energies

MDPI

Article

Reduction of Computational Burden and Accuracy Maximization in Short-Term Load Forecasting

Alfredo Candela Esclapez *, Miguel López García, Sergio Valero Verdú and Carolina Senabre Blanes

Electrical Engineering Area, Miguel Hernández University, Av. de la Universidad, s/n, 03202 Elche, Spain; m.lopezg@umh.es (M.L.G.); svalero@umh.es (S.V.V.); csenabre@umh.es (C.S.B.)

* Correspondence: acandelaesclapez@gmail.com; Tel.: +34-677059024

Abstract: Electrical energy is consumed at the same time as it is generated, since its storage is unfeasible. Therefore, short-term load forecasting is needed to manage energy operations. Due to better energy management, precise load forecasting indirectly saves money and CO_2 emissions. In Europe, owing to directives and new technologies, prediction systems will be on a quarter-hour basis, which will reduce computation time and increase the computational burden. Therefore, a predictive system may not dispose of sufficient time to compute all future forecasts. Prediction systems perform calculations throughout the day, calculating the same forecasts repeatedly as the predicted time approaches. However, there are forecasts that are no more accurate than others that have already been made. If previous forecasts are used preferentially over these, then computational burden will be saved while accuracy increases. In this way, it will be possible to optimize the schedule of future quarter-hour systems and fulfill the execution time limits. This paper offers an algorithm to estimate which forecasts provide greater accuracy than previous ones, and then make a forecasting schedule. The algorithm has been applied to the forecasting system of the Spanish electricity operator, obtaining a calculation schedule that achieves better accuracy and involves less computational burden. This new algorithm could be applied to other forecasting systems in order to speed up computation times and to reduce forecasting errors.

Keywords: short-term load forecasting; computational burden; forecasting schedule; performance evaluation

check for updates

Citation: Candela Esclapez, A.; García, M.L.; Valero Verdú, S.; Senabre Blanes, C. Reduction of Computational Burden and Accuracy Maximization in Short-Term Load Forecasting. *Energies* **2022**, *15*, 3670. https://doi.org/10.3390/en15103670

Academic Editors: António Gomes Martins, José Luís Sousa and Luís Pires Neves

Received: 11 April 2022
Accepted: 16 May 2022
Published: 17 May 2022

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

Copyright: © 2022 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

Short-term load forecasting (STLF) is used to manage the production and distribution of electricity, and operate electricity markets from the current hour to the following days. If the electricity operator overestimates future electricity load, there will be extra production costs that will lead to economic losses. On the other hand, if the electricity demand is underestimated, power plants may not have sufficient reserves for their generators to meet the energy demanded by the grid, compromising its stability and risking the possibility of a blackout. In addition, an accurate demand forecast indirectly facilitates the management of electrical energy from renewable energies.

In addition to the STLF, there are very short-term load forecasting systems, which forecast load for the forthcoming seconds or minutes for control purposes, such as that developed by Zhang et al. [1]; medium-term load forecasting for more than 3 weeks to manage operations, such as that made by Han et al. [2]; and long-term load forecasting for the months or years ahead, forecasting for facilities planning, for example the system presented by Hong et al. [3].

Electricity networks are systems that distribute power at both a regional and national scale, therefore even a slight improvement in the accuracy of STLF supposes a significant economic saving, which increases the competitiveness and development of multiple large-scale companies. In addition to electricity operators, other entities benefit from accurate

electricity load forecasts, such as power marketers, independent system operators, or load aggregators.

Forecasting systems must process multiple variables whose relationships among them vary with location and weather: previous loads, meteorological data, calendar day, etc. Furthermore, electric load always contains a random unpredictable variation. Forecasting electricity load is a complex problem, which has been approached through various methods and from different points of view over the last few decades, which has led to a great variety of forecasting algorithms implemented in different electricity networks.

Many techniques are based on neural networks [4–8]. Other algorithms use statistical methods [9,10]. Hybrid systems that combine neural networks with other techniques are also common [11–14].

1.1. Main Problem

STLF systems calculate forecasts frequently—most likely hourly. The system must obtain fast forecasts, so the operator can read the results and manage the actions that adjust to future load. The latest measurement systems tend to use quarter-hour intervals. In addition, forecasts within the EU must be calculated at quarter-hour intervals, as required by article 19 of Commission Regulation (EU) 2017/2195, of 23 November 2017, establishing a guideline on electricity balancing. The STLF systems that currently work with hourly intervals will considerably increase their computational burden, whenever they change to quarter-hour intervals. They will have to forecast four times more values due to increased granularity and they will do this four times more often, due to an increase in frequency. This paper addresses the problem of computational burden while attempting to increase the accuracy of the STLF systems already implemented.

The Spanish transmission system operator (TSO) is Red Eléctrica de España (REE) and it is working on hourly intervals. It needs forecasts for 19 electrical regions, which is 19 times more calculations than a single STLF, with a total of 2.5 s for every hour that is predicted nationwide. Due to time limits for submitting predictions, it is not always feasible to calculate all future hours. Therefore, there is a schedule that determines which future intervals are predicted during each hour of the day.

The REE forecasting system cannot keep the previous forecasting schedule with quarter-hour intervals, since it is too computationally heavy to work within the time restriction. Therefore, there is need for a new schedule to forecast only the most useful intervals. Furthermore, the new schedule must be based on a criterion that numerically determines which predictions are most useful. Satisfying the need for a systematic method to optimize schedules is the main motivation of this work.

1.2. Solution Approach

It was assumed that as we approach the forecast moment, the forecast becomes more accurate. However, this hypothesis does not always hold true. As an example, Figure 1 represents the mean absolute percentage error (MAPE) when predicting each hour of the Spanish national load, in 2019. Predicted values are distributed along the abscissa axis as they are executed hourly from 9 days before until the previous day. These forecasts are calculated with the auto-regressive model from REE [11]. There is an obvious trend in which accuracy increases as more recent information (weather and load) becomes available. Nevertheless, there seem to be some periods in which new forecasts are actually less accurate. The temperature data availability produces sudden drops in error, but during some periods the error increases. If accuracy loss periods can be known in advance, then the unproductive forecasts made at these times can be spared, saving computational effort while achieving a more accurate forecast. This paper aims to determine the optimal schedule of forecasts so that the system only computes new forecasts when an accuracy improvement is expected.

