Very Short-Term Load Forecaster Based on a Neural Network Technique for Smart Grid Control

Fermín Rodríguez [1,2,*], Fernando Martín [1,2], Luis Fontán [1,2] and Ainhoa Galarza [1,2]

[1] Ceit-Basque Research and Technology Alliance (BRTA), Manuel Lardizabal 15,
   20018 Donostia/San Sebastián, Spain; fmartin@ceit.es (F.M.); fontan@ceit.es (L.F.); agarlaza@ceit.es (A.G.)

[2] Universidad de Navarra, Tecnun, Manuel Lardizabal 13, Ceit, Manuel Lardizabal 15,
   20018 Donostia, San Sebastián, Spain

* Correspondence: frlalanne@ceit.es

Abstract: Electrical load forecasting plays a crucial role in the proper scheduling and operation of power systems. To ensure the stability of the electrical network, it is necessary to balance energy generation and demand. Hence, different very short-term load forecast technologies are being designed to improve the efficiency of current control strategies. This paper proposes a new forecaster based on artificial intelligence, specifically on a recurrent neural network topology, trained with a Levenberg–Marquardt learning algorithm. Moreover, a sensitivity analysis was performed for determining the optimal input vector, structure and the optimal database length. In this case, the developed tool provides information about the energy demand for the next 15 min. The accuracy of the forecaster was validated by analysing the typical error metrics of sample days from the training and validation databases. The deviation between actual and predicted demand was lower than 0.5% in 97% of the days analysed during the validation phase. Moreover, while the root mean square error was 0.07 MW, the mean absolute error was 0.05 MW. The results suggest that the forecaster's accuracy is considered sufficient for installation in smart grids or other power systems and for predicting future energy demand at the chosen sites.

Keywords: smart grid; energy demand; very short-term forecaster

1. Introduction

Predicting energy demand has a significant effect on the operation and schedule of any power system, no matter its dimensions. Whether the system is a microgrid or a traditional power network, forecasting load electricity demand is required if a reliable and efficient power system is desired. Many elements, such as energy price, unit commitment, load dispatch for generators and so on, depend on the value obtained through load forecasters [1].

Due to the deregulation of the energy market and new free competition policies, the number of stakeholders involved in the power system has unexpectedly increased. Moreover, traditional energy storage systems are not able to store the overproduction of energy produced in a specific moment if reversible hydropower plants are not taken into account. For this reason, it is necessary to balance power generation and energy demand curves so that frequency or voltage issues are avoided and power system stability is guaranteed. Therefore, big power system operators and control algorithms at microgrids need reliable and accurate forecasts for their demand response programs, keeping in mind that uncertainty will be reduced but not entirely removed [2].

Microgrids can be defined as a small-scale replication of the traditional power system, one in which distributed energy generators, renewable and/or non-renewable resources, storage technologies and loads are integrated. In addition, they can be connected to or islanded from the

main grid, depending on policies and the interests of the microgrid's owner and the power system operator. Microgrids can provide certain benefits, such as fewer distribution and transmission power loses, lower dependence on the volatile price of energy due to the intermittency of renewable resources, the use of renewable resources, and so on [3]. Because of these advantages, the number of microgrids has increased and they have been applied in different environments, such as support for traditional networks, university campuses, islanded regions and military command centres.

The development of microgrids makes it necessary to improve current control algorithms by introducing forecasters so the microgrids can operate in optimal conditions and maximise their efficiency. In particular, being able to obtain accurate energy demand forecasts is one of the key challenges that researchers are trying to overcome by using different techniques [4–7] to develop better control strategies. Nevertheless, predicting energy demand in microgrids is usually more complex than in conventional grids owing to the fact that the load time curves of a microgrid are much more volatile than those of a traditional power system [8]. Although the forecaster proposed in this paper has been developed for microgrids, it can also be used for bigger power systems due to their lower volatility.

A review of the available literature reveals that researchers are setting their forecasters for different prediction horizons depending on what information is needed to make a good decision. In terms of prediction horizons, forecasters are classified into four different categories:

- Very Short-Term Forecast (VSTF): The forecaster makes predictions for a few minutes ahead, and prognosticated values are given to the control unit for real-time dispatch. This forecasting is used for getting a quick response to the intra-day energy demand fluctuations [9,10].
- Short-Term Forecast (STF): The forecaster makes predictions from few hours to days ahead, and the results are used for a wide range of decisions related to unit commitment, economic dispatch and power system operation [11,12]. Currently, VSTF and STF have become more relevant due to the need for accurate forecasters, and thus in the last few years the area has been addressed by many studies on different forecasters.
- Medium-Term Forecast: In this case, the forecaster gives information from a few hours to a few weeks ahead. The obtained predictions provide information about weekly fluctuations and this information is mainly used for scheduling maintenance for the power system's stakeholders, such as power plants, transmission and distribution lines, transformers, and so on. [13,14].
- Long-Term Forecast: The forecaster provides predictions from few weeks to months and the given information is commonly used for power assessment or to analyse the necessity of new power lines [15,16].

This paper presents a very short-term load demand forecaster for implementation in a microgrid. Although several methods are proposed in the literature [9,10,17,18], this forecaster is based on an Artificial Neural Network (ANN), specifically on a layer recurrent neural network. The forecaster is able to predict the demand for energy that will be consumed by the loads 15 min ahead with sufficient accuracy. The predicted values obtained through the forecaster is given to the microgrid's control unit so it can take better decisions and improve the system's overall efficiency. Moreover, due to the need to have a high standard of accuracy in an energy demand forecaster that will be implemented in a microgrid, the forecaster can also be used in bigger power electric systems owing to the lower volatility of such systems, providing information to the power system operator. It must be taken into account that the bigger a power system is, the lower the energy demand volatility is due to the large number of loads.

An iterative process was used to design the final structure of the forecaster, and this paper describes the most relevant steps and the rationale for those steps. The key contributions of this paper can be summarised as follows:

1. A VSTF (15 min ahead) for load demand forecasting is presented for implementation in a microgrid. The forecaster is based on a layer recurrent neural network which takes into account the following input parameters to predict the future energy demand: time, season and the energy demand for the previous 24 h.

2. The calculated error metrics used to analyse the difference between predicted and consumed energy demonstrate that the forecaster has sufficient accuracy. If the accuracy of the forecaster is compared against the available literature, the developed forecaster provides a slight improvement. Moreover, the architecture and steps taken in developing this tool are easier than for forecasters presented in the literature.
3. As research in the field of energy demand forecasting has only focused on improving methods, the analysis of determining the optimal input vector and the optimal database length has been ignored for the most part. In this paper, a sensitivity analysis was performed, examining different forecasters by changing these parameters to obtain the optimal ones.

2. Related Works

The demand prediction models fall into two main categories: parametric and non-parametric methods. While parametric methods are based on analytical models, non-parametric methods rely on artificial intelligence. Although many demand forecasters have been proposed in the last few years, in what follows, some of the most used models that researchers are currently basing their work on are described [17–24].

2.1. Parametric (Analytical) Methods

A parametric method is defined as a group of probability distributions (P_θ), where each distribution in the group is characterised by a finite number of parameters (θ). These models work by analysing the information in a given database in order to fix the coefficients of the parameters (θ) in the method. Once the model's parameters are fixed, the forecaster is able to predict future values. The principal disadvantage of these models is their low capacity for making accurate predictions in non-linear models when sudden changes related to environmental or social incidents occur. Below is a brief description of two popular models for very short-term load forecasting.

