each iteration i by going in the opposite direction.

$$E(p) = \text{MSE} = \frac{1}{n} \sum_{i=1}^n (f(x_i, p) - t_i)^2 \quad (5)$$

$$W_{[i+1]}^{[L]} = W_{[i]}^{[L]} - \eta \frac{\partial E}{\partial W^{[L]}} \quad (6)$$

$$b_{[i+1]}^{[L]} = b_{[i]}^{[L]} - \eta \frac{\partial E}{\partial b^{[L]}} \quad (7)$$

Using the chain rule, one can write these partial derivatives as Equations (8) and (9):

$$\frac{\partial E}{\partial W^{[L]}} = \frac{\partial E}{\partial X^{[L]}} \circ \frac{\partial X^{[L]}}{\partial A^{[L]}} \cdot \frac{\partial A^{[L]}}{\partial W^{[L]}} \quad (8)$$

$$\frac{\partial E}{\partial b^{[L]}} = \frac{\partial E}{\partial X^{[L]}} \circ \frac{\partial X^{[L]}}{\partial A^{[L]}} \cdot \frac{\partial A^{[L]}}{\partial b^{[L]}}^T \quad (9)$$

where the dot (.) symbol stands for matrix multiplication and the circle (o) symbol for the Hadamard or element-wise product. At this stage, it is useful to introduce the following notation:

$$\delta^{[L]} = \frac{\partial E}{\partial X^{[L]}} \circ \frac{\partial X^{[L]}}{\partial A^{[L]}} \quad (10)$$

where $\delta^{[L]}$ is a value known as delta and represents the error that the layer $L-1$ sees. Moving on to the layer $L-1$, the calculations are as in Equations (11)–(13).

$$\frac{\partial E}{\partial W^{[L-1]}} = \frac{\partial A^{[L]}}{\partial X^{[L-1]}}^T \cdot \left(\frac{\partial E}{\partial X^{[L]}} \circ \frac{\partial X^{[L]}}{\partial A^{[L]}} \right) \circ \frac{\partial X^{[L-1]}}{\partial A^{[L-1]}} \cdot \frac{\partial A^{[L-1]}}{\partial W^{[L-1]}}^T \quad (11)$$

$$\frac{\partial E}{\partial b^{[L-1]}} = \frac{\partial A^{[L]}}{\partial X^{[L-1]}}^T \cdot \left(\frac{\partial E}{\partial X^{[L]}} \circ \frac{\partial X^{[L]}}{\partial A^{[L]}} \right) \circ \frac{\partial X^{[L-1]}}{\partial A^{[L-1]}} \cdot \frac{\partial A^{[L-1]}}{\partial b^{[L-1]}}^T \quad (12)$$

$$\delta^{[L-1]} = \frac{\partial A^{[L]}}{\partial X^{[L-1]}}^T \cdot \delta^{[L]} \circ \frac{\partial X^{[L-1]}}{\partial A^{[L-1]}} \quad (13)$$

From layer $L-1$ to the first layer, it is possible to write the next deltas as Equation (14):

$$\delta^{[l]} = \frac{\partial A^{[l+1]}}{\partial X^{[l]}}^T \cdot \delta^{[l+1]} \circ \frac{\partial X^{[l]}}{\partial A^{[l]}} \quad (14)$$

and, therefore, the partial derivatives can be obtained as Equations (15) and (16):

$$\frac{\partial E}{\partial W^{[l]}} = \delta^{[l]} \cdot \frac{\partial A^{[l]}}{\partial W^{[l]}}^T \quad (15)$$

$$\frac{\partial E}{\partial b^{[l]}} = \delta^{[l]} \cdot \frac{\partial A^{[l]}}{\partial b^{[l]}}^T \quad (16)$$

Finally, the parameter updates can be performed using Equations (15) and (16) with Equations (6) and (7), respectively. The process presented above constitutes the backpropagation learning algorithm. With this approach, the algorithm goes through each layer in reverse, measuring the error contribution from each connection by means of the deltas and updating the parameters accordingly [26]. By computing the gradient in reverse, the backpropagation algorithm avoids unnecessary calculations as it reuses previous ones. This is the major reason for this method's higher computational effectiveness when compared to numerical methods such as finite differences [17] and one of the cornerstones to FFNN's popularity.

To improve the accuracy of the forecast methodology proposed in the present work, the FFNN algorithm is executed two times. First, electric load measurements are used as input for FFNN to obtain the first iteration of the load forecast. Afterwards, comparing the obtained values with the measured values, it is possible to compute the errors of the method. These errors are used as inputs for a second FFNN predictor. Finally, the errors forecasts are merged with the initial loads forecasts to obtain the global loads forecasts. This process is illustrated in Figure 2.

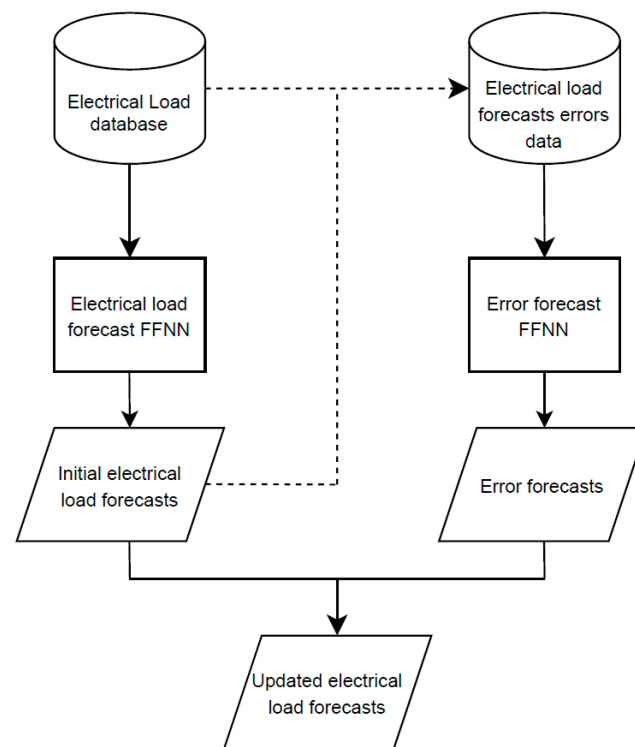


Figure 2. Electrical load forecast methodology.

3. Case Study

3.1. Database and Preliminary Analysis

The collected data refer to the measured load demand in an industrial area connected to the medium voltage grid and were sampled every 10 min. Furthermore, they came in a .csv format, which could be directly used by the Python scripts that were developed later. The retrieved period ranges from 1 October 2016 to 31 March 2017 in a total of 26,208 samples. In Figures 3 and 4, an overview of the power load measurements is presented.