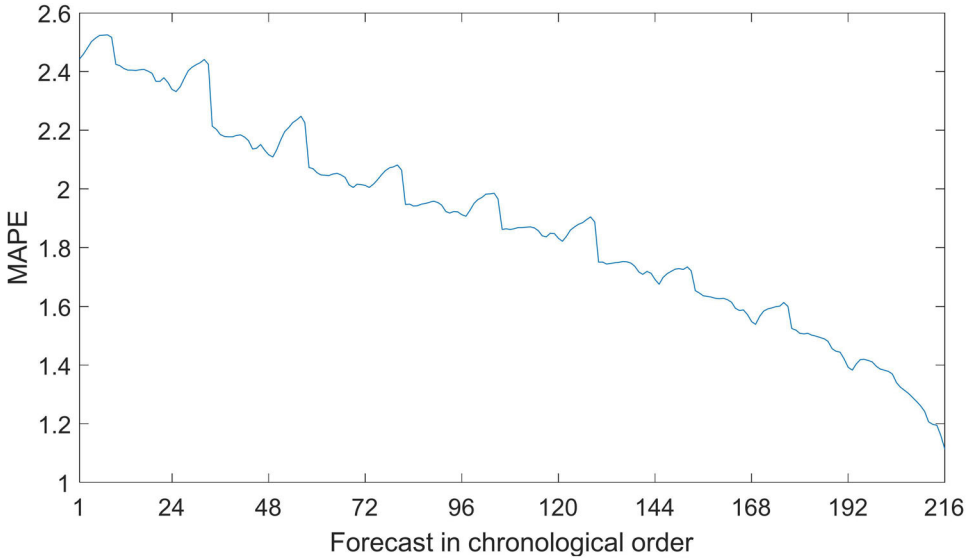

Figure 1. Error of auto-regressive forecasts in 2019.

The variations in precision are due to the availability of temperature and demand data. However, the objective of this paper is not to improve the input selection, but to determine at which moments it should be executed. The input selection problem had already been studied and solved [11].

The forecast needs to be computed within a time limit; therefore, each computer has a limit of N hourly values, beyond which the forecast would arrive late. This limit depends on the time allowed and the calculation speed, which again depends on the computer itself and on the forecasting algorithm used. To sum up, it may not be feasible to predict the whole range needed, therefore, in order to select the best N forecasts that can be calculated at each moment—a method to prioritize them needs to be developed. In addition, even if all predictions can be calculated, they may be counterproductive, since some of them have a greater predictive error than some of the previous ones.

1.3. Literature Review

The STLF field is extensive, since innumerable works have been published for decades; consequently, reviews of the state of the art have been published, such as those made by Mamum et al. [15], Hippert et al. [16], or Hong et al. [17].

The previous work on the STLF system used in this paper [18] compared the autoregressive and neural models used. The research defined which one performs better in different contexts, being determined by the model configuration, availability of data and the use of exogenous variables. The performance of both models was studied, in such a way that predictions that were calculated in the same time intervals with different models were compared; the time lapse between the execution moment and the predicted one is not a variable. On the other hand, the proposed research took into account the performance of the same model for different time lapses.

Other researchers [19–25] built and compared different STLF mathematical models employing error measures as performance indicators. After that, they did not consider how to apply those models in an optimized schedule, to avoid producing larger errors than past predictions that had already been calculated. J. Mohammed et al. [26] did something similar, which also included reliability indicators to assess the model's performance.

Another example is the work carried out by G. Veljanovski et al. [27], in which they proposed a forecasting system based on a neural network, that uses the day type and air temperature of the last seven days as inputs to obtain day-ahead load forecasts as output. The system was executed one day before the forecast day but they did not consider the best time at which to obtain data and execute the computation.

E. Weyermüller et al. [28] built a minimalistic adaptive neuro-fuzzy inference model, which uses the hourly load at the same time for the preceding day, day of week, and month of yearas input. It forecasts the load of one hour 24 h before, so this research could be applied to organize the calculation schedule if more forecast hours are added to the model.

The present work could complement automatic forecasting systems. This paper offers an automated extra step at the end of the modeling process, in order to obtain an optimized execution schedule. An example of automatically modeled systems is the work conducted by L. Shufen et al. [29], in which they proposed an algorithm to automate time series forecasting for nonexperts.

The analysis proposed in this work could be applied to future works of theoretical research. For example, T. Panapongpakorn et al. [30] developed a STLF system for Mae Hong Son, Thailand. It is made up of two different time series models and three different neural networks. It predicts 30 min ahead by employing historical load, temperature, irradiance, and ATR. If the forecast horizon is extended beyond 30 min in order to build a practical system, the main ideas of this work could be applied to make an appropriate schedule. Another example is the work by D. Shuai [31]; he created a least-squares support vector machine model to forecast load from relative humidity, rainfall, maximum, minimum, and average temperature. If a complete forecasting system is built with his ideas, this paper could be applied to help improve its performance.

H. Jiang et al. [32] examined their model for different anticipation times; they also compared different STLF models, taking into account error and computation times. However, anticipation times varied just from 5 min to 16 h ahead and they were used to assess models, in the same way that calculation times were employed to compare entire models. On the other hand, this work employs accuracy from different anticipation times to assess them and optimize calculation schedules, whilst computation times govern how many calculations can be chosen.

There is research which focuses on reducing computational burden, such as that by A. McIlvenna et al. [33], who modified a forecasting system through the reduction of integer variables by relaxation. This paper aims to optimize the use of a previously built forecasting system regardless of which one it is.

With a different approach, M. Weimar et al. [34] evaluated the improvement of a STLF system according to the economic savings with an econometric model. This is an example of how improving accuracy offers benefits that overcome developing costs.

1.4. Paper Contributions

The contributions of this paper can be summarized as follows.

This article offers a method to evaluate, at each moment of the day, the most useful predictions that are calculated and renewed on the prediction horizon. As far as we know there is nothing similar in the state of the art, making this is a first approach to evaluating single predictions. The most similar found works evaluate complete forecasting systems, for example H. Jiang et al. [32] or M. Weimar et al. [34].

The forecast evaluation allows the creation of an an optimal forecast schedule that reduces computational burden and increases overall accuracy, instead of focusing on improving computationally a single model, such as in the work by A. McIlvenna et al. [33].

The analysis of the prioritized schedule provides an understanding of the effect of new information on the quality of forecasts for both near and distant hours. This analysis could be applied to other researches, which use different inputs to understand its influence in accuracy, such as T. Panapongpakorn et al. [30] or D. Shuai [31].

According to Hong et al. [17], there is already a high number of STLF models for point predictions. For this reason, the present work does not develop a new model, but rather develops a technique with the potential to improve existing ones with respect to precision and particularly to computational burden.

1.5. Sections Summary

The paper is organized as follows. Section 2 gives an overview of the forecasting system used in this research. Section 3 explains the predictive scheduling algorithm in detail. Section 4 analyzes the accuracy of the algorithm when applied to the forecasting system. Its performance is compared with the current system. Section 5 summarizes the conclusions obtained.

2. Forecasting System Employed

The STLF system used in this research is that developed by the UMH [11], which was implemented in REE. The system has been operating for more than 4 years, and during this time REE and UMH have continued to collaborate in continuous improvement efforts [15,35].

The system entails an auto-regressive exogenous model and an exogenous autoregressive neural network (NARX) model. Each technique provides a separate forecast, which is then combined into a final forecast. This combination is more accurate than each separate result, so it is employed as final result by the operator.

The auto-regressive exogenous model employed by REE is described by Equation (1).

$$\ln(Y_t) = \sum_{i=1}^{d} E_{t-i}\,\alpha_i + \sum_{v=1}^{n} X_{v\,t}\,\Phi_v + \varepsilon_t \quad (1)$$

where Y is the time series of forecast load for an interval of the day t, E is the error time series from previous forecasts, α coefficients associated with the error time series, d is the number of previous days that are taken into account, X is the input vector to predict a day t composed of a number n of variables v, Φ are their respective coefficients, and ε is a Gaussian random variation with mean zero.