2.1.1. Polynomial Regression Model

These models, also known as polynomial evaluations, are applied to linear models in order to develop tools where databases and actual data are needed to make predictions. The polynomial evaluation model is described as

$$y = p_1 x^n + p_2 x^{n-1} + \cdots + p_n x + p_{n+1} \tag{1}$$

where y is the value forecasted by the model, p represents the polynomial coefficients whose values are defined through the database, x is the currently measured data, and n is the degree of the obtained polynomic function. The coefficients and the degree are obtained by solving a linear equation model. For instance, in [17], the authors used cubic and quadratic fitting, arguing that as the degree increases, the extrapolated points become more erratic.

2.1.2. Time Series Models

Time series techniques have been widely applied to predict energy demand in different horizons. The most popular are the autoregressive moving average (ARMA) and autoregressive integrated moving average (ARIMA) models, which are used to develop new forecasters as well as like a benchmark to compare the results obtained through other parametric or non-parametric methods.

The ARMA model combines autoregressive (AR) and moving average (MA) models, where the AR model relies on the linear regression of actual values based on a database and the MA model makes regressions based on the forecast errors of past values [18]. The ARMA model was developed for the specific condition in which the time series is stationary, i.e., all vectors of the model have the same probability distribution function for any time. To deal with this challenge, the ARIMA model was developed [19,20].

2.2. Non-Parametric (Artificial Intelligence) Methods

Artificial Intelligence (AI) methods have been developed in the last decade as an alternative to analytical ones. Although AI methods are able to determine complex nonlinear relationships, their main drawback is their complex development. Some of the models used for load forecasting are described below.

2.2.1. Artificial Neural Networks (ANNs)

ANNs rely on a connection of elementary units called artificial neurons, which try to emulate the way the human brain works. The connection between neurons is known as the synaptic weight and the strength of the connection is established through a training step. For this step, a historical database is needed, one that is characterised by two pieces of data. While historical observations are the input variables to the ANN, output parameters are related to predicted values. The optimisation of the synaptic connection weight is usually done by minimising a quadratic function output error [21]. According to Kuster et al. [18], ANNs are one of the most popular non-parametric methods for developing new energy demand forecasters, and many tools are based on this method.

2.2.2. Support Vector Machines (SVMs)

Like ANNs, Support Vector Machines (SVMs) are used to solve different non-linear regression situations, such as planning, learning, classification or perception. This method is based on the kernel model, which is an optimisation technique for determining the vectors of the forecaster. The kernel method relies on extracting a small piece of data from the database and using it to solve a quadratic programming model to establish the vectors [22]. Moreover, to decrease the generalisation errors, this method applies structural risk minimisation [23]. Therefore, the goal of SVMs is to generate an optimal separating hyperplane (vectors) where future observations can be classified to generate forecasts.

2.2.3. Regression Tree (RT)

Regression Trees (RTs) are a new version of the traditional decision trees. While decision trees were designed as a classification method, regression trees have been developed to forecast just a single value through input parameters. The RT model uses the Classification and Regression Trees algorithm to develop forecasters. This algorithm consists of a recursive splitting of the data into different branches and all options (or branches) at each iteration are studied. The algorithm chooses the branch that minimises the mean squared error (MSE) between the two separate branches. The splitting continues until the minimum node size established by the user is obtained. If a node obtains a zero MSE value, it will be considered to be a final node even though it has not obtained the condition of minimum size [24].

The literature therefore suggests that energy forecasters should not be based on analytical techniques due to the presence of non-linear phenomena and their decreased ability to make accurate forecasts under unexpected changes in energy demand [17]. For these reasons, AI was chosen to develop the tool presented in this study.

In the last few years, several energy demand forecasters have been developed for the different time horizons described in Section 1. The available literature provides evidence that, as the time horizon increases, the structure of the forecaster needs to be more complex if accurate predictions are requested [9,14,16]. To solve this reduction in the accuracy of the forecaster when the forecast horizon time increases, researchers have proposed hybridisation, which combines different techniques in order to take advantage of each method and thus improve the accuracy of the forecasters. However, despite the advantages of hybridisation, for the purposes of this study's aim to develop a forecaster for VSTF and given the research that shows that accurate predictions can be made with single methods [8,25,26], hybridisation was discarded.

Because a single method was going to be used to develop our forecaster, Table 1 presents the advantages and disadvantages of the non-parametric techniques presented in this section.

Table 1. Advantages and disadvantages of described non-parametric methods.

Technique	Advantage	Disadvantage
ANN	• Lower training time • Ability to establish complex relationships between parameters • Different choices for training algorithm • Establish interactions between forecast parameters	• Risk of overfitting • Black box nature, trust the obtained net • There is no easy way to fix the optimal network structure. An iterative process must be run • Does not perform well with little data
SVM	• Overfitting can be controlled • It is defined as a convex problem, so there are many ways to solve it	• It is not easy to properly choose the kernel function for each problem • There is no procedure for establishing the optimum speed and size of training and validation steps. An iterative process must be performed • The final model is difficult to understand, so it is not easy to make small modifications to the model provided by the technique
RT	• Simple to understand and interpret, so calibrations can be made • Large amounts of data are not needed to develop them	• Unstable, small change in data will affect the final result • If there are unknown values, calculations become complex

An analysis of the single methods in Table 1 led us to conclude that, for the purposes of this study, the best option for developing the forecaster was the ANN technique. Moreover, as Kuster et al. note in [18], ANNs are one of the usual choices for implementing and developing new forecasters. Although ANNs have some drawbacks, such as running the iterative process until the optimal structure is found or the risk of overfitting, a solution is proposed in this paper for each of the disadvantages. The black box nature of the ANN is the only disadvantage that cannot be avoided.

With regard to the fixing of an optimal structure, a methodology that has addressed this trouble was developed in [27] with accurate results. This procedure was used to select the final structure for our forecaster. To avoid the overfitting risk, the training time versus forecaster accuracy was analysed to determine the optimal training time.

2.3. Artificial Neural Networks (Description and Classification)

2.3.1. Description

Traditional computer processing and the human brain do not work in the same way. While the human brain uses abilities such as nonlinearity and parallelisation to process information and make decisions, the best conventional software programs try to emulate human brain capacity through analytical techniques, although their ability to do so is quite limited. There are certain areas, such as planning, classifying and perception, where the human brain's ability still outperforms computational techniques [28]. To deal with this challenge, AI, and the ANN in particular, has been proposed as a potential solution for improving computing skills. ANNs are one of several solutions that are based on replicating the biological neural system of the human brain to achieve the desired improvement [29].

ANNs are based on linking single units, namely, artificial neurons, which imitate the information processing of the biological neurons of the human brain. While biological neurons are connected to each other by synaptic links that are built through the experiences of life, artificial neurons are also linked, and the strength of those links is established by the learning, or training, step. For this learning step, it is essential to have an accurate historical database, due to the fact that the links between the neurons of the net are going to be established through the information in the database.

With regard to the values forecasted by ANNs, a generalisation is defined as the ability of the forecaster to generate accurate outputs from data that have not been used in the learning step [28]. A generalisation can be just obtained by a proper training step; this allows the forecasters to generate accurate predictions. In terms of the learning step, different algorithms that basically rely on the calibration of the synaptic weight connections between neurons until the optimal network is obtained [28,30] can be used for this purpose. After analysing the related literature and the different choices for training algorithms, the Levenberg–Marquardt (LM) algorithm was chosen to develop the forecaster presented here. In several research studies [3,8,27], the LM algorithm has demonstrated its robustness and the forecasted values have been shown to be sufficiently accurate.