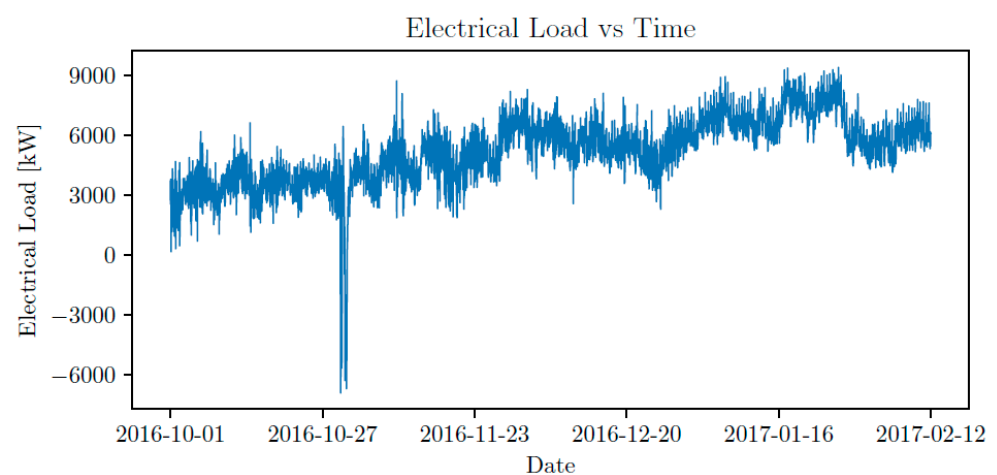


Figure 3. Electrical load measurements.

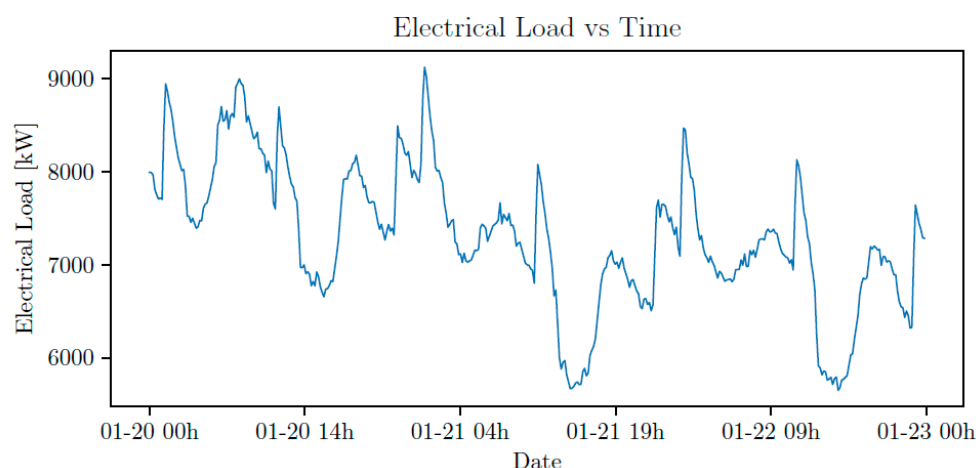


Figure 4. Electrical load measurements (detail).

A preliminary analysis of the electrical load data was carried out in order to find relevant information that could be used to prepare the models. At first, from Figure 3, it is noticeable that the retrieved data had some noisy samples close to 30 October 2016 as the load pattern drastically changes. These points were removed after the application of a statistical test that will be detailed in the next section. Additionally, it appears that the load demand has a growth trend in this period. Moreover, from Figure 4, it is possible to observe that the load behavior is cyclical throughout the days and that several peaks take place around 02:00, 12:00 and 23:00. In Figures 5 and 6, this pattern also seems to change depending on the weekday, with lower power demand on weekends, and on the month, increasing significantly from October to January and decreasing in February.

Figure 7 shows the autocorrelation function (ACF) plot for the electrical load data. This graph is frequently used to analyze time series and shows the correlation between the time series and past observations (lags). The autocorrelation coefficient ranges from -1 to 1 and measures how strong the relationship between two variables is, with 1 indicating a strong positive relationship, -1 strong negative relationship and 0 meaning no relationship. The values above the blue area of the graph are those that are statistically significant ($p > 95\%$).

Therefore, in this case, the plot suggests that the previous power demand is strongly related to the current one for lags up to 7 days (1008 periods). The ACF also shows that there is a descending trend of the autocorrelation coefficient as time progresses and that spikes occur at the end of each day period (lags 144, 288, etc.).

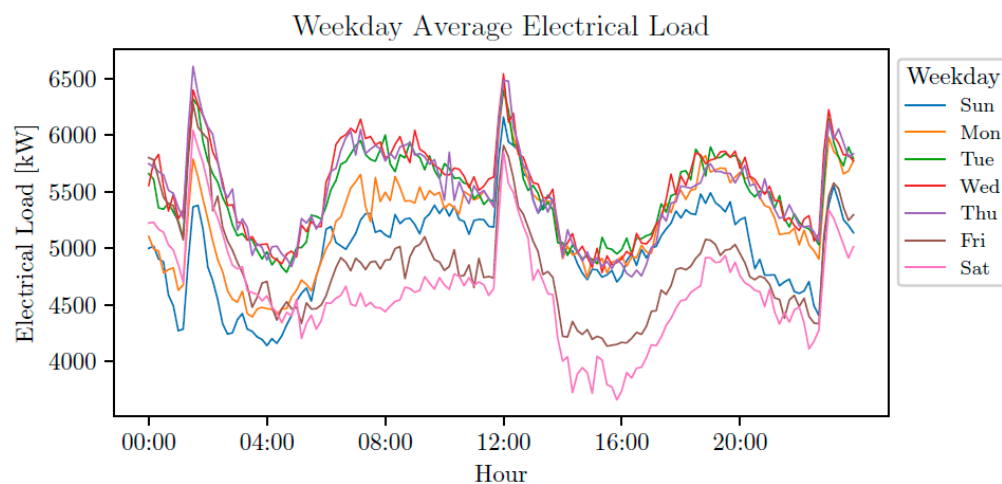


Figure 5. Electrical load: daily average.

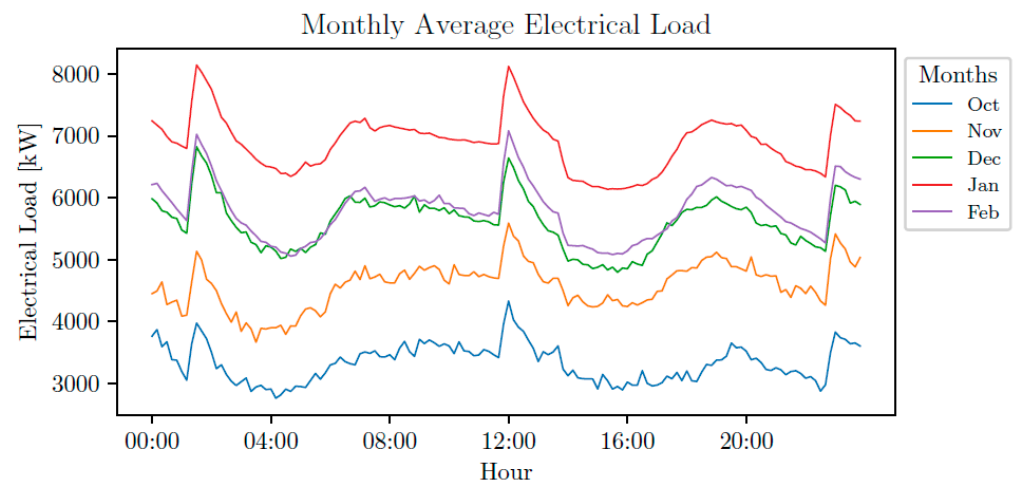


Figure 6. Electrical load monthly average.