This model is named auto-regressive since it takes into account previous forecasts through variable E. In addition, it is exogenous due to the input vector X, which incorporates all the inputs mentioned previously.

The NARX model is a category of neural networks; it is a modification of feed-forward neural networks, which only have exogenous inputs. NARX models have an extra input: the forecast value from previous steps; then they have a feedback loop. To train the network or compute single-step forecasts, the feedback loop is replaced by the real load as input, since the real load from preceding load is available.

Both models have the same inputs to forecast the peninsular load, which are the following: temperature forecasts of the current and next 9 days for Madrid, Barcelona, Seville, Zaragoza, and Biscay; peninsular demand of the previous hour to execution; average demand for the 52 weeks prior to the last week; and calendar information [35], which distinguishes the following day variables: weekdays, the different national and provincial holidays including Christmas, the previous and following days, days after the time change, month of the year, and August weeks. As output, the STLF system provides the predicted loads for the current day and the next nine, as required by the Spanish TSO. Furthermore, this horizon will be increased in five more days in future years. In addition, each forecast may have different relevance: the accuracy of the current day and the next day is more important than that of the 8th or 9th.

Both models employ the same input preprocessing. A filtering system is included in the forecasting system to identify and correct misleading information about historical load; it identifies and corrects abnormalities by comparing them to previous data. Temperatures are linearized using three linear equations that model load versus temperature, each one for different temperature levels classified as cooling, heating, and warm; the last one is not employed, so there are two variables (cooling and heating) to define each temperature value. Calendar information is represented as one-hot code for every variable about type of day.

2.1. Forecasting Schedule

The system schedule with the current configuration of the electricity operator is represented in Table 1, which shows the ranges of forecast hours for each run time. Each cell in the table contains the future hours of the day to forecast, so that 0 corresponds to the time span between 0:00 a.m. and 1:00 a.m. The hours in between are also predicted. Rows indicate the time of the day when the system is running, and columns indicate the future forecast day. There is the possibility of predicting the current hour, since it is still useful for the operator during the first 10 min of the hour itself. For example, at run time 0:00, on the first row of table (a), only the ninth following day is forecast. As another example, at run time 9:00 a.m., on the first row of table (a), all the following days except for the second are forecast, also the current day is forecast from 9:00 until 23:00 (the entire rest of the day).

Table 1. Current schedule employed by REE.

(a)						(b)					
Run Time	**Future Day Whose Load Is Forecast**					**Run Time**	**Future Day Whose Load Is Forecast**				
	Current Day	**1**	**2**	**3–8**	**9**		**Current Day**	**1**	**2**	**3–8**	**9**
0					0–23	12	12–23				
1						13	13–23	0–23			
2						14	14–23				
3	3–23					15	15–23				
4						16	16–23				
5	5–23	0–23	0–23			17	17–23	0–23	0–23		
6						18	18–23				
7	7–23	0–23				19	19–23				
8	8–23	0–23	0–23			20	20–23				
9	9–23	0–23		0–23	0–23	21	21–23	0–23			
10	10–23					22	22–23				
11	11–23	0–23				23	23	0–23			

(a) Schedule from 0 to 11. (b) Schedule from 12 to 23.

2.2. Data Employed

During training and forecasting, the STLF system uses calendar data. Holiday information is obtained from the Spanish Official State Gazette, known as Boletín Oficial del Estado (BOE). It also uses the historical load of the national network, which is provided by REE. Additionally, the system uses historical records and forecasts of maximum and minimum daily temperatures, which are provided by the State Meteorological Agency.

The data employed by the algorithm is divided into three groups: training, validation, and test.

- The training group is employed to train the STLF models, these are the data from the years from 2012 to 2018 inclusive. Therefore, the 7 years prior to the desired year to forecast was selected, as recommended in the analysis of the STLF system performance [18].
- The validation group is used by the algorithm to obtain error records, comprising the year prior to the test period: 2018, therefore it coincides with the last year of training. The validation period coincides with the end of training because it must be done with data from the 7 years prior to the year to be predicted. Moreover, the post-training data cannot be used, since the validation is simulated without it.
- The test group is used to verify the performance of the forecasting system with the implemented algorithm. In this case the year 2019 was selected.

3. Methodology

This study has been performed with a CPU i7-8700, 16 GB RAM and Windows Home 64 bits as the operating system. The STLF system employed runs in Matlab, so all the calculations from this paper have been carried out with the same software, version R2020a.

The time a computer needs to calculate a set of predictions depends on three factors: the time that it takes to load new input data (I), the number of forecasts (n), and the time that it takes to make each prediction (P). So, the run time (t) can be modeled with the linear Equation (2).

$$t = I + P \cdot n \tag{2}$$

Variables I and P depend on the computer and code employed. At the beginning of each hour, the forecasting system loads new input data (temperatures and previous measured load). This task requires 0.6329 s approximately, so this is the value for I. Moreover, every forecast hour requires different auto-regressive and neuronal models, since there is a model of both to forecast every hour from every run hour and previous day. Loading models from hard drive is not needed since they do not change throughout the year, therefore they can be kept in RAM memory to save loading times. Calculating each future hour with both models and combining the results requires 0.1336 s, then this is the p value.

As mentioned before, due to the limitation of accuracy or computational burden, for each execution, there is a maximum number, N, of predictions that can be performed without exceeding the response time limit. Therefore, for each execution period, up to N predictions with greater value can be selected.

The algorithm does not depend on the mathematical model used, but on the errors that it makes regarding historical records. Therefore, the major benefit of applying the algorithm is the error reduction sparing us from calculations with worse errors than previous ones already executed. In this way, it can be applied to a system that is organized by time intervals and it is possible to choose which intervals to predict in each hour of the day.

To measure error, the methodology employs the *MAPE* as indicator, described in Equation (3), where F is the forecast load at time d, L is the actual load, and n is the number of evaluated forecasts.

$$MAPE = \frac{100}{n} \sum_{d=1}^{n} \left| \frac{F_d - L_d}{L_d} \right| \tag{3}$$

The Figure 2 shows how to use the algorithm in a forecasting system. First, the maximum number of predictions that can be calculated per interval is obtained, which depends on available time and computing power. At the same time, the previous year can be predicted to calculate historical error records. Finally, the algorithm is applied to obtain a schedule that will serve to forecast load during the next year.