2.3.2. Classification of ANNs

Artificial neural networks are traditionally divided into two main categories, depending on the flow of the data through the network [28]: feedforward or recurrent neural networks. While, in feedforward networks, the information is processed in a linear direction from input parameters to forecasted values, recurrent networks add feedback loops where the output information obtained by the neurons passes through a time delay and is used for future forecasts. Figure 1 presents the layout of these principal neural networks. Although both ANN architectures have a single layer (output layer), there is the option of increasing the number of hidden layers. Thus, when an ANN forecaster is being developed, certain parameters need to be fixed: input/output variables, number of neurons, number of hidden layers, and the delay parameter if a recurrent network is used.

The main advantage of recurrent versus feedforward neural networks comes from the learning capability of the recurrent network, where it is able to establish more complicated relationships between the parameters involved [28]. Therefore, the ANN is able to generate more accurate results in a time series forecast.
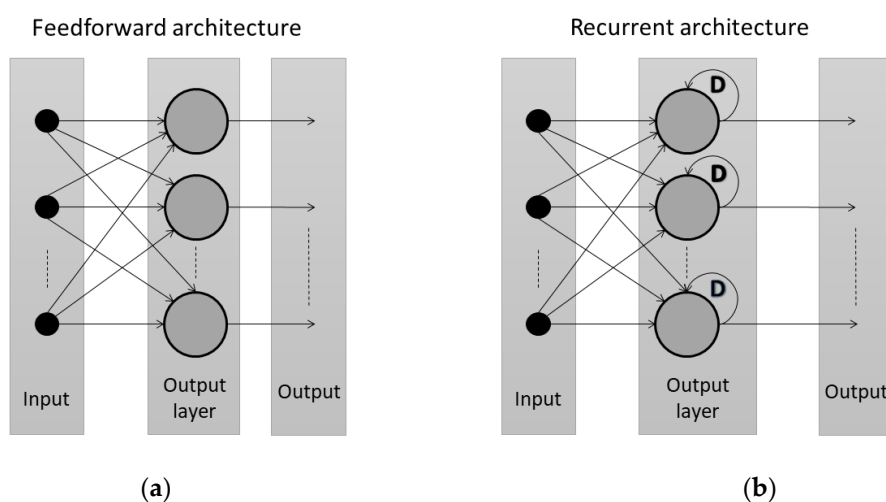

Figure 1. (**a**) Feedforward and (**b**) recurrent architecture layouts.

Following an analysis of different recurrent neural network (RNN) structures, such as the Elman [31], simultaneous [32], nonlinear autoregressive exogenous (NARX) [33] or layer [27] networks, the layer recurrent neural network was used to develop our forecaster. In this case, a three-layer architecture was chosen: an input layer to introduce the data, a hidden layer to process the data and an output data to make the forecast. To set the structure (the delay and the number of neurones of the hidden layer) of the forecaster, an iterative process was applied to fix parameters such as the delay or number of neurons in the hidden layers. After the final architecture was set and the training step was completed, the validation step, which used data for a different year than the years used for the training step, was done to determine the forecaster's real accuracy.

2.3.3. Input Parameter Selection

Before fixing the forecaster's parameters, it is necessary to guarantee that the input parameters used in the forecaster have a strong relationship with the predicted parameters. Therefore, before using any parameter as an input to the forecaster, the strength of the relationship between an input variable and the forecaster's output was tested. The goal of this test was to eliminate the parameters that do not really characterise the procedure.

In order to choose input variables that have a strong relationship with the outputs of the forecaster, some correlation analyses were performed. These analyses help to quantify the strength of the relationship between two variables. For this study, Pearson's correlation test was selected. Pearson coefficients can be positive, negative or null, and the value must be included within the range from −1 to +1. Moreover, as Figure 2 shows, the relationship between the involved parameters can be weak or strong.
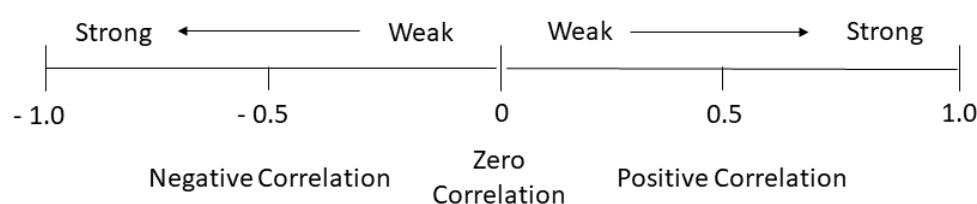

Figure 2. Pearson correlation scale.

2.3.4. The Importance of the Historical Database

When an ANN is developed, choosing an appropriate database is a key step. If the proper variables are not selected or the database contains improperly recorded measurements, the accuracy of the forecaster will be affected. Moreover, the distance between the place where the database is recorded and the site where the forecasts will be done must be taken into account; the shorter the distance, the greater the accuracy. The information used in the learning step of this forecaster was obtained through two different organisations: the load demand database contains data measured at a university campus, which includes a hospital, and the meteorological data were obtained from the meteorological agency of the government of Navarre.

After the input variables are selected, the next step consists of generating the databases of those variables. Before the learning step is started, input and output databases have to be examined for outliers. If abnormal values are found within the database, there are different options for dealing with them, depending on the extension of the affected data. If there are few abnormal values, a linear interpolation can be used. However, if the affected extension is larger, a similar day or an exponential smoothing equation could be used [34]. Moreover, to ensure a quicker learning step and guarantee accurate forecasts, each parameters of the database needs to be normalised. Therefore, each variable within the database will be scaled between zero and one. Because input variables are normalised, the forecast figure will be normalised, and thus it is necessary to de-normalise the predicted value before it is compared with the actual figure.

With regard to the extension of the meteorological and load energy demand databases, there is no agreement on what the extension of these databases should be. Moreover, although some authors argue that weather variables can be neglected because they have little influence on the energy demand on VSTF [10,35], no study was found to support this statement. Therefore, different forecasters were developed in this study by including or not including the meteorological parameters and by varying the length of the databases to obtain the optimal forecaster and to compare it with the state of the art.

2.4. Error Metrics

In both the learning step and in the validation step, it is necessary to examine the accuracy of the different proposed forecasters in order to select the best choice. Moreover, it is also important to compare the results obtained through the developed forecaster with those that are proposed by other authors in the literature. Hence, we used mean absolute error (MAE), mean absolute percentage error (MAPE) and root mean squared error (RMSE) throughout the different steps of this study. The following Equations (2)–(4) show how these parameters were calculated:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - Y_i')^2} \tag{2}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}(Y_i - Y_i') \tag{3}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{(Y_i - Y_i')}{Y_i}\right| \times 100\% \tag{4}$$

where Y_i is the measured value in a moment; Y_i' is the predicted value in a moment, and N are the number of predictions that are analysed. With regard to the RMSE parameter, it must be taken into account that the error is squared, which means that a few inaccurate forecasts strongly distort the overall metric. To avoid these situations, error histograms and normal distribution graphs are presented to show the obtained real error dispersion.

3. Forecaster Development and Parameter Selection

This section presents the steps taken to develop our energy demand forecaster. In order to fix the RNN-related parameters, such as input variables, length of the database and number of neurons, a methodology developed for a similar solar irradiation forecaster [27] based on an RNN was used. However, some steps were changed, and the final architecture of the forecaster was different.