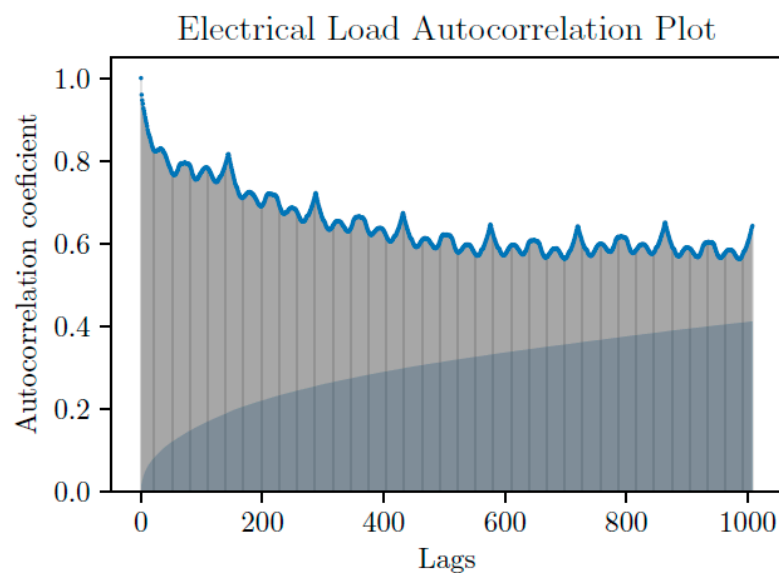


Figure 7. Electrical load database autocorrelation.

3.2. Load Forecast Model Parameterization

The methodology described in Section 4 was implemented in Python's TensorFlow library, together with scikit-learn, numpy and others. Three different forecast horizons, 10 min ahead, 1 h ahead and 12 h ahead, were tested. For each horizon, six different input combinations were evaluated. They were divided into two categories, one considering only the previous recorded loads and the other also including the sine and cosine transformations of time. The choice of these inputs was based on the preliminary analysis already described and did not include other possible explanatory variables such as the weekday or month to keep the models simpler and to verify if the selected inputs would be able to reach good results. Therefore, the configurations were:

(a) Inputs without time:

1. Loads at the previous six (6) periods;
2. Loads at the previous three (3) periods and three (3) days at the time of the forecast;
3. Loads at the previous three (3) periods, three (3) days and three (3) weeks at the time of the forecast;

(b) Inputs with time:

1. Loads at the previous six (6) periods and time of the forecast (transformed into sine and cosine);
2. Loads at the previous three (3) periods, three (3) days at the time of the forecast and time of the forecast (transformed into sine and cosine);
3. Loads at the previous three (3) periods, three (3) days and three (3) weeks at the time of the forecast and time of the forecast (transformed into sine and cosine).

In the first moment, 20% of the data were separated (≈ 5000 samples) to be tested by the initial electrical load model, while the other 80% were used as the training set. In order to have more reliable results, the models were trained with three-fold cross validation. Before feeding the data into the models, the Z-score, also called standard score, of each sample in the training set was calculated as in Equation (17), and those that were more than 3 times the standard deviation away from the set mean were removed ($Z_{score} > 3$).

$$Z_{score} = \frac{x - \mu}{\sigma} \quad (17)$$

In Equation (17), x is the sample value, μ stands for the set mean and σ represents its standard deviation. Later, the training data were scaled to be between 0 and 1. Data scaling is a common pre-processing technique that can improve the model learning process. After the training, the models were ranked based on the average of the validation set root mean squared error (RMSE) over the three folds. Additionally, the MAE was another metric used to evaluate the performances.

To find the best parameters for the FFNN, a grid search was performed. In an initial search, the number of nodes varied from 12 to 48 in steps of 12 with the hyperbolic tangent or rectified linear unit activation function (Equations (18) and (19)).

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (18)$$

$$f(x) = \max(0, x) \quad (19)$$

This resulted in a total of eight models per input configuration that are presented in Table 2. At this stage, some parameters were kept constant, such as the number of layers (3), the LR (0.01), the number of maximum epochs (1500) and batch size (144). The chosen optimizer was the Adam algorithm, which is a stochastic gradient descent method based on adaptive estimation of the first and second-order moments [27]. Additionally, to avoid overfitting, they were programmed to stop the training if the validation MSE did not decrease between five consecutive epochs, an approach known as early stopping. After this step, the model with the best overall performance for each time horizon was further explored by changing the LR to 0.005 and 0.001. Finally, the best configuration had its results compared to those of the persistence model in the 10-min horizon and linear regression of the same inputs in the 1-h and 12-h horizons by making predictions on a previously unseen test set.

Table 2. Electrical load models.

Model	Nodes in the Hidden Layer	Activation Function
1	12	ReLU
2	24	ReLU
3	36	ReLU
4	48	ReLU
5	12	tanh
6	24	tanh
7	36	tanh
8	48	tanh

3.3. Load Forecast—Results of First Forecast Step

The initial load forecasts for each time horizon are presented in this section. The tables in this subsection present only the results of the best model for each horizon in order to make the text more fluid.

3.3.1. Ten-Minute Horizon

The results obtained for the initial forecast (Step 1) with the best 10-min-ahead model are presented in Table 3. The best configuration in this scenario had the previous three periods and days as inputs. Configuration I, which was the original one with an LR of 0.01, achieved almost the same results of configuration II with a lower execution time (ET). This probably happened due to the higher LR. On the other hand, model III appeared to learn at a much slower pace and did not reach the same results. When compared to the persistence model, the RMSE and the MAE of the best proposed model were 10.6% and 23.8% higher. These findings are somehow understandable as making predictions for such a short time in the future can be easier by assuming that the next value will be the same as the one observed now. Indeed, the proposed ANN converged to a solution that generated forecasts similar to those of the persistence model (Figure 8). However, giving inputs other than the previous recorded value to the model seemed to hinder its predictions.

Table 3. Results of the 10-min-ahead initial and baseline models.

Model	LR	RMSE (kW)	MAE (kW)	Test RMSE (kW)	Test MAE (kW)	Test MAPE (%)	ET (s)
I	0.01	223.00	159.34	234.24	177.66	3.61	354.13
III	0.005	223.62	160.90	-	-	-	524.84
III	0.001	247.98	179.50	-	-	-	548.39
Baseline	-	-	-	211.72	143.53	2.84	-

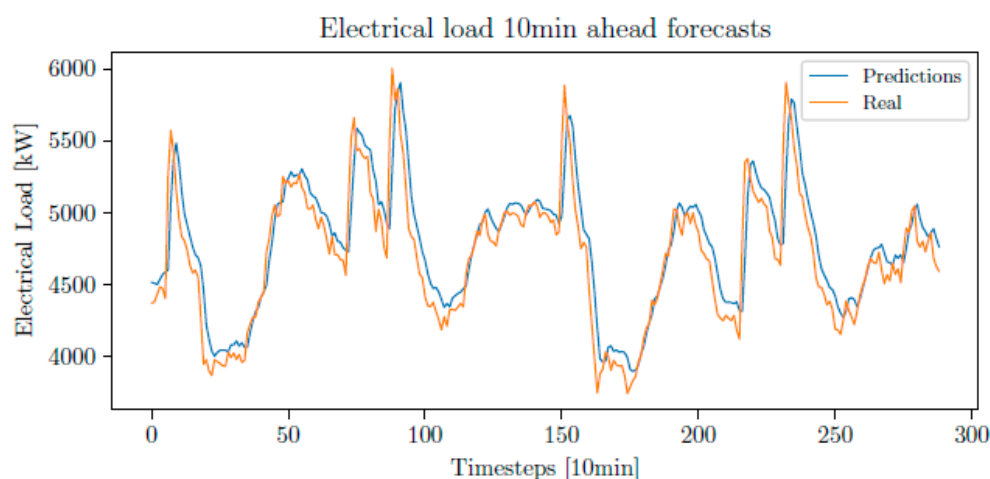


Figure 8. Ten-minutes-ahead initial forecasts.