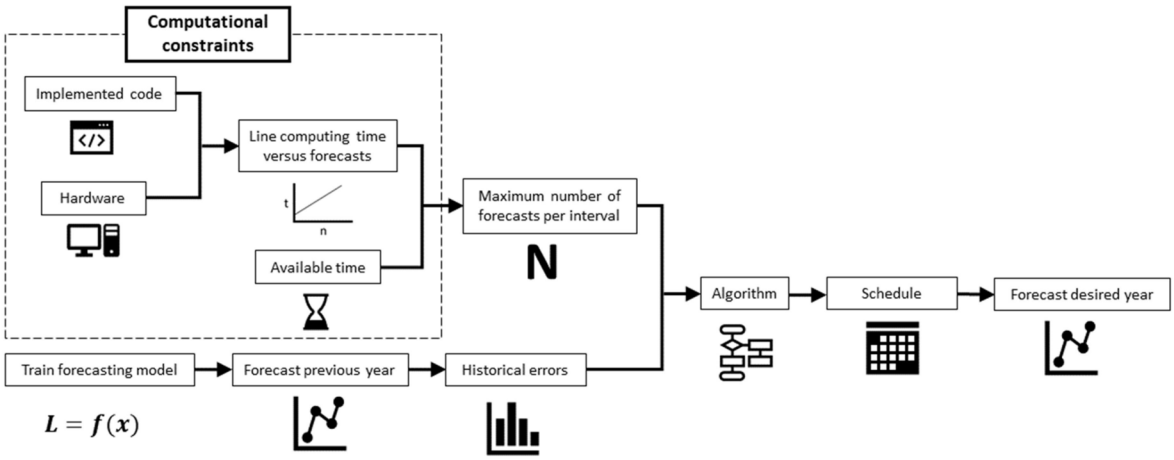

Figure 2. Process summary.

Other scheduling alternatives have been tested to compare the performance of the proposed algorithm. They are explained in Table 2.

Table 2. Scheduling options.

Name	Explanation
Proposed algorithm	Employ the proposed algorithm to obtain a schedule and then use it to forecast during the entire year.
Complete schedule	Predict every future intervals up to 9 days in advance, at every hour of the day.
Current planning	Use the current schedule from REE, which is explained with Table 1.
Optimized algorithm	Employ the proposed algorithm with the optimal forecast number.
Random selection	Forecast a number, N, of random future intervals at each hour of the day.
Last-day selection	This algorithm, at each moment, predicts the current day. It also forecasts the future day that has gone the longest time without updating, prioritizing those days which have never been forecast.

4. Proposed Algorithm

To measure the value of a prediction, a numerical indicator called accuracy improvement expectation (AIE) is used. As the name suggests, it reflects the expectation of improvement in accuracy of a predicted demand if it is recalculated. To calculate this parameter, historical records of predictions calculated under the same conditions are used; that is, forecasts calculated at the same time of day with the same advance period.

In order to define AIE, let P be a series of predictions, which have been calculated at different intervals of time, t. Each value in P predicts a load, L, of one same hour of a given future day. Then, the AIE_t of a specific prediction P_t is defined by Equation (4) as the difference of percentage errors between such prediction and a prediction of the same load made one interval before. Relative errors are multiplied by 100, so they are expressed as percentages.

$$\mathrm{AIE}_t = 100\left|\frac{P_{t-1}-L}{L}\right| - 100\left|\frac{P_t-L}{L}\right| \tag{4}$$

The implemented algorithm prioritizes the hourly forecasts according to larger AIE over the results of a full year, so only the first N values will be predicted in order to adhere to the time allowed.

Before executing the algorithm, the forecasting systems must be trained with the training data and ready to calculate forecasts. It is also necessary to determine the number of maximum forecasts, N, that the computer employed can execute. This is determined empirically by its computational speed and the calculation time limit. The data preprocessing from the original forecasting system is used in the same way, since it is not modified.

The algorithm core consists of two matrices named H and R. Matrix H is computed at algorithm initialization, so it remains constant during the rest of the process. It represents the historical records of the errors made in the validation period. It is used as a reference to estimate the error that the predictive model will have if it is executed at a given time to obtain the demand for a future hour. The matrix R varies with each iteration, representing the estimated error of the predicted demands in the prediction horizon. The values of matrix R are obtained by copying them from matrix H, according to the execution moment and the hour whose demand is expected.

The flowchart of the algorithm is described in Figure 3 and explained in depth in the following paragraphs:

Initialization: as a reference to estimate the AIE of each forecast, the algorithm forecasts the validation period. Each hour is predicted between 216 and 240 times, from the first time it enters the forecasting horizon until such an hour actually occurs. These reference forecasts make it possible to elaborate the H_{ijk} reference matrix, which contains the MAPE of the predicted hour, i, with j days of anticipation, where $j = 1$ the current day and $j = 10$ the ninth before, executed at time of the day k. The matrix H is obtained according to Equation (5), where F is the forecast load, L is the real load, t is the forecast day, and n is the number of days employed.

$$\mathrm{H}_{i\,j\,k} = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{F_{t\,i\,j\,k}-L_{t\,i}}{L_{t\,i}}\right| \tag{5}$$

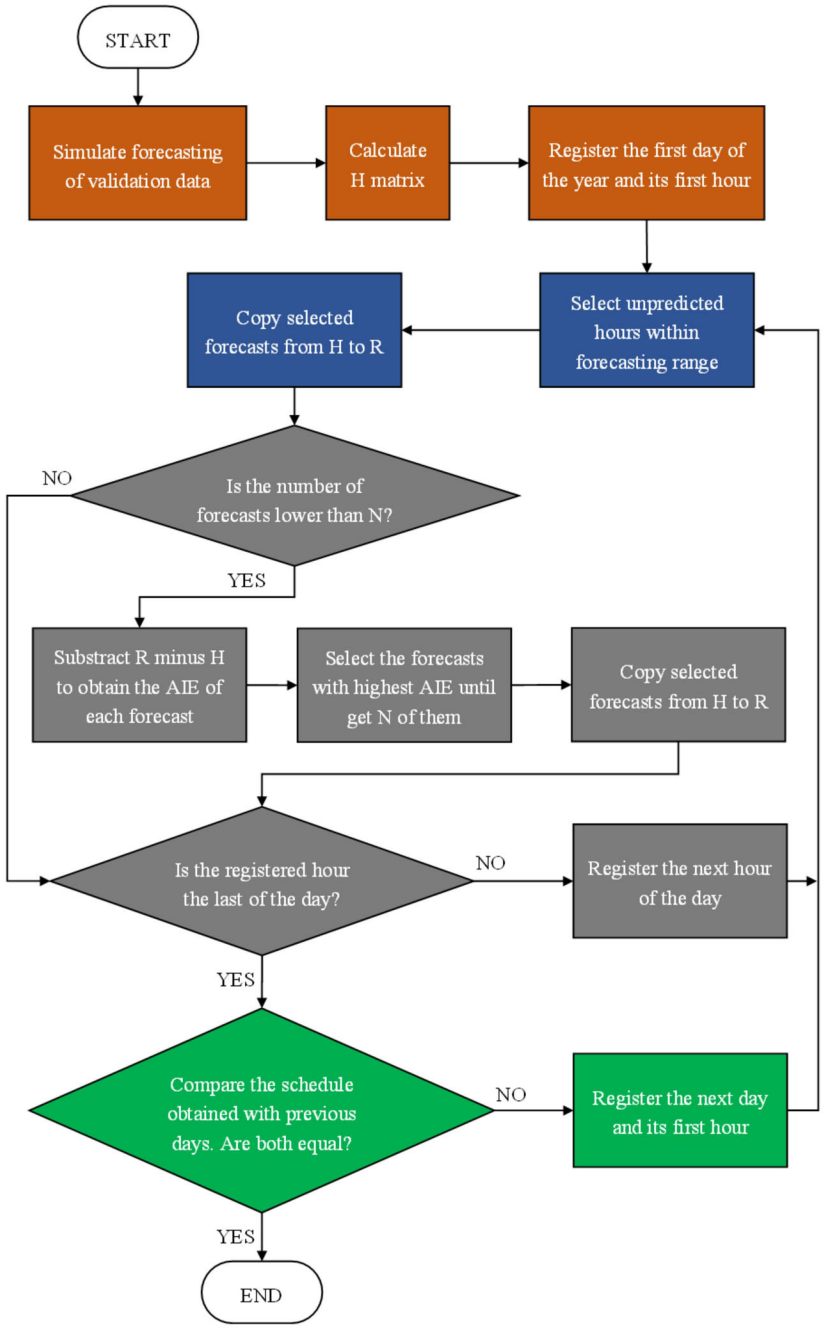

Figure 3. Algorithm flow chart.