3.1. Input Parameter Selection

As explained in Section 2.3.3, it was necessary to select the variables that really characterise the process as input parameters. Firstly, because the parameter that we wanted to forecast was energy demand, past values of this parameter were used as inputs in the forecaster. In this case, the energy demand values for the previous 24 h, recorded in intervals of 15 min, were used. Secondly, from among the different meteorological variables, we analysed whether temperature could be a possible parameter that characterises the process. To examine whether there is a true relationship between both parameters, temperature and energy demand, a Pearson correlation test was performed. Figure 3 shows the results obtained through this test.
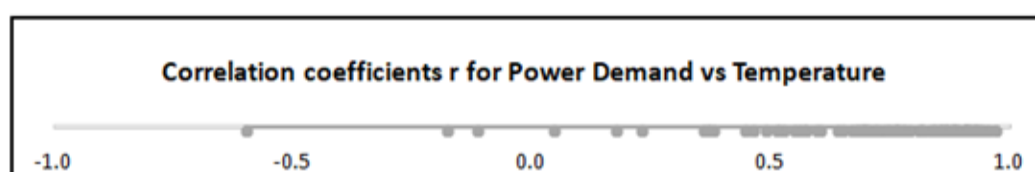

Figure 3. Pearson correlation tests for power demand vs. temperature.

Figure 3 shows the correlation analysis between power demanded by a building and the temperature measured in a meteorological station near this building. Each of the dots that are represented in the graph of Figure 3 is related to the Pearson correlation coefficient for a sample day, where the relationship between power demand and temperature was analysed. Temperature and

power demand measurements for these sample days were taken from the different seasons of a two-year database covering 2016–2017.

As explained above, some authors [10,35] argue that meteorological variables such as temperature can be avoided in VSTF of power demand. However, from Figure 3, it can be concluded that there is a strong relationship between energy demand and temperature. Therefore, the temperature parameter was used in some forecasters as an input in order to analyse whether it would be a relevant input parameter in the final forecaster.

3.1.1. Power Demand Evolution

A building's power demand is strongly related to the activities carried out inside it and the weather conditions [36]. Although the power curve for building's energy demand is more or less the same throughout a year, it is also affected by other phenomena such as the season and time of day [37]. Therefore, it was concluded that season and time in 15-min intervals should be analysed as possible input parameters.

3.1.2. Proposed Forecasters

As there are different combinations of the selected input parameters, and taking into account that there is no agreement on the length of the database for the training step, the forecasters purposed in Table 2 were tested to determine which was most accurate. Table 2 shows the parameters that were selected for each forecaster and the length of the database used.

Table 2. Proposed options for the final energy forecaster.

		Option 1	Option 2	Option 3	Option 4	Option 5	Option 6
Chosen Input Parameters	Power demand	x	x	x	x	x	x
	Season			x	x	x	x
	Hour			x	x	x	x
	Temperature					x	x
Length database	1 year	x		x		x	
	2 years		x		x		x

3.2. Selecting the Optimal Forecaster

After the input parameters were combined, six possible forecasters were proposed. For options 1 and 2, 96 values were used in the training input array, X. These 96 values were related to the power demand for the building under study, for the previous 24 h and in 15-min intervals. For options 3 and 4, 98 values were used in the input array: the season, the hour of the day at which the forecast was to be made, and the 96 values of the power demand. Finally, for options 5 and 6, the input array had 194 values: the season, the hour, the 96 values related to the power demand and the 96 figures related to the temperature. All forecasters had a single target output, T—power demand.

Moreover, to select the optimal structure of each forecaster, data from January 2016 to April 2016 and January 2017 to April 2017 were used as an approximation to reduce each forecaster's training computation time, taking into account that the longer the database, the longer the forecasters' training time. Once the forecasters' optimal structures were defined, the databases for the whole of 2016 and 2017 were applied to develop final forecasters. For the options in Table 2 that indicate that the length of the applied database was a single year, the structure was optimised with data from 2017. However, for the options in Table 2 that indicate a two-year length database was used, the data were from 2016 and 2017. The optimal structure for each forecaster was selected by examining the RMSE of the forecasts.

Because an RNN had been chosen, it was necessary to first select delay and then the number of neurons. To select the delay, the delay parameter was modified, while the number of neurons was kept at the same value. When the delay had been fixed, it was held constant while the number of neurons parameter was modified.

Two ANNs designed with the same structure and trained on the same number of epochs will not yield the same forecasted value. This happens because of the black box nature of ANNs. Therefore, if the desire is to ensure that the selected architecture is the best one, it is necessary to run the possible structures a certain number of times and average the forecasted values. In our case, the process described in [27] was used to calculate the "RMSE Training Data" and "RMSE Validation Data" values shown in Tables 3–8. While the "RMSE Training Data" are obtained by averaging the RMSE values obtained from the five repeated tests for the same structure for the January 2016 forecast, the "RMSE Validation Data" are obtained by averaging the forecast for January 2018.

Table 3. Structure design test for option 1.

Parameter	Test	Neurons	Delay	RMSE Training Data (MW)	RMSE Validation Data (MW)
Delay	1	10	1:2	0.0840	0.0806
	2	10	1:3	0.0818	0.0790
	3	10	1:4	**0.0792**	**0.0781**
	4	10	1:5	0.1007	0.0982
	5	10	1:6	0.1067	0.1022
Number of Neurons	6	5	1:4	0.0896	0.0859
	7	15	1:4	0.1152	0.1117
	8	20	1:4	0.1338	0.1317

Bold: the best structure.

Table 4. Structure design test for option 2.

Parameter	Test	Neurons	Delay	RMSE Training Data (MW)	RMSE Validation Data (MW)
Delay	1	10	1:2	0.0814	0.0799
	2	10	1:3	0.0889	0.0853
	3	10	1:4	**0.0807**	**0.0790**
	4	10	1:5	0.0853	0.0821
	5	10	1:6	0.0942	0.0923
Number of Neurons	6	5	1:4	0.0941	0.0932
	7	15	1:4	0.1063	0.1005
	8	20	1:4	0.1184	0.1170

Bold: the best structure.

Table 5. Structure design test for option 3.

Parameter	Test	Neurons	Delay	RMSE Training Data (MW)	RMSE Validation Data (MW)
Delay	1	10	1:2	0.0924	0.0931
	2	10	1:3	**0.0803**	**0.0783**
	3	10	1:4	0.0857	0.0843
	4	10	1:5	0.0830	0.0815
	5	10	1:6	0.0943	0.0922
Number of Neurons	6	5	1:3	0.0913	0.0904
	7	15	1:3	0.0858	0.0845
	8	20	1:3	0.0868	0.0860

Bold: the best structure.

Table 6. Structure design test for option 4.

Parameter	Test	Neurons	Delay	RMSE Training Data (MW)	RMSE Validation Data (MW)
Delay	1	10	1:2	0.0800	0.0776
	2	10	1:3	0.0825	0.0781
	3	10	1:4	0.0876	0.0851

4	10	1:5	0.0806	0.0784
5	10	1:6	0.0856	0.0824
Number of Neurons	6	5	1:2	0.0816	0.0779
	7	15	1:2	**0.0755**	**0.0737**
	8	20	1:2	0.0832	0.0796

Bold: the best structure.

Table 7. Structure design test for option 5.

Parameter	Test	Neurons	Delay	RMSE Training Data (MW)	RMSE Validation Data (MW)
Delay	1	10	1:2	0.0769	0.0764
	2	10	1:3	0.0792	0.0784
	3	10	1:4	**0.0749**	**0.0754**
	4	10	1:5	0.0814	0.0792
	5	10	1:6	0.0874	0.0854
Number of Neurons	6	5	1:4	0.0811	0.0810
	7	15	1:4	0.0794	0.0792
	8	20	1:4	0.0883	0.0895

Bold: the best structure.

Table 8. Structure design test for option 6.