3.3.2. One-Hour Horizon

The results of the best 1-h-ahead model are shown in Table 4. Here, the best configuration also took the previous three periods and days as inputs; however, the arrangement with time performed better. Similarly to the previous case, the model with the original LR had better results than the other two but with similar execution times, which suggests that the higher LR enabled the model to be closer to a loss curve minimum at the end of the training. Furthermore, it achieved superior performance than the linear regression model in the order of 10.7% over the RMSE and of 11.3% over the MAE. In Figure 9, the forecasts made with the ANN can be seen. It shows that the model was able to make accurate

predictions but missed some of the load spikes and had a slightly smoother behavior when compared to the actual values.

Table 4. Results of the 1-h-ahead initial and baseline models.

Model	LR	RMSE (kW)	MAE (kW)	Test RMSE (kW)	Test MAE (kW)	Test MAPE (%)	ET (s)
I	0.01	339.43	261.33	376.25	303.76	6.41	513.99
II	0.005	357.58	276.05	-	-	-	546.20
III	0.001	1037.59	736.56	-	-	-	515.76
Baseline	-	-	-	421.48	342.30	6.96	-

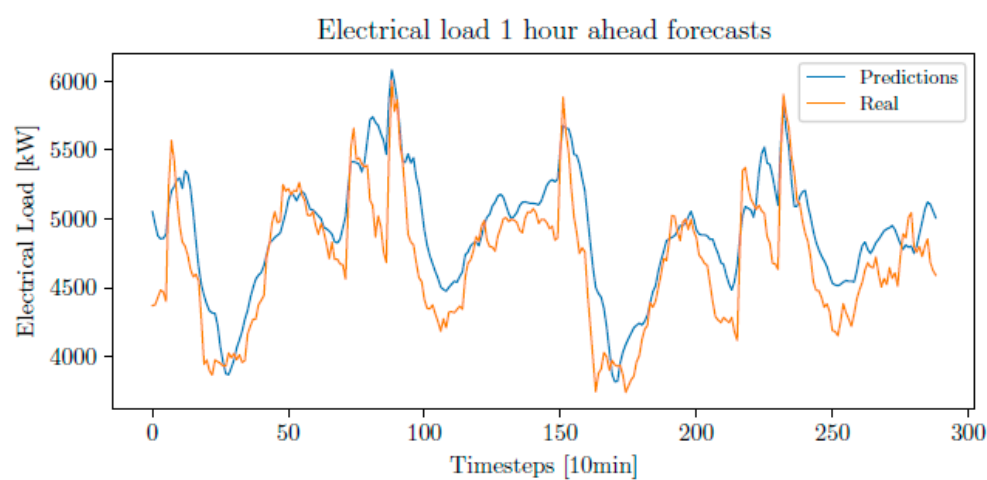


Figure 9. One-hour-ahead initial forecasts.

3.3.3. Twelve-Hour Horizon

Figure 10 presents the results of the best 12-h-ahead model. The best configuration was the same as the 10-min horizon with only the previous three periods and days as inputs. Here, decreasing the LR to 0.005 enhanced the forecasts when compared to the other two configurations. The lower execution time combined with the worse performance of the original configuration suggests that an LR of 0.01 was too high and the training stagnated and stopped at some point, while for model III it was too low and did not allow the results to improve sufficiently. In comparison to the linear regression model, the performance of the best proposed model was 12.6% and 13.5% superior in terms of the RMSE and the MAE (Table 5). Figure 10 shows results that were similar to the previous case. Yet, the predictions seemed more inaccurate, with more substantial errors.

Table 5. Results of the 12-h-ahead initial and baseline models.

Model	LR	RMSE (kW)	MAE (kW)	Test RMSE (kW)	Test MAE (kW)	Test MAPE (%)	ET (s)
I	0.01	548.03	432.91	-	-	-	184.89
II	0.005	543.55	431.31	501.46	399.92	8.36	531.64
III	0.001	547.33	520.63	-	-	-	507.03
Baseline	-	-	-	573.76	462.44	9.67	-

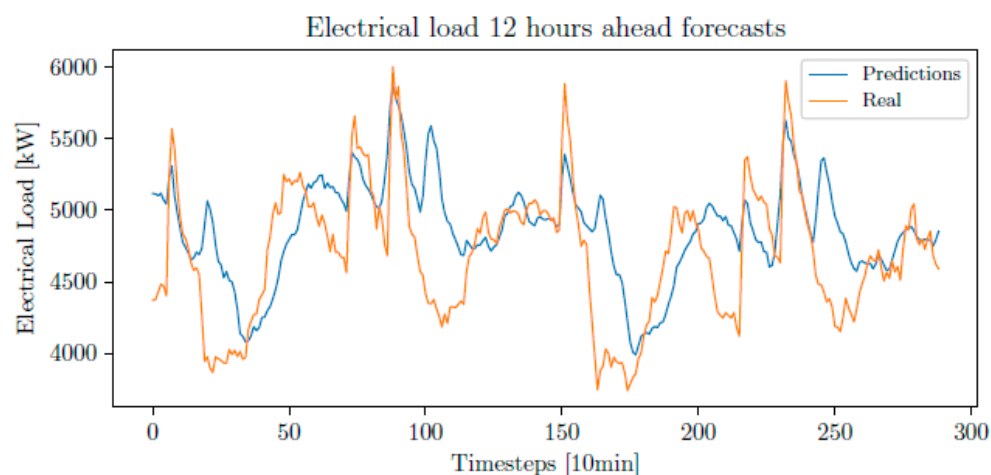


Figure 10. Twelve-hours-ahead initial forecasts.

3.3.4. Summary of Forecast Results in Step 1

An interesting observation that was valid for all the models is that often those with lower execution times had the worst performances, which can suggest that they were stuck in a local minima and were not able to progress with the chosen LR. Additionally, it seemed that the differences of the number of nodes in the hidden layer and the activation function from one model to another did not impact the execution times.

Finally, Table 6 summarizes the results of the proposed initial models for the three forecast horizons. These results showed that the forecast accuracy decreased as the forecast horizon increased and indicated the greater degree of difficulty to make longer forecasts due to the higher uncertainty of the data, justifying the use of more complex models. Furthermore, for very short-term forecasts, simple methods such as the persistence model can provide relatively good estimates with low computational resources.

Table 6. Summary of the adjusted models for the three horizons.

Horizon	RMSE (kW)	MAE (kW)	MAPE (%)
10-min	234.24	177.66	3.61
1-h	376.25	303.76	6.41
12-h	501.46	399.92	8.36

3.4. Load Forecast—Error Forecast (Step 2)

Considering the results of the first step of the forecast model, the errors will be analyzed. Figures 11–13 show the error distributions for the three forecast horizons. It is possible to notice that the error distribution is similar to the Gaussian distribution, especially for the 1-h and 12-h forecasts. Moreover, the figures also present the mean and standard deviation of these errors. The fact that the means are not centered at zero suggests that there is still some error bias and that more information could be retrieved from the data. Additionally, the standard deviation increases for longer forecast horizons which, again, can indicate the higher degree of uncertainty on long-term predictions.