Hourly iterations: For each of the 24 h on the first day of the year, the algorithm selects the N possible values within the forecasting horizon that will reduce the prediction error the most. To do this, it first checks if there are values without prior predictions, and it includes them among the selected ones. In addition, it copies the selected values from matrix H to matrix R as indicated in Equation (6). In this way, matrix R_{ij} contains the expected error of the last prediction made for hour *i* of future day *j*. At this point, all values within the horizon have either a current or a previous forecast. Next, if the number of predictions to be calculated is less than N, the rest of the forecasts are prioritized according to their AIE. To do this, according to time *k*, the submatrix H_{ij} is subtracted from R_{ij}. The models whose difference is greater will be included among those selected and the values of H will be transferred to R until the *N* predictions are completed. This process is carried out

for 24 h until a list of predictions to be calculated for each hour is obtained, so matrix R changes with every hourly iteration.

$$R_{i\ j} = H_{i\ j\ k} \quad (6)$$

Daily iterations: The algorithm repeats the previous iterations with the following days of the year, thus obtaining a schedule of executions for each day. The algorithm ends when two consecutive days get the same schedule for all 24 h.

5. Results

The result obtained by applying the algorithm is a schedule similar to the one in Table 3. Each cell in the table contains the future hours of the day to forecast, so that 0 corresponds to the time span between 0:00 a.m. and 1:00 a.m. Hours separated by a dash indicate that the hours in between are also predicted. Rows indicate the times of the day when the system is running and columns indicate the future forecast days. For example, at run time 19 (7:00 p.m.), for the second future day, hours forecast are 12, 14, 16, 17, 18, 19, and 21; in other words, periods between 12:00 and 13:00, between 14:00 and 15:00, and between 16:00 and 20:00. The relationship between the algorithm and Table 3 is described as follows: The algorithm uses the variable k as "hour of forecast", the variable j as "future day whose load is forecast", and the variable i as the "forecast hour". The combination of three variables is stored during the execution of the algorithm as explained in hourly iterations.

Table 3. Schedule for combined model.

Run Time	Future Day Whose Load Is Forecast									
	Current Day	1	2	3	4	5	6	7	8	9
0	0–1, 5, 9, 23	0, 18, 22	2–5, 10, 14, 16–18, 22–23	1, 10, 19, 21–23	4, 18, 22	2, 4, 13, 16, 22	11, 14, 22–23	2, 19–20, 23	2, 7, 12, 16, 22–23	0–23
1	1–5, 7, 22	1–2, 5–6, 9, 11, 23	0, 2–3, 7–8, 18–19	0–1, 4, 10, 17	0–4, 15, 18, 21	0, 3, 6–7, 11	0, 9, 19–20, 23	3, 5, 11–12, 15, 21	0, 3, 12, 14, 19	0–5, 7–8, 10–11, 13–14, 16–17, 19, 22
2	2–6, 8, 10, 13–14, 22	0–1, 6–7, 10–11, 17–18	2, 7, 10, 19, 21	1–2, 10, 12, 15, 18–19, 22	2–3, 12–13, 15, 18, 21	5, 10, 12–14, 18, 23	10, 12, 14	0, 4, 7, 11–12, 18, 21	0, 3, 12–13	0, 5–6, 9–14, 17, 19, 22
3	3–7, 9, 12–13, 18, 20	1–4, 6–7, 10, 12, 21	0, 6–7, 11, 18–19, 23	4, 8, 10, 13–14, 18–20	2–3, 13–14, 17–18, 22	3, 12, 15, 18	0, 2, 13, 19, 22–23	0, 11, 13, 23	0, 3, 11–14, 23	2–4, 9, 13–15, 18, 22
4	4–7, 10, 14, 16, 19, 22	1–6, 8, 10, 13, 18–19	1, 4, 7, 10, 15, 18–19, 21, 23	2–4, 7, 10, 12–13, 19	0, 2–3, 5, 17–18, 22	3, 10, 13, 15	11–12, 14–15, 17, 19, 23	0, 10, 12, 16, 18, 22	3, 12, 18	4, 10–12, 14, 18, 23
5	5–8, 12, 14, 16–17, 20–21	2, 4–5, 7, 17, 19	2, 5, 7, 10–11, 17, 21	7, 12–15, 17–20	2–3, 10, 13, 16, 18, 21–22	2, 4–5, 10, 12, 16–17, 22–23	8, 12, 15, 20, 23	0, 10, 12, 15, 18, 23	11, 19, 23	0, 5, 7, 11–13, 15, 18
6	6–9, 12, 14, 17, 19, 21, 23	0, 3–7, 9, 17, 20, 23	0, 3, 14–15, 22	1–2, 4–5, 9, 12–14, 16–18	2–3, 8, 12, 17–18, 21	2, 6, 13–14, 16, 22	0, 9, 15, 17, 22	0, 10–11, 15, 21, 23	3, 6, 10–11, 13, 18, 23	6, 8, 20, 22
7	7–9, 11–20, 22–23	0–3, 5–6, 8–9, 15, 18–20, 23	0, 2–5, 9, 11–12	1, 5, 7, 13, 18	2, 16, 22	2, 4, 12–14, 16–17, 19, 22	0, 8, 15, 19, 21–22	0, 11, 21	13, 18, 23	0, 5, 9–10, 13, 23
8	8–10, 16–18, 23	0–1, 4, 6–8, 10–12, 15, 17, 19–20, 23	1, 11, 13, 22–23	2, 9, 11, 14, 17, 20, 23	0, 2, 19, 21	4–5, 8, 17–19, 22	0, 10, 15, 18, 20–23	0, 7, 9, 15, 17, 19, 22–23	13, 18–19	2, 4, 6, 13, 15, 17, 21, 23
9	9		3, 19	15	0, 2, 8–14, 18, 20–21	8, 10, 12–14, 16, 22	8–10, 14–23	0, 2–7, 10, 13, 15–23	3, 6, 9, 13–19, 21–23	2, 13, 20–21
10	10–12, 18	6	2, 5, 13, 22–23	1, 13, 17, 21–23	3, 6, 15–17, 19, 22–23	0–1, 3–4, 11, 16–21, 23	0, 2, 4–5, 11, 13, 22	1, 8–9, 11–12, 14, 16	0–2, 5–6, 10–12, 20	0, 4–6, 10, 12, 15, 17–19, 22–23
11	11–20, 23	4, 7–10, 18, 23	1–2, 17–18, 21	2, 6, 8, 11–12, 14, 18–20, 22–23	1, 5, 9, 16–17	0, 2, 5–7, 9, 15, 23	3, 6–7, 12–13, 23	1, 8–9, 18, 20	0, 2, 4, 7–8, 15, 22	11, 13, 15–17, 20