Parameter	Test	Neurons	Delay	RMSE Training Data (MW)	RMSE Validation Data (MW)
Delay	1	10	1:2	0.0819	0.0775
	2	10	1:3	0.0841	0.0828
	3	10	1:4	0.0791	0.0775
	4	10	1:5	0.0824	0.0797
	5	10	1:6	0.0824	0.0797
Number of Neurons	6	5	1:4	**0.0754**	**0.0726**
	7	15	1:4	0.0779	0.0768
	8	20	1:4	0.0810	0.0779

Bold: the best structure.

To summarize, Table 9 contains the optimal structures for each option described in Table 2 after each forecaster's sensitivity analysis was done (see Tables 3–8). In addition, there are other relevant parameters such as the number of outputs or layers that have been chosen for each forecaster without performing a sensitivity analysis. Thus, all developed forecasters have a single output that will provide the demanded energy 15 min ahead and a three-layer structure (input, hidden and output).

Table 9. Proposed options for the final energy forecaster.

Parameter	Option 1	Option 2	Option 3	Option 4	Option 5	Option 6
Learning Algorithm	LM	LM	LM	LM	LM	LM
Inputs	96	96	98	98	194	194
Neurons	10	10	10	15	10	5
Delay	1:4	1:4	1:3	1:2	1:4	1:4

Finally, a sensitivity analysis was performed in order to choose the optimal training time for each forecaster. This test relies on examining the evolution of the accuracy of each option by increasing the epochs of the learning step. The database used for these tests covers January 2016–December 2017 or January 2017–December 2017, depending on the length of the database required by each option. To analyse the accuracy of these tests, the average error in percentage for the forecasted period that covers January 2017–April 2017 was calculated. Because each test was

repeated five times, the values presented in Table 10 are an average of the results obtained for those five tests.

Table 10. Evolution of the average error in percentage for each option.

Epochs	Option 1	Option 2	Option 3	Option 4	Option 5	Option 6
3	1.40	2.00	1.20	0.93	1.07	1.32
6	0.28	0.45	0.19	0.38	0.24	0.35
9	0.23	0.23	0.17	0.18	0.20	0.34
12	0.24	0.22	0.15	0.18	0.22	0.30
15	0.29	0.26	0.15	0.16	0.28	0.39

An analysis of the results in Table 10 revealed that the relationship between the forecasters' improvement in accuracy and the epochs of the learning step was similar to an exponential function. For this reason, little improvement was expected in the average error if the epoch was increased when the exponential elbow point was achieved. In addition, it can be seen how, in certain cases, such as options 2 and 5, the error increased if the epoch time continued increasing. Hence, Table 11 presents the optimal epoch number chosen for each option when the final RNNs were trained on their respective whole databases.

Table 11. Epochs number for each option.

Forecasters	Epochs
Option 1	9
Option 2	12
Option 3	6
Option 4	9
Option 5	9
Option 6	12

4. Results and Discussion

Once the architecture and the training step epochs for each forecaster had been determined, the accuracy of these needed to be analysed. To select the best forecaster between the different choices, each forecaster predicted the power demand from January 2018 to August 2018. For these tests, data outside the database applied to the learning steps were used, so that the real accuracy of the developed forecasters could be examined. Figure 4 shows the demand curve of the building under study prior to analysing the results, in order to have an idea of the order of magnitude of the demanded energy. Moreover, Figures 5–7 compare the forecasters' computed error metrics in the analysed period.
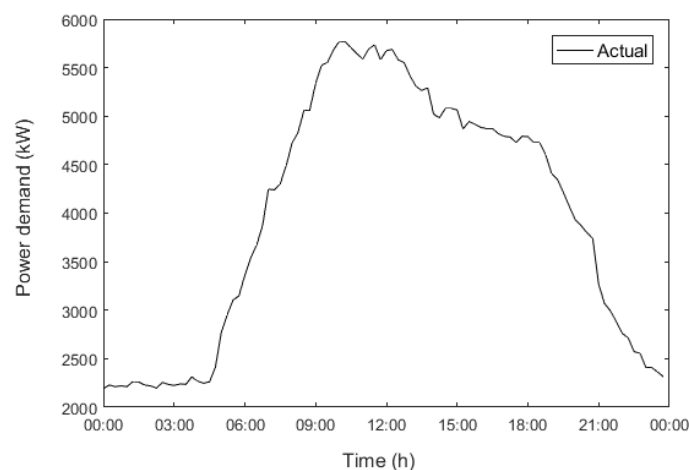

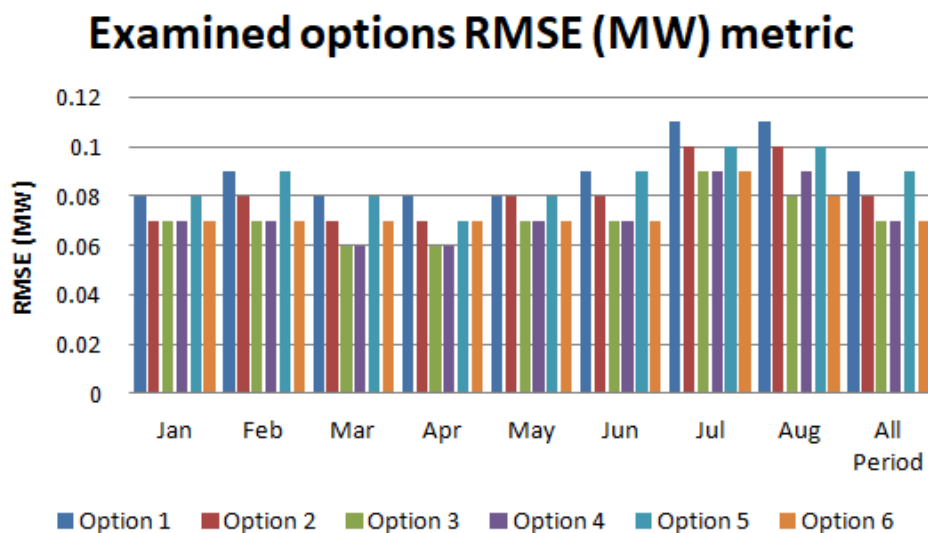**Figure 4.** Power demand of the building under study on 15 January 2018.

Figure 5. Root mean squared error (RMSE) (MW) metric results for examined options (period January–August 2018).
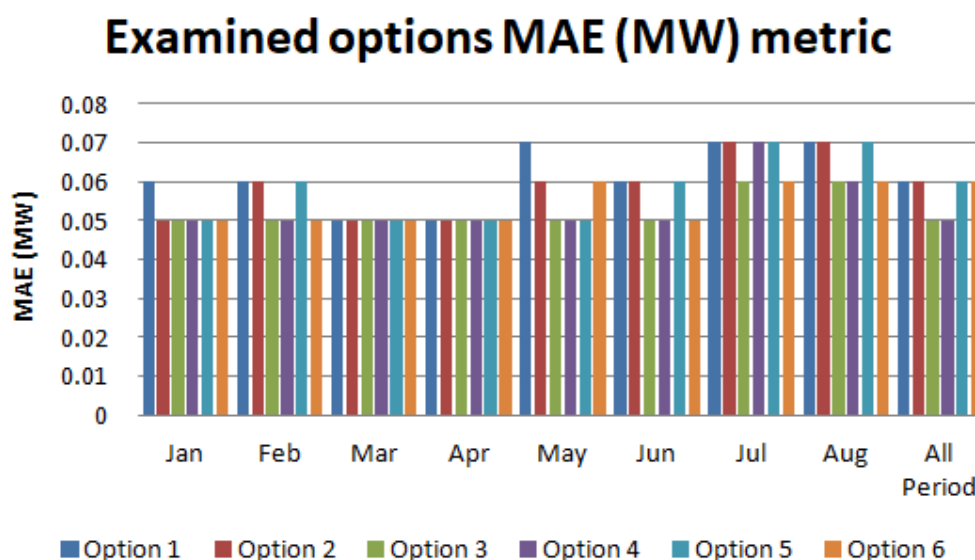

Figure 6. Mean absolute error (MAE) (MW) metric results for examined options (period January–August 2018).