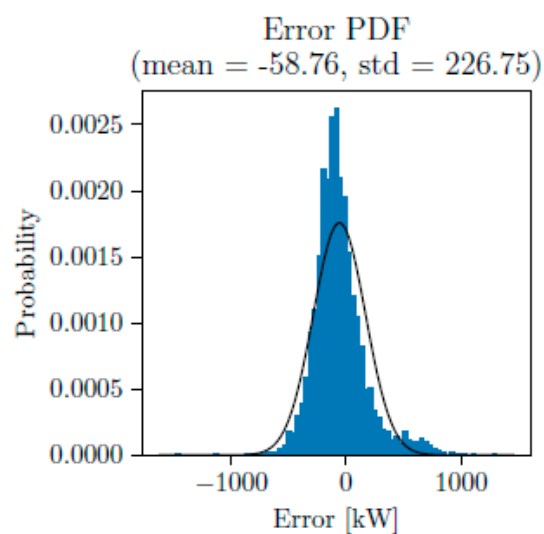


Figure 11. Ten-minutes-ahead error distribution.

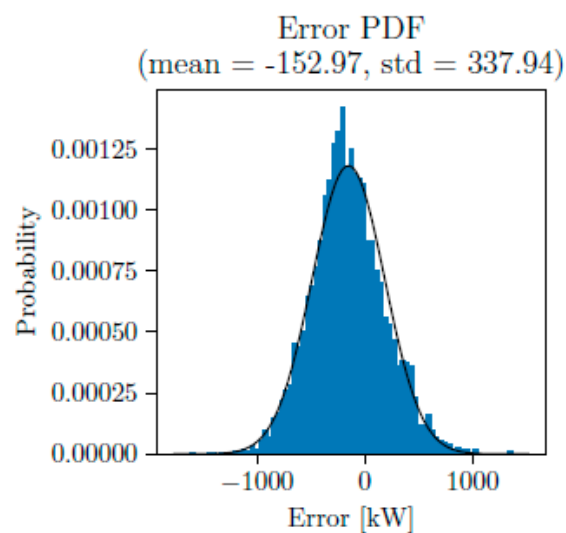


Figure 12. One-hour-ahead error distribution.

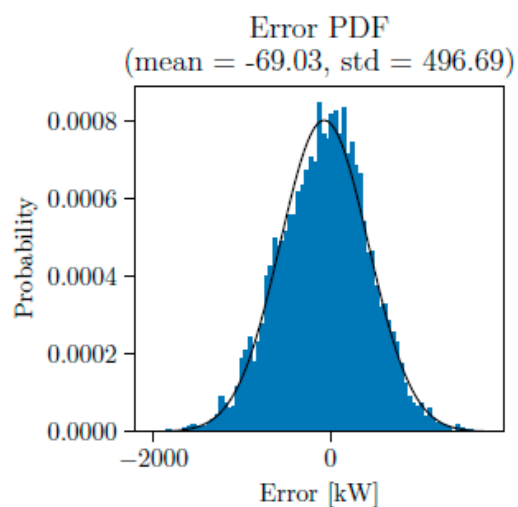


Figure 13. Twelve-hours-ahead initial forecasts.

Figures 14–16 show the ACF plot for the three horizons. As explained in Section 3.1, this graph reveals the strength of the relationship between a time series and its previous values and it was used to construct the error forecasting models. One can verify that the autocorrelation coefficients rapidly decreased and showed an oscillating behavior with more significant spikes at the end of each day (lags 144, 288, 432 and 576). Therefore, configurations combining these lags and the most recent values were tested to forecast the initial models' errors.

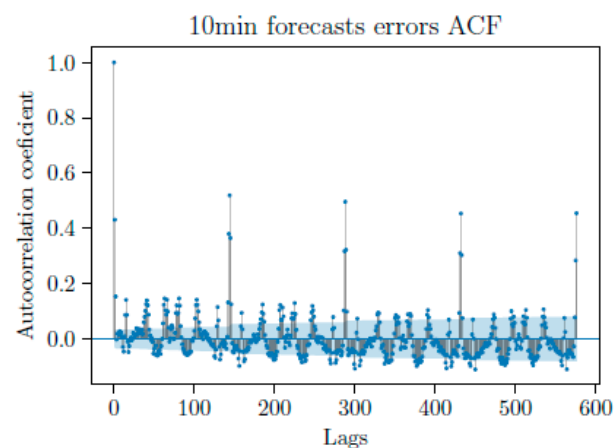


Figure 14. Ten-minutes-ahead error autocorrelation.

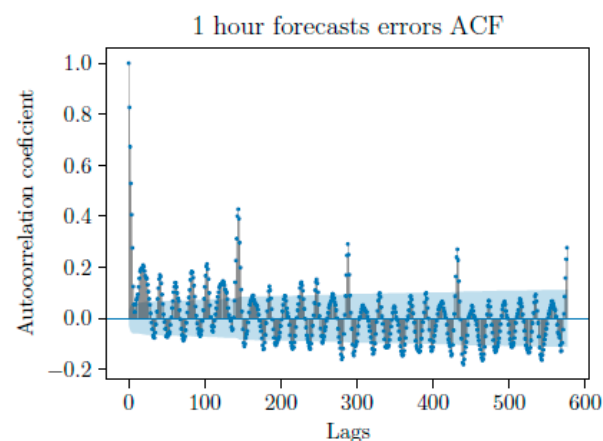


Figure 15. One-hour-ahead error autocorrelation.

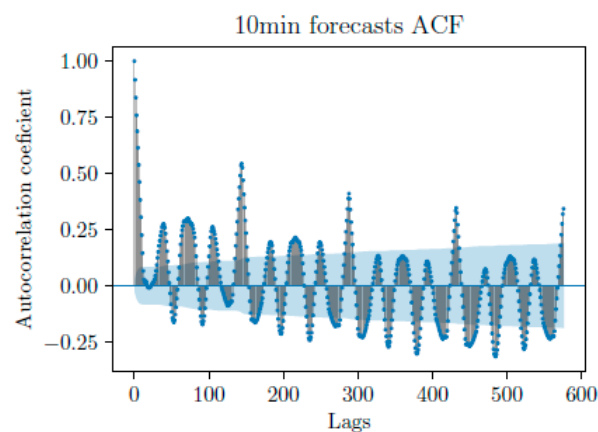


Figure 16. Twelve-hours-ahead error autocorrelation.

The error forecasts are presented below separately for each time horizon. The tables in this subsection show only the results of the best model for each horizon in order to make the text more fluid.

3.4.1. Ten-Minute Horizon

Table 7 shows the results of the best 10-min-ahead error forecasting model. In this scenario, all the models had similar results in terms of error metrics. The major difference was regarding the training time, which was higher for smaller LRs. The most probable reason for these facts is that the three models were able to find the same loss function minimum, but those with smaller LRs had smaller weight updates and, therefore, took longer. Figure 17 shows the error predictions made with model II. One can notice that the predictions were very accurate, especially on the error spikes. After this step, it was expected that the error correction model would improve the initial results.

Table 7. Results of the 10-min-ahead error forecasting models.