Table 3. *Cont.*

Run Time	Future Day Whose Load Is Forecast									
	Current Day	1	2	3	4	5	6	7	8	9
12	12–17,19–21, 23	0, 4, 6, 9, 11,14, 18, 22	4, 7–9, 12, 15,17–18, 20–21	0, 4, 7, 9–11,13–16, 19–21	4, 7, 14, 16, 23	1, 5, 15	0–1, 6,11–12, 14–16	2–4, 7–8, 14	1, 3, 7–8	1, 14, 17, 23
13	13–23	10, 15, 17, 20	0, 5, 10, 12, 14, 16, 20–21	0–2, 4–5, 13–14, 16, 18, 20, 23	7, 13	2–3, 5, 12, 17–18	1, 6, 11–12, 17, 22	0, 2, 5, 8, 11, 14, 16, 18–20	0, 3–4, 10, 18, 22	3, 7, 9, 11, 13, 16, 19
14	14–19, 21, 23	1, 3, 10–12, 14, 16–17, 22	4, 11, 13, 15–16, 18–20, 23	3, 5, 7, 9, 12, 18–19, 23	5, 8, 13–14, 16–17, 21, 23	4, 9, 18	0–1, 21–23	0, 7, 9–10, 12 13, 15–16, 19, 21	1, 5, 20	1, 3, 11, 15, 17, 19, 22–23
15	15–23	0–1, 4–6, 8, 13, 15–17, 23	2, 9, 11–13, 16, 20	7, 10–12, 14, 16, 20	4, 6, 8–9, 12–13, 15, 19–21	13–14, 17–18	1, 5, 21–22	0–1, 4, 8–9, 12–13, 20, 23	1, 4, 9, 16–18, 20	5, 16, 18
16	16–22	0–3, 10–11, 13, 16, 18–19, 22	0, 2, 4, 12–13, 17–18, 23	1, 9, 11–12, 17–18, 23	1, 17–18, 20	2, 12, 22	2–3, 12–13, 21–22	1, 5, 7, 11–13, 15, 18–21, 23	2–3, 18, 20	1, 3, 6–7, 11–13, 19, 23
17	17–22	0, 2–5, 11–13, 15–19, 21	1, 4, 10–11, 14, 17–18, 20, 22	0, 5, 9–11, 18	4, 9, 11, 14–15, 20–21, 23	14, 17	5, 21, 23	0–2, 7, 9, 12, 19, 21, 23	1, 11–12, 18–19, 22–23	1, 3, 6, 8, 17, 19, 23
18	18–20, 22–23	1–2, 5, 9, 11, 13–15, 17, 19–22	2, 4, 12, 15–17, 20, 22	3, 10–11, 13, 16–17, 19, 21, 23	4, 10, 12–15	1, 7, 9, 14, 17, 21–22	0, 5–6, 8, 12, 18	1, 5, 8–9, 11, 15–16, 22–23	1, 5, 11, 19–20	7, 12–13
19	19–23	0, 2, 9, 12–13, 18–20, 22–23	12, 14, 16–21	11, 14, 16–19, 21, 23	0, 2, 12, 14, 20, 22	4–5, 13, 17–18, 20–22	1–3, 12, 14, 16, 21–23	12, 20, 23	0–3, 11, 13, 19, 21	3, 10, 15, 17, 19–20
20	20–23	0, 2, 5, 10, 14–20, 22–23	1, 3, 9–11, 16–18, 23	10, 15, 18–20	2, 14, 16–17, 19–21	1, 3, 9, 20–22	11, 14–15, 21, 23	1, 4, 6, 14, 19, 23	0, 2, 4, 8–9, 18, 20–21	1, 3–4, 7, 12, 19, 22–23
21	21–23	0–2, 4, 8–9, 11, 14, 17, 20, 22–23	3, 14, 16–17, 20	1, 4, 11, 16, 18–20	1–2, 10, 14, 17, 19–21	3, 11, 15, 18–23	0, 2–3, 13–14, 17, 23	4, 8, 12, 14, 16, 18–19, 23	3, 11–12, 18, 22–23	3, 7, 9, 19–20, 22
22	22–23	0–1, 13, 18–21	0, 2–4, 9–10, 17–23	0, 4, 10–11, 13, 18, 20, 23	1, 5, 10, 13, 15, 17–23	3, 9, 21–22	2–3, 11, 14, 17–18	0, 4, 12, 19–20, 23	1, 3, 8, 18–19	0, 3, 6–7, 18–20, 23
23	23	2, 4, 7, 9–10, 19–20, 22–23	1–2, 9, 18, 20–23	1–2, 7, 19, 21–23	1, 4, 10, 17, 21–22	1–2, 4–5, 18–19, 21–22	0, 3, 5, 12–13, 15–16, 20–21	1–2, 7–8, 18–20, 22–23	1, 4, 11, 15, 18, 20–22	1, 9, 12, 18, 20–21

Table 3 shows the 2019 schedule with the combined model and a forecasting limit per interval, N, of 71. The table is somewhat cumbersome as it describes all hours to be forecast at any given time. Nevertheless, the following conclusions can be extracted from it and are further substantiated in the following sections:

1. At 9:00 a.m., only 18.3% of forecasts are for the first 8 h of each future day. This allows us to infer that there is usually not a significant accuracy improvement in forecasting the first 8 h of each future day at 9:00 a.m. This conclusion is further developed in Section 5.1. "Temperature Influence on Early Morning".
2. At 9:00 a.m., only 5.6% of the predictions executed correspond to the first 4 days (including the current day). This selection makes sense since obtaining temperature data benefits more distant days than close ones, as seen in Section 5.1 "Temperature Influence on distant days".
3. For all running hours, the next hour is always forecast; while 22 of 24 running hours forecast the next two future hours and 21 of 24 forecast the next three. This fact coincides with the reasoning from Section 5.1 "Recent Loads", as it states that hours prior to the forecast moment tend to perform a great accuracy improvement.

5.1. Error Analysis

Error analysis has been performed on the results of both the auto-regressive and the NARX models. Nevertheless, the NARX model shows a more random behavior than the AR model and, even though the same conclusions apply to both, the identified patterns appear more clearly in the AR model. Therefore, for clarity reasons, the following section focuses on the conclusions obtained using the results of the AR model. Notwithstanding, all conclusions are valid for the AR, the NARX, and the combined model.

5.1.1. AIE and Error Analysis

The STLF system has been used to predict 2019, without applying the selection algorithm, so that it always calculates all future hours up to and including the ninth day after. The errors have been analyzed to detect patterns.

Temperature Influence on Early Morning

In order to quantify the effect of temperature updates, the MAPE of the entire forecasting horizon has also been calculated for each time of the day when forecasts are calculated. These errors are drawn in Figure 4; in order to distinguish the effects of temperature updates when predicting early morning and the rest of the day, the hours of the day that are predicted have been separated into two graphs.