With regard to the proposed choices, three forecasters (Options 3, 4 and 6) slightly outperformed the others when error metrics were analysed. Because the error metrics of these forecasters were quite similar, a deeper analysis needed to be done to select the optimal forecaster. For this new analysis, the difference in percentage between the forecasted and real power demand was analysed for each sample day in the period January-August 2018 for forecasters 3, 4 and 6. While Figure 8a presents the trend of the forecasted and real demanded power curves obtained on 31 August, 2018 by forecaster 3, Figure 8b shows the accumulated energy for same sample day. On this sample day, the difference between the actual and forecast power demand is 0.14%; while the real power demand was 62.00 MWh and the predicted power demand was 62.09 MWh.

Examined options MAPE (%) metric
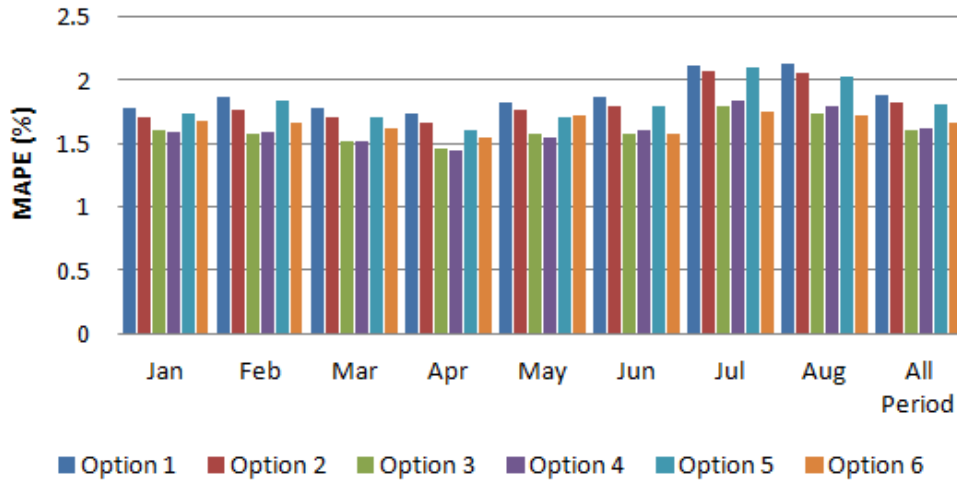
Figure 7. Mean absolute percentage error (MAPE) (%) metric results for examined options (period January–August 2018).

| (**a**) | (**b**) |
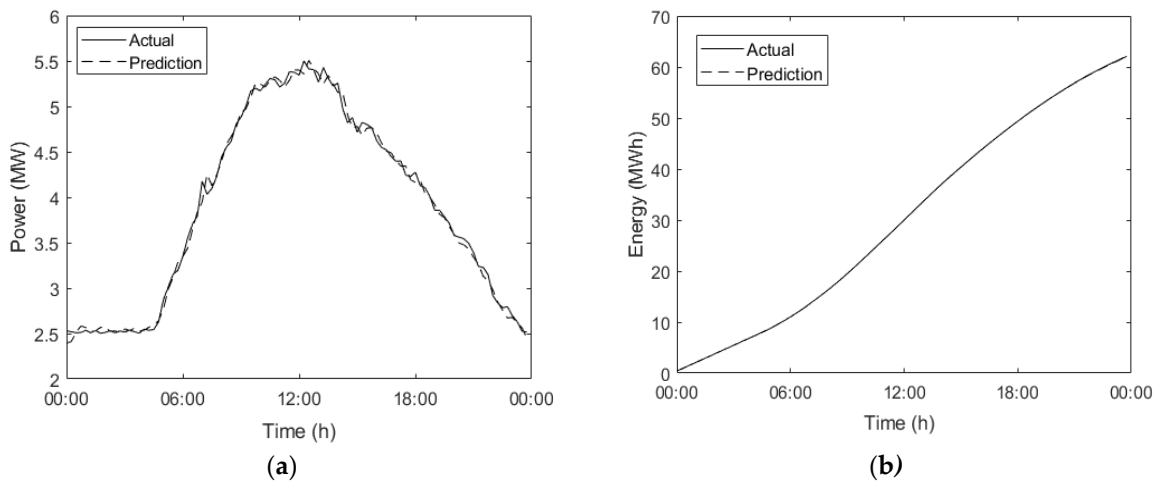
Figure 8. (**a**) Difference between real and predicted power demand on 31 August 2018 (**b**) accumulated energy.

Moreover, the pie charts in Figures 9–11 present the results of the proposed test to select the optimal forecaster from among the available choices.
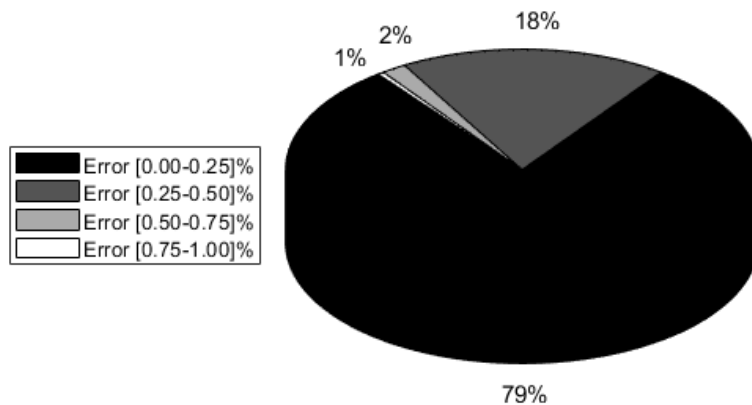

Figure 9. Error distribution in % of forecaster 3 (period January–August 2018).
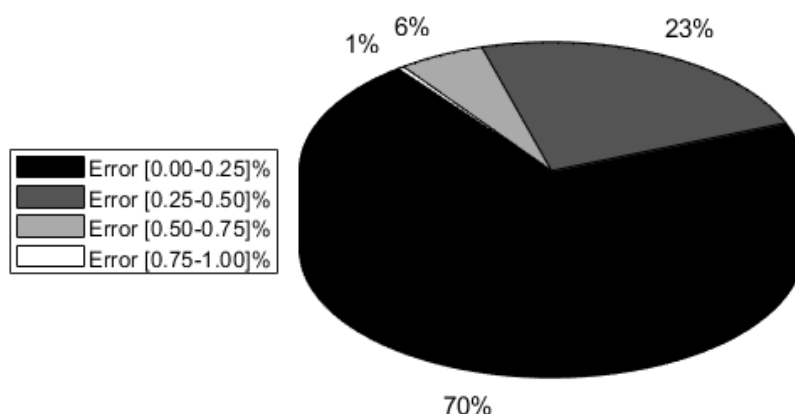
Figure 10. Error distribution in % of forecaster 4 (period January–August 2018).
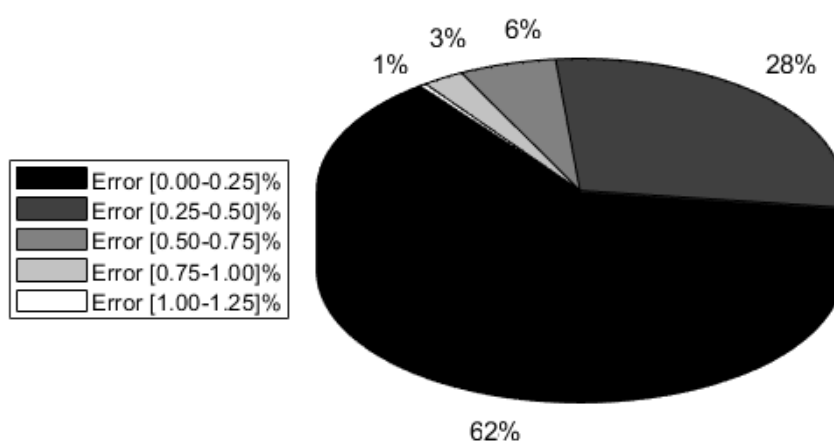
Figure 11. Error distribution in % of forecaster 6 (period January–August 2018).