Model	LR	RMSE (kW)	MAE (kW)	Test RMSE (kW)	Test MAE (kW)	ET (s)
I	0.005	170.14	118.07	-	-	23.57
II	0.0025	169.97	118.26	103.69	78.01	50.90
III	0.001	172.17	119.87	-	-	96.70

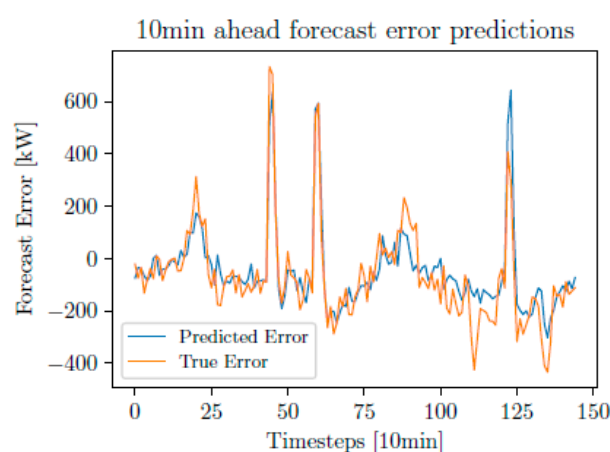


Figure 17. Ten-minutes-ahead error forecasts.

3.4.2. One-Hour Horizon

In Table 8, the results of the 1-h-ahead error forecasting models and the results of the best 1-h-ahead error forecasting model can be seen. The remarks made for the 10-min-ahead error forecasting model are also valid for this case. In Figure 18, the forecasts made with model II are illustrated. Here, the model was not as accurate as in the previous case to predict spikes and trends.

Table 8. Results of the 1-h-ahead error forecasting models.

Model	LR	RMSE (kW)	MAE (kW)	Test RMSE (kW)	Test MAE (kW)	ET (s)
I	0.005	284.81	230.45	-	-	16.66
II	0.0025	284.43	230.86	236.54	186.57	37.54
III	0.001	284.71	230.93	-	-	78.75

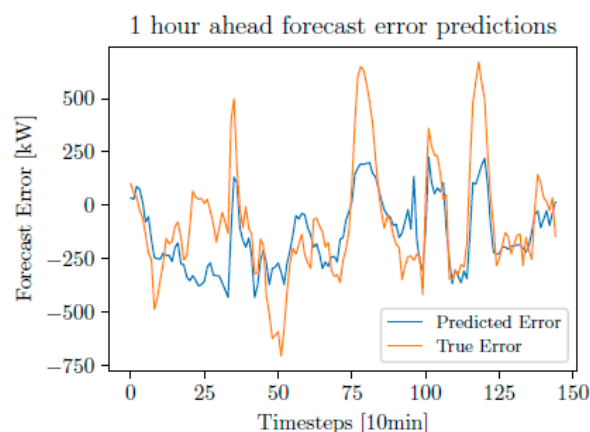


Figure 18. One-hour-ahead error forecasts.

3.4.3. Twelve-Hour Horizon

Table 9 presents the results of the best 12-h-ahead error forecasting model. In comparison to the other two scenarios, here there is a significant difference in the training time of the models. Although the results were similar, model III (with the smaller LR) was able to perform slightly better than the other two with a considerably higher execution time. This was probably because the smaller LR enabled the model to continue improving despite the closeness to a loss function minimum, while the other two stagnated and stopped training. Figure 19 shows that, as in the 1-h horizon, the model underestimated some spikes and, also, presented smoother behavior when compared to the true error.

Table 9. Results of the 12-h-ahead error forecasting models.

Model	LR	RMSE (kW)	MAE (kW)	Test RMSE (kW)	Test MAE (kW)	ET (s)
I	0.005	406.92	334.34	-	-	6.83
II	0.0025	407.37	334.82	-	-	9.54
III	0.001	405.60	332.62	34.053	257.43	56.57

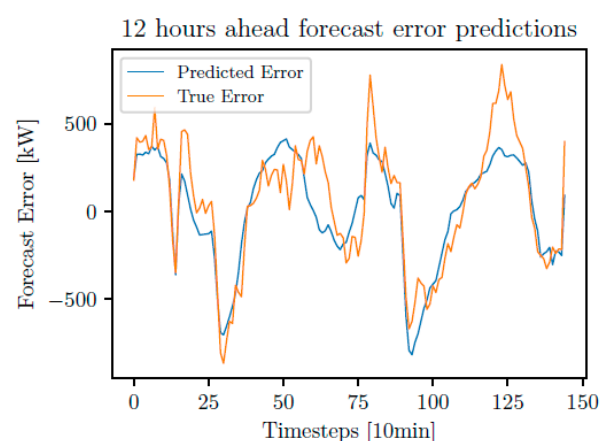


Figure 19. Twelve-hours-ahead error forecasts.

3.4.4. Summary of Errors Forecast Results in Step 2

Unlike the initial models (Step 1), here all the models in each forecast horizon presented similar performances regarding error metrics and execution times. In every case, the best model was the one that combined both the previous four periods and days as inputs. Lastly, Table 10 summarizes the results of the proposed error models for the three forecast horizons. Again, the forecast accuracy decreased as the forecast horizon increased, which

emphasizes the greater degree of difficulty to make longer forecasts due to the higher uncertainty of the data. Furthermore, one can notice that both the RMSE and MAE were lower than the ones achieved with the initial models, which can possibly indicate that these models were more accurate than the initial models.

Table 10. Summary of the adjusted models for the three horizons.

Horizon	RMSE (kW)	MAE (kW)
10-min	103.69	78.01
1-h	236.54	186.57
12-h	343.05	257.43

3.5. Load Forecast—Final Forecast (Step 3)

The adjusted forecast results, considering the initial load forecast (Step 1) and the errors forecast (Step 2), are presented below separately for each time horizon. As explained in Section 4, the adjusted model was obtained by adding the forecasted error to the initial prediction and had its results compared to those of the baseline models considering only the error test set.

3.5.1. Ten-Minute Horizon

Table 11 shows a comparison between the results obtained with the initial, adjusted and baseline forecasting models for the 10-min-ahead scenario. It can be noticed that the initial model performed worse than the persistence model and that, by means of the proposed methodology, it was able to outperform it. The improvements of the adjusted forecasts over the baseline were of 38.2% on the RMSE and 29.1% on the MAE. These results are illustrated in Figure 20, which shows that the proposed methodology apparently allowed the correction of the “persistence” (Baseline) behavior previously presented by the initial model and achieved more accurate forecasts.

Table 11. Results of the 10-min-ahead forecasts.