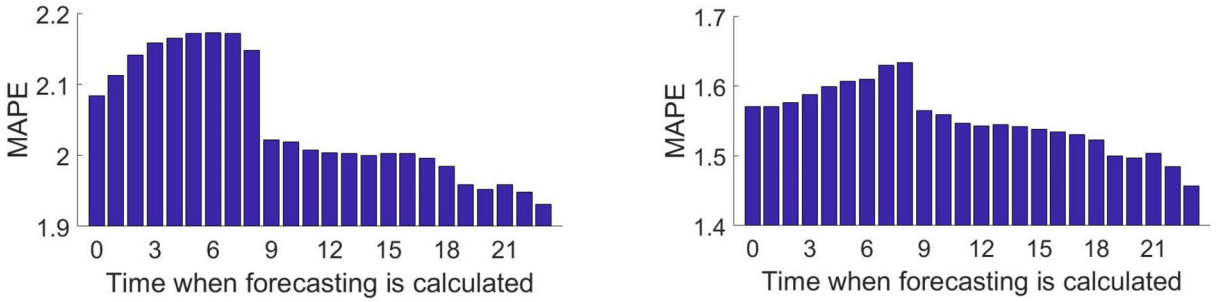

Figure 4. MAPE of every forecast moment during previous days. (**a**) Forecasting load consumed at 8:00 to 24:00. (**b**) Forecasting load consumed at 24:00 to 8:00.

Waking hours (8 a.m.–11 p.m.) experience an error decrease of 0.126% when executed at 9:00 a.m. compared to 8:00 a.m. On the other hand, resting hours have an error decrease of 0.068% at 9:00 a.m. compared to 8:00 a.m. At 9:00 a.m. the forecasting system receives new temperature data, then there is a larger correlation between temperature and load during waking hours than resting hours.

Temperature Influence on Distant Days

Figure 5 shows the AIE average of all the predictions at 9:00 a.m. from the previous 8 days to the preceding day.

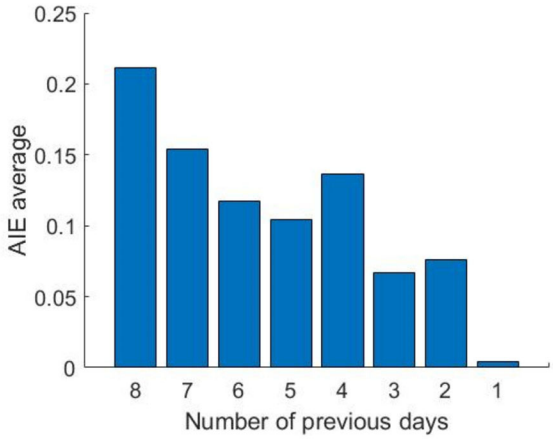

Figure 5. AIE average of previous days at 9:00 a.m.

The new temperature data results in greater improvements for distant days while not so much for proximate ones. Two reasons explain this: on one hand, temperature forecasts 2 or 3 days ahead are almost completely accurate and, therefore, each new forecast does

not add any value beyond that point. On the other hand, other variables (recent load) become more relevant than temperature as the forecasting hour approaches. Day 9 is not represented in Figure 5 because the temperature received at 9:00 a.m. is not an update, but brand-new information (there was no previous temperature forecast).

Recent Loads

To visualize the impacts that the latest demands have on forecast accuracy, actual temperature registers as opposed to forecasts have been used to avoid confusion between both variables. Figure 6 shows the MAPE made by the latest forecasts, which are distributed on the abscissa axis so that the first one corresponds to the most recent one.

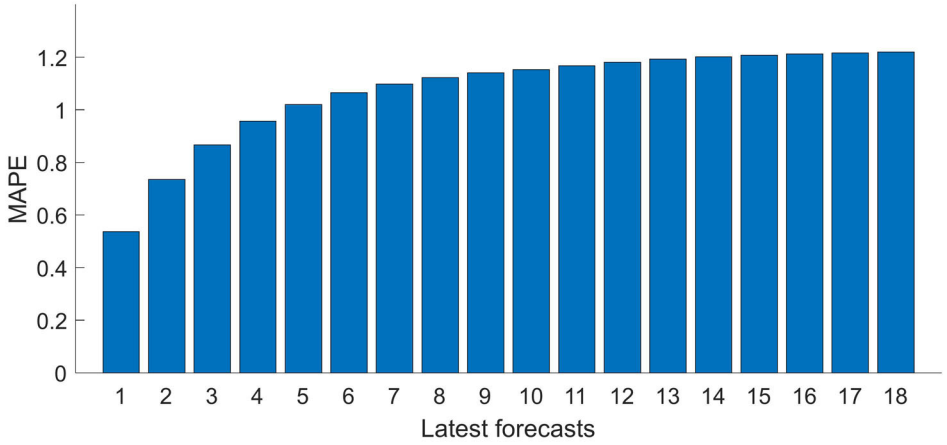

Figure 6. MAPE of latest forecasts.

According to Figure 6, the 3 h prior to the forecast moment have a higher error reduction than the rest. The greatest reduction in MAPE occurs on the latest forecast with 0.299%; the previous ones have a reduction of 0.131% and 0.090%, in that order. Therefore, it follows that there is a close correlation between the load for one hour and the previous ones.

5.2. Obtaining the Optimal Forecast Number

The algorithm has been applied to predict 2018 and 2019; so that to predict 2018, the years from 2011 to 2017 have been used as training and 2018 as validation datasets, while to predict 2019, the years from 2012 to 2018 have been used as training and 2018 as validation datasets. Figure 7 shows the MAPE average for each number of predictions that the system calculates. The solid lines refer to the STLF system with the algorithm applied and the dashed lines to the error made in predicting all the future hours that the system can cover; that is, the entire forecasting horizon, also understood as N equal to 240.

A pattern can be seen in which accuracy increases with more forecasts up to a certain point, beyond which allowing more forecasts increases computational burden, but does not improve accuracy. Both curves present a minimum between 60 and 100, which proves not only that the priority determined with data from previous years is valid for the next year but also that the optimal amount of forecast can also be established from such data.

The minimum MAPE of 2018 is 1.755% and achieved with N = 71; for this reason, the N value has been applied when obtaining the prediction times. Coincidentally, the optimal N value coincides with the 2019 minimum with N = 71 and 1.698% of MAPE.