An examination of the pie diagrams for the different options led to the conclusion that the best architecture for the forecaster is option 3. While, in option 3, the average error in percentage is 0.16%, options 4 and 6 have an error average of 0.19% and 0.22%, respectively. Table 12 summarises the key parameters of the forecaster that was selected for this study.

Table 12. Key parameters of the developed forecaster.

Parameter	Value
Learning algorithm	Levenberg–Marquardt (LM)
Inputs	Season, Hour and Power demand
Database length	1 year
Outputs	Power demand
Layers	3
Neurons	10
Delay	1:3
Epochs	6

To compare the results obtained by the forecaster developed in this study with the forecasters proposed in literature, it was necessary to check not only whether the prediction horizon time was similar but also whether the power demand values were on the same order of magnitude. The fact that small variations in power demand would have different effects on the error metrics if the analysed system demanded MW, kW or W also had to be taken into account.

De Andrade et al. [8] proposed a nonlinear autoregressive model with exogenous input (NARX) neural network for predicting power demand in two cities from 5 to 25 min ahead, in steps

of 5 min. As the forecaster developed in this study predicts power demand 15 min ahead, the error metrics for both cities in this horizon time were compared with the error metrics presented in this study. Moreover, while the data that were given in [8] for City I and City II had a power demand range of 3–9 MW and 5–15 MW, respectively, the building that was analysed in this study had a power demand range of 2.25–5.75 MW. Therefore, the order of magnitude was the same and the error metrics of both forecasters could be compared.

If the MAPE error metric of both forecasters is compared, City I and II have a MAPE of 2.60% and 1.57%, respectively, while the forecaster developed in this study has a MAPE of 1.61%. Even when the power range of the building analysed in this study was lower than in the cities analysed in [8], it was hoped that the accuracy of the forecaster developed in this study would be worse due to the fact that slight and sudden changes in power demand would have a bigger effect in the building analysed in this study. However, the predictions made by the forecaster developed in this study exhibited slightly greater accuracy than the predictions made by the forecaster in [8].

De Andrade et al. [25] proposed a fuzzy neural network for predicting power demand for four different points 5 min ahead. It must be taken into account that the forecast horizon in [25] is lower than in the forecast developed in this study, so it was expected that the MAPE error was bigger in this study. The power demand ranges analysed in [8] were 11–14 MW, 5–11 MW, 1–8 MW and 6–16 MW and the MAPE errors were 0.84%, 1.45%, 0.47% and 1.88%. Although it seems that the forecaster proposed in [25] slightly outperforms the forecaster proposed in this study, it should be highlighted that the forecaster presented in this article predicted 15 min ahead with a degree of accuracy that was similar to the forecaster proposed in [25] makes for 5 min ahead. Therefore, it can be concluded that the proposed forecaster is comparable to the forecaster proposed in [25] since the MAPE error is similar and the prediction horizon is higher in this study.

Rana et al. [10] proposed a wavelet neural network (WNN) for forecasting a power demand range of 6000–9250 MW for 5–60 min ahead in 5-min time steps. WNNs combine wavelet theory and neural networks. Wavelet theory is based on decomposing the high and low frequency components of the parameters used in the input vector data that are applied in an ANN. The error metrics provided by [10] for a 15minute ahead horizon forecast time are a MAE of 39.95 MW and a MAPE of 0.45%. However, the error metrics provided by this study are a MAE of 0.05 MW and MAPE of 1.61%. Although the forecast horizon is the same, the order of magnitude of the power demand range is much higher in [10] than in this study. For this reason, the MAE error metric is better in this study than in [10], whereas in the MAPE error metric [10] obtains better results than those provided in this study. To summarise, while the MAE error metric improves from 39.95 MW to 0.05 MW, which is an improvement of 99%, the MAPE error metric worsens from 0.45% to 1.61%, which is a worsening of 72%. As explained above, the higher the order of magnitude of the power range, the higher the MAE and the lower the MAPE error metrics are. Nevertheless, the MAE error improves more than the MAPE worsens, so it can be concluded that the forecaster presented in this study is comparable to the forecaster proposed in [10].

5. Conclusions

This study presents a forecaster that is able to predict power demand in the very short term, specifically 15 min ahead. The main conclusions drawn from this study are explained below.

The study presents a sensitivity analysis where not only the optimal input vector, but also the length of the database were examined. It was found that the optimal architecture for the forecaster was a layer recurrent neural network with ten neurons and a 1:3 delay ratio in the hidden layer. In addition, while the pie chart diagram for the third forecaster demonstrates that 79% of the error was between 0 and 0.25%, the average error in percentage for the validation step was the lowest (0.16%) of the different options analysed.

Although the forecasters' accuracy was validated with real data, the steps explained in Section 3.2 had to be taken to ensure that the optimal number of neurons and delay ratio was selected. Nevertheless, it was not necessary to repeat the process of selecting from among the different

possible forecasters, due to the fact that the results of forecaster 3 outperformed the predictions made by the other options.

The analysis of the results provided by each option demonstrated that the season and hour of the day were key parameters in predicting power demand. In addition, after analysing the results, it was concluded that a database for a single year provided better results than a two-year database. Unexpectedly, if temperature was included as a parameter in the input vector, the forecaster's error rate increased. This happened because the temperature changes were more volatile than the power demand changes, thus causing the uncertainty of the forecaster to increase.

The error metrics presented in this article were compared with the available literature and it was concluded that there was a slight improvement when the prediction horizon was 15 min and the power demand range was similar to or higher than the building examined in this study.

Since microgrids and smart grids are equipped with smart metres that record energy demand in real time, the methodology proposed in this research study can be applied to develop other forecasters to predict with sufficient accuracy the power demand 15 min ahead and send this information to the central unit control. This will allow the unit control to have more information about the future situation and make better decisions related to increasing the efficiency of the whole system.