Steps	RMSE (kW)	MAE (kW)	MAPE (%)
Initial	200.83	152.96	3.44
Adjusted	103.69	78.01	1.75
Baseline	167.77	110.00	2.40

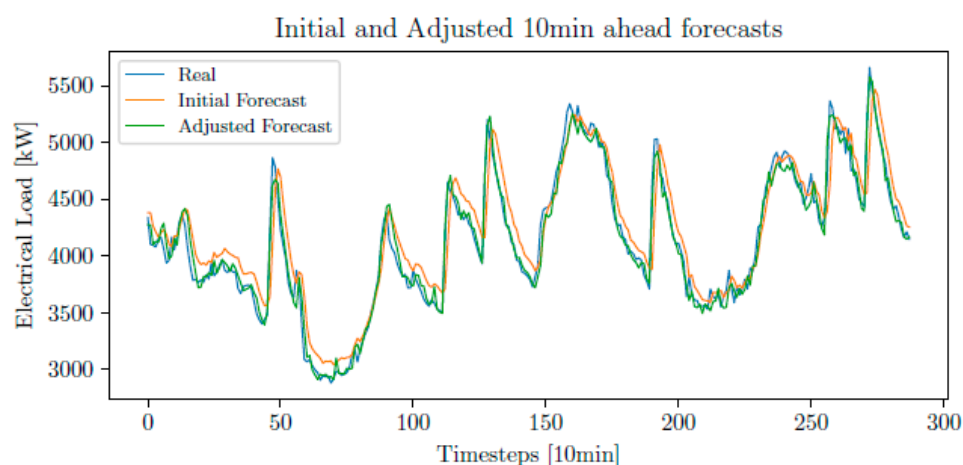


Figure 20. Ten-minutes-ahead adjusted forecasts.

3.5.2. One-Hour Horizon

In Table 12, a comparison between the results obtained with the initial, adjusted and baseline forecasting models for the 1-h-ahead scenario can be seen. Originally, the initial

model had a performance only slightly superior to the one of the linear regression model. With the proposed methodology, the initial results were improved by 32.1% on the RMSE and 36.2% on the MAE. In Figure 21, it is possible to notice that the model correctly adjusted the initial forecasts between timesteps 0 and 50 and close to timestep 150, while it did not change the accurate predictions of some load spikes.

Table 12. Results of the 1-h-ahead forecasts.

Steps	RMSE (kW)	MAE (kW)	MAPE (%)
Initial	352.95	295.43	6.95
Adjusted	239.34	188.49	4.30
Baseline	391.58	307.11	6.87

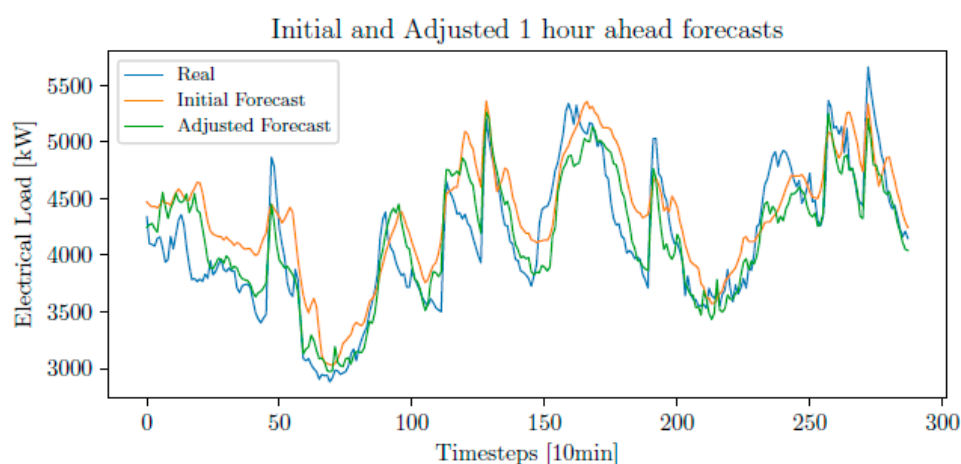


Figure 21. One-hour-ahead adjusted forecasts.

3.5.3. Twelve-Hour Horizon

Table 13 presents a comparison between the results obtained with the initial, adjusted and baseline forecasting models for the 12-h-ahead scenario. Like in the previous case, the proposed methodology was able to enhance the initial forecasts, which were only slightly better than that of the linear regression model. Here, the improvements over the initial model were of 22.5% on the RMSE and 29.0% on the MAE. Figure 22 shows the results of the initial and adjusted forecasts. It can be seen that, again, the proposed methodology properly corrected the forecasts close to the timesteps 50 and 150, while it did not affect the initial predictions on the load spikes.

Table 13. Results of the 12-h-ahead forecasts.

Steps	RMSE (kW)	MAE (kW)	MAPE (%)
Initial	442.55	362.55	8.35
Adjusted	343.05	257.43	5.79
Baseline	480.68	381.89	9.00

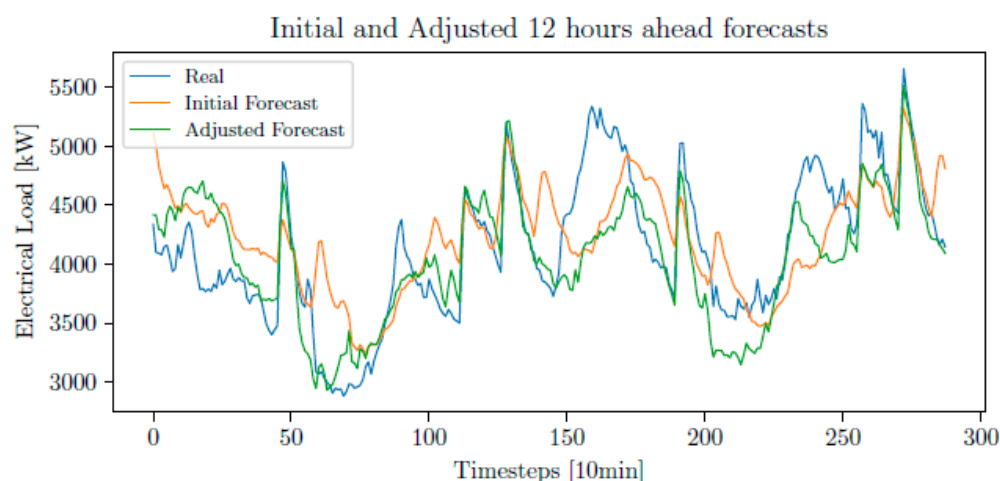


Figure 22. Twelve-hours-ahead adjusted forecasts.

3.5.4. Summary of Load Forecast Results in Step 3

Finally, Table 14 summarizes the results of the proposed methodology for the three forecast horizons. Like in the initial and error predictions, the forecast accuracy decreased as the forecast horizon increased, which reinforces the greater degree of difficulty to make longer forecasts due to the higher uncertainty of the data.

Table 14. Summary of the adjusted models for the three horizons.

Horizon	RMSE (kW)	MAE (kW)
10-min	103.69	78.01
1-h	239.34	188.49
12-h	343.05	257.43

4. Conclusions

The proposed methodology utilized an FFNN for both initial and error forecasts as the error time series often contains useful information that can be used to improve initial estimations. After a careful state-of-the-art review and an extensive data preliminary analysis, the models were created. A search for the best model inputs and parameter configurations was also conducted.

With regard to the electrical load demand forecasts, the data referred to the measured load demand in an industrial area connected to the medium voltage grid and were sampled every 10 min. The forecasts were made for three time horizons: 10 min, 1 h and 12 h ahead. The results demonstrated that the proposed initial models outperformed the linear regression model for the last two horizons, while, for the first, the results were worse than those achieved with a simple persistence model. By comparing the results for the three time scales, it was verified that the forecast accuracy decreased for longer time horizons, which highlights the higher difficulty to make longer forecasts due to the higher uncertainty of the data. Additionally, for very short time scales, the findings suggest that a simpler model can provide better estimates.