Author Contributions: Conceptualization, L.F.; Formal analysis, F.R. and F.M.; Funding acquisition, L.F.; Investigation, F.R. and A.G.; Methodology, F.R., F.M. and A.G.; Supervision, L.F. and A.G.; Validation, F.R. and F.M.; Writing – original draft, F.R.; Writing – review & editing, F.M., L.F. and A.G. All authors have read and agreed to the published version of the manuscript

References

1. Che, J.; Wang, J. Short-term load forecasting using a kernel-based support vector regression combination model. *Appl. Energy* **2014**, *13*, 602–609.
2. He, Y.; Zheng, Y. Short-term power load probability density forecasting based on Yeo-Johnson transformation quantile regression and Gaussian kernel function. *Energy* **2018**, *154*, 143–156.
3. Chitsaz, H.; Shaker, H.; Zareipour, Wood, D.; Amjady, N. Short-term electricity load forecasting of buildings in microgrids. *Energy Build.* **2015**, *99*, 50–60.
4. Saber, A.Y.; Alam, A.R. Short term load forecasting using multiple linear regression for Big Data. In Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–6.
5. Elgarhly, S.M.; Othman, M.M.; Hasanien, A.T. Short term load forecasting using ANN technique. In Proceedings of the Nineteenth International Middle East Power Systems Conference (MEPCON), Menoufia University, Egypt, 19–21 December 2017.
6. Chen, Y.; Xu, P.; Chu, Y.; Li, W.; Wu, Y.; Ni, L.; Wang, K.; Bao, Y. Short-term electrical load forecasting using the Support Vector Regression (SVR) model to calculate the demand response baseline for office buildings. *Appl. Energy* **2017**, *195*, 659–670.
7. Ahmad, T.; Chen, H. Utility companies strategy for short energy demand forecasting using machine learning based models. *Sustain. Cities Soc.* **2018**, *39*, 401–417.

8.　L.C.M. de Andrade, M.; Oleskovicz, A.; Quaresma, D.V.; Souza, R.A. Very short-term load forecasting based on NARX recurrent neural networks. In Proceedings of the IEEE PES General Meeting Conference & Exposition, National Harbor, MD, USA, 27–31 July 2014.

9.　Guan, C.; Luh, P.B.; Michel, L.D.; Friedland, P.B.; Wang, Y. Very short-term load forecasting: Wavelet neural networks with data pre-filtering. *IEEE Trans. Power Syst.* **2013**, *28*, 30–41.

10.　Rana, M.; Koprinska, I. Forecasting electricity load with advanced wavelet neural networks. *Neurocomputing* **2016**, *182*, 118–132.

11.　Jiang, J.; Li, G.; Bie, Z.; Xu, H. Short-term load forecasting based on higher order partial least squares (HOLPS). In Proceedings of the IEEE Electrical Power and Energy Conference, Saskatoon, SK, Canada, 22–25 October 2017.

12.　Jiang, H.; Ding, F.; Zhang, Y. Short-term load forecasting based automatic distribution network reconfiguration. In Proceedings of the IEEE Power & Energy Society General Meeting, Chicago, IL, USA, 16–20 July 2017.

13.　Dudek, G.; Pelka, P. Medium-term electric energy demand forecasting using Nadaraya-Watson estimator. In Proceedings of the 18th International Scientific Conference on Electric Power Engineering (EPE), Koutynad Desnou, Czech Republic, 17–19 May 2017.

14.　İlseven, E.; Göl, M. Medium-term electricity demand forecasting based on MARS. In Proceedings of the IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), Torino, Italy, 26–29 September 2017.

15.　Carvallo, J.P.; Larsen, P.H.; Sanstad, A.H.; Goldman, C. Long term load forecasting accuracy in electric utility integrated resource planning. *Energy Policy* **2018**, *119*, 410–422.

16.　Ahmad, T.; Chen, H. Potential of three variant machine-learning models for forecasting district level medium-term and long-term energy demand in smart grid environment. *Energy* **2018**, *160*, 1008–1020.

17.　Farzana, S.; Liu, M.; Baldwin, A.; Hossian, M.U. Multi-model prediction and simulation of residential building energy in urban areas of Chongqing, south west China. *Energy Build.* **2014**, *81*, 161–169.

18.　Kuster, C.; Rezgui, Y.; Mourshed, M. Electrical load forecasting models: A critical systematic review. *Sustain. Cities Soc.* **2017**, *35*, 257–270.

19.　Dudek, G. Pattern-based local linear regression models for short-term load forecasting. *Electr. Power Syst. Res.* **2016**, *130*, 139–147.

20.　Verdejo, H.; Awerkin, A.; Becker, C.; Olguin, G. Statistic linear parametric techniques for residential electric energy demand forecasting. A review and an implementation to Chile. *Renew. Sustain. Energy Rev.* **2017**, *73*, 1104–1122.

21.　Baliyan, A.; Gurav, K.; Mishra, S.K. A review of short term load forecasting using artificial neural network models. *Procedia Comput. Sci.* **2015**, *48*, 121–125.

22.　Ramli, M.A.M.; Twaha, S.; Al-Turki, Y.A. Investigating the performance of support vector machines and artificial neural networks in predicting solar radiation on a tilted surface: Saudi Arabia case study. *Energy Convers. Manag.* **2015**, *105*, 442–452.

23.　Shi, J.; Guo, J.; Zheng, S. Evaluation of hybrid forecasting approaches for wind speed and power generation time series. *Renew. Sustain. Energy Rev.* **2012**, *16*, 3471–3480.

24.　Lusis, P.; Khalilpour, K.R.; Andrew, L.; Liebman, A. Short-term residential load forecasting: Impact of calendar effects and forecasting granularity. *Appl. Energy* **2017**, *2017*, 654–669.

25.　de Andrade, L.C.M. Efficient Neurofuzzy model to very short-term load forecasting. *IEEE Lat. Am. Trans.* **2016**, *14*, 721–728.

26.　Staats, J.; Bruce-Boye, C.; Weirch, T.; Watts, D. Makrov chain based very short-term load forecasting realizing conditional expectation. In Proceedings of the International ETG Congress, Bonn, Germany, 28–29 November 2017.

27.　Rodríguez, F.; Fleetwood, A.; Galarza, A.; Fontán, L. Predicting solar energy generation through artificial neural networks using weather forecasts for microgrid control. *Renew. Energy* **2018**, *126*, 855–864.

28.　Haykin, S. *Neural Networks and Learning Machines*; Pearson Education, Inc: Upper Saddle River, NJ, USA, 2009.

29.　Kalogirou, S.A. Artificial neural networks in renewable energy systems applications: A review. *Renew. Sustain. Energy Rev.* **2001**, *5*, 373–401.

30.　Demuth, H.; Beale, M. *Neural Network Toolbox User's Guide*; The MathWorks, Inc: Natick, MA, USA, 2002.

31. Ruiz, L.G.B.; Rueda, R.; Cuéllar, M.P.; Pegalajar, M.C. Energy consumption forecasting based on Elman neural networks with evolutive optimization. *Expert Syst. Appl.* **2018**, *92*, 380–389.

32. Welch, R.L.; Ruffing, S.M.; Venayagamoorthy, G.K. Comparison of feedforward and feedback neural network architectures for short term wind speed prediction. In Proceedings of the International Joint Conference on Neural Networks, Atlanta, GA, USA, 14–19 June 2009.

33. Koschwitz, D.; Frisch, J.; van Treeck, C. Data-driven heating and cooling load predictions for non-residential buildings based on support vector machine regression and NARX recurrent neural network: A comparative study on district scale. *Energy* **2018**, *165*, 134–142.

34. Ouynag, T.; Zha, X.; Kin, L. A combined multivariate model for wind power prediction. *Energy Convers. Manag.* **2017**, *144*, 361–373.

35. Taylor, J.W. An evaluation of methods for very short-term load forecasting using minute-by-minute British data. *Int. J. Forecast.* **2008**, *24*, 546–658.

36. Alhussein, M.; Haider, S.I.; Aurangzeb, K. Microgrid-level energy management approach based on short-term forecasting of wind speed and solar irradiance. *Energies* **2019**, *12*, 1487.

37. Khalid, R.; Javaid, N.; Al-zahrain, F.A.; Aurangzeb, K.; Qazi, E.H.; Ashfaq, T. Electricity load and price forecasting using Jaya-long short term memory (JLSTM) in smart grids. *Energies* **2020**, *22*, 10.