The initial electrical load demand forecasting errors analysis showed that the model might not have extracted all the information from the input data as the error distribution was not centered at zero and there was a correlation between a given error and the previous ones. Indeed, the error forecasting models achieved good forecasting accuracy, with a lower RMSE and MAE than the initial models. Furthermore, by combining the predicted error with the initial forecasts, it was possible to improve the initial results significantly for all time scales. Especially for the 10-min-ahead forecasts, the application of the proposed methodology resulted in a more accurate model than the persistence model.

The proposed methodology, including two forecast steps, based on historical data and errors, achieved accurate results in all time horizons when compared with the baseline method. This is a good indicator of the applicability of the method in real scenarios when short-term forecasts are required. To prove the effectiveness of the method, application in larger datasets considering consumers with different profiles will be tested. Depending on the type of consumers, residential, industrial or service buildings, some adjustments can be included in the method, namely the use of more hidden layers in the FFNN. These hidden layers can increase the complexity of the method as well as the execution time. A balance between the effectiveness and efficiency of the method should be identified according to the specific applications.

Author Contributions: Conceptualization, E.M., T.P., V.G. and H.M.; methodology, E.M., T.P., V.G. and H.M.; software, E.M., T.P.; validation, T.P., V.G. and H.M.; investigation, E.M.; writing—original draft preparation, E.M.; writing—review and editing, T.P., V.G. and H.M.; supervision, T.P., V.G. and H.M.; funding acquisition, T.P. and H.M. All authors have read and agreed to the published version of the manuscript.

Funding: Tiago Pinto has received funding from FEDER Funds through the COMPETE program and from National Funds through FCT under projects CEECIND/01811/2017 and UIDB/00760/2020. Hugo Morais was supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Intergovernmental Panel on Climate Change. Climate Change 2014: Synthesis Report. Contribution of Working Groups I, II and III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change. 2014. Available online: <https://www.ipcc.ch/report/ar5/syr/> (accessed on 8 July 2021).
2. European Parliament. Directive 2009/28/EC of the European Parliament and of the Council of 23 April 2009 on the Promotion of the Use of Energy from Renewable Sources; European Parliament: Brussels, Belgium, 2009.
3. European Parliament and of the Council. Framework for Achieving Climate Neutrality and Amending Regulation (EU) 2018/1999 (European Climate Law). 2020. Available online: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:52020PC0080> (accessed on 8 July 2021).
4. Intergovernmental Panel on Climate Change. Climate Change 2014 Mitigation of Climate Change; Cambridge University Press: Cambridge, UK, 2014; ISBN 9781107415416.
5. International Energy Agency (IEA). Secure Energy Transitions in the Power Sector. 2021. Available online: <https://www.iea.org/reports/%0Asecure-energy-transitions-in-the-power-sector> (accessed on 21 July 2021).
6. International Energy Agency (IEA). Power Systems in Transition. 2020. Available online: <https://www.iea.org/reports/power-systems-in-transition> (accessed on 21 July 2021).
7. Debnath, K.B.; Mourshed, M. Forecasting methods in energy planning models. *Renew. Sustain. Energy Rev.* **2018**, *88*, 297–325. [CrossRef]
8. Raza, M.Q.; Khosravi, A. A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings. *Renew. Sustain. Energy Rev.* **2015**, *50*, 1352–1372. [CrossRef]
9. Ahmad, T.; Chen, H. A review on machine learning forecasting growth trends and their real-time applications in different energy systems. *Sustain. Cities Soc.* **2020**, *54*, 102010. [CrossRef]
10. Khan, A.R.; Mahmood, A.; Safdar, A.; Khan, Z.A.; Khan, N.A. Load forecasting, dynamic pricing and DSM in smart grid: A review. *Renew. Sustain. Energy Rev.* **2016**, *54*, 1311–1322. [CrossRef]
11. Hippert, H.S.; Pedreira, C.E.; Souza, R.C. Neural networks for short-term load forecasting: A review and evaluation. *IEEE Trans. Power Syst.* **2001**, *16*, 44–55. [CrossRef]
12. Solyali, D. A Comparative Analysis of Machine Learning Approaches for Short-/Long-Term Electricity Load Forecasting in Cyprus. *Sustainability* **2020**, *12*, 3612. [CrossRef]
13. Kuster, C.; Rezgui, Y.; Mourshed, M. Electrical load forecasting models: A critical systematic review. *Sustain. Cities Soc.* **2017**, *35*, 257–270. [CrossRef]

14. Eskandari, H.; Imani, M.; Moghaddam, M.P. Convolutional and recurrent neural network based model for short-term load forecasting. *Electr. Power Syst. Res.* **2021**, *195*, 107173. [[CrossRef](#)]
15. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press Book: Cambridge, MA, USA, 2016.
16. Ahmad, A.; Javaid, N.; Mateen, A.; Awais, M.; Khan, Z.A. Short-Term Load Forecasting in Smart Grids: An Intelligent Modular Approach. *Energies* **2019**, *12*, 164. [[CrossRef](#)]
17. Anzai, Y. *Pattern Recognition and Machine Learning*; Elsevier: Amsterdam, The Netherlands, 2012.
18. Rana, M.; Koprinska, I. Forecasting electricity load with advanced wavelet neural networks. *Neurocomputing* **2016**, *182*, 118–132. [[CrossRef](#)]
19. Ozer, I.; Efe, S.B.; Ozbay, H. A combined deep learning application for short term load forecasting. *Alexandria Eng. J.* **2021**, *60*, 3807–3818. [[CrossRef](#)]
20. Karthika, S.; Margaret, V.; Balaraman, K. Hybrid short term load forecasting using ARIMA-SVM. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017; pp. 1–7.
21. Orrù, P.F.; Zoccheddu, A.; Sassu, L.; Mattia, C.; Cozza, R.; Arena, S. Machine Learning Approach Using MLP and SVM Algorithms for the Fault Prediction of a Centrifugal Pump in the Oil and Gas Industry. *Sustainability* **2020**, *12*, 4776. [[CrossRef](#)]
22. Jiang, H.; Zhang, Y.; Muljadi, E.; Zhang, J.J.; Gao, D.W. A Short-Term and High-Resolution Distribution System Load Forecasting Approach Using Support Vector Regression With Hybrid Parameters Optimization. *IEEE Trans. Smart Grid* **2018**, *9*, 3341–3350. [[CrossRef](#)]
23. Silva, J.; Praça, I.; Pinto, T.; Vale, Z. Energy Consumption Forecasting Using Ensemble Learning Algorithms. In *Advances in Intelligent Systems and Computing*; Herrera-Viedma, E., Vale, Z., Nielsen, P., Martin Del Rey, A., Casado Vara, R., Eds.; Springer International Publishing: Cham, Switzerland, 2020; Volume 1004, pp. 5–13. ISBN 978-3-030-23945-9.
24. Jozi, A.; Pinto, T.; Praca, I.; Vale, Z. Day-ahead forecasting approach for energy consumption of an office building using support vector machines. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bangalore, India, 18–21 November 2018; pp. 1620–1625.
25. Kong, X.; Li, C.; Wang, C.; Zhang, Y.; Zhang, J. Short-term electrical load forecasting based on error correction using dynamic mode decomposition. *Appl. Energy* **2020**, *261*, 114368. [[CrossRef](#)]
26. Aurélien, G. *Hands-On Machine Learning With Scikit-Learn, Keras, and Tensorflow—Concepts, Tools, and Techniques to Build Intelligent Systems*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2017.
27. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.