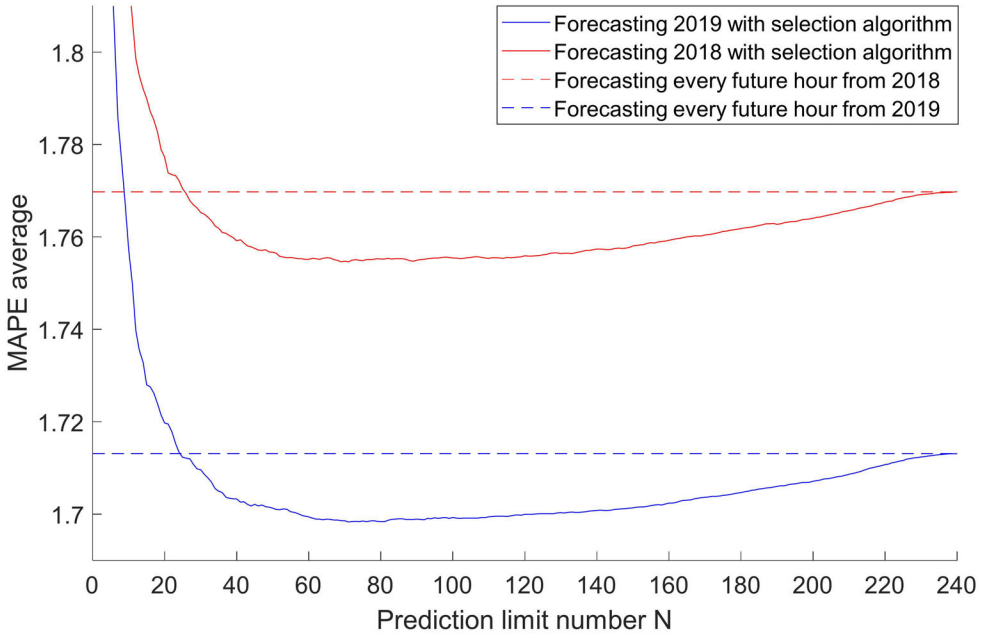

Figure 7. Accuracy of STLF system in 2018 and 2019.

5.3. Accuracy Results of Optimized Scheduling

In order to test the proposed algorithm, the year 2019 has been predicted, with each one of the scheduling options explained in Table 2. The MAPE average has been calculated for all the advances of all the hours of the year. Figure 8 shows the mean of the MAPE for each scheduling method. Methods with dashed lines are those that have a predetermined number, N, of future predictions.

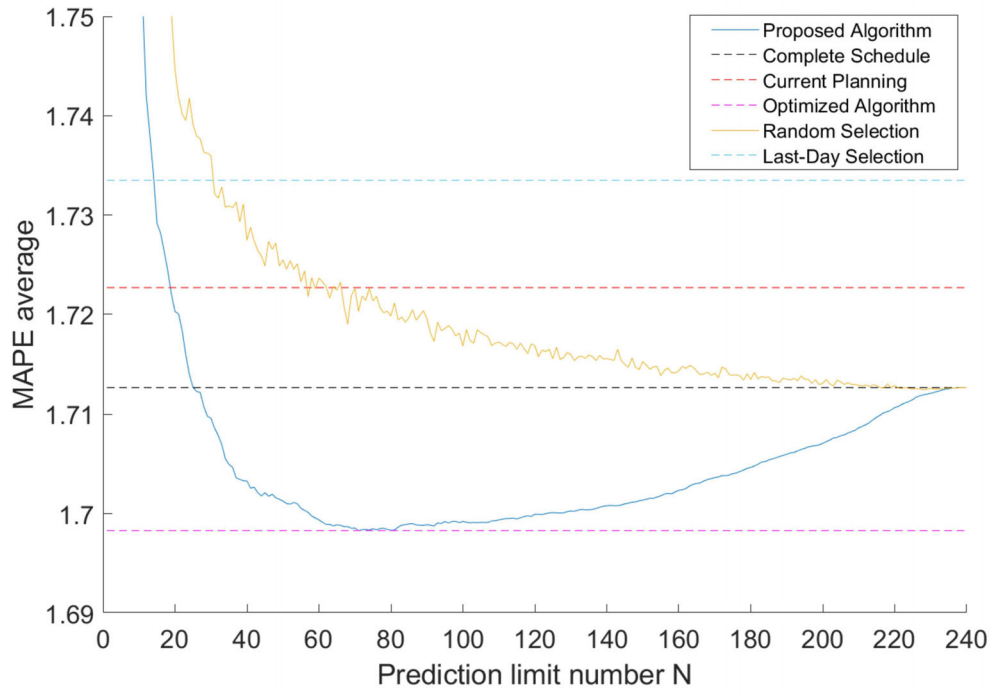

Figure 8. Accuracy of STLF system with different scheduling.

According to the test, random selection could perform worse for almost all cases; it is also inconsistent, so it is ruled out as a candidate. On the contrary, the proposed algorithm

offers significantly better precision than the rest of the methods, using between 60 and 100 limit predictions.

As a final test, the year 2019 has been predicted with every method. Figure 9 represents the MAPE obtained with forecasts made every hour from the previous 9 days until the previous day.

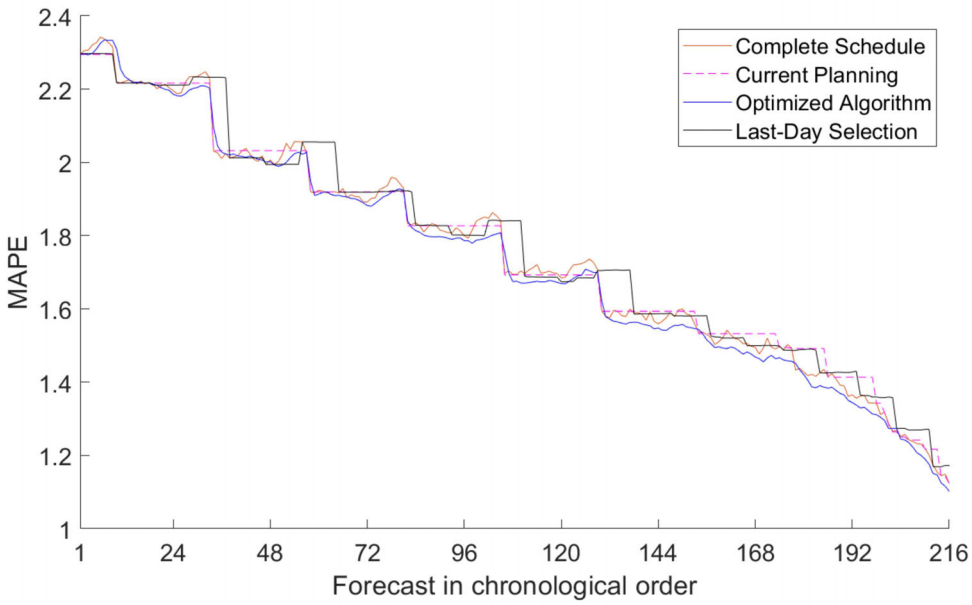

Figure 9. Accuracy of STLF in respect of order of execution.

In most advances, the optimized algorithm performs better than other methods. Certain combinations of execution time and future forecast days do not occur; in consequence, the last-day selection method has constant error periods due to a lack of updates.

As a final result, the year 2019 has been predicted with the current REE schedule, the algorithm with N = 71, and by calculating all future hours. Table 4 shows the MAPE of the entire year for each schedule and day in advance.

Table 4. MAPE of 2019 when employing each schedule and advance.

		Days in Advance								
Schedule	**Current**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Current Planning	1.089%	1.276%	1.478%	1.555%	1.630%	1.743%	1.861%	1.961%	2.101%	2.246%
Optimized Algorithm	1.066%	1.239%	1.417%	1.517%	1.612%	1.723%	1.841%	1.946%	2.088%	2.256%
Complete Schedule	1.077%	1.258%	1.443%	1.542%	1.634%	1.752%	1.862%	1.958%	2.095%	2.254%
Last-Day Algorithm	1.077%	1.291%	1.468%	1.550%	1.663%	1.768%	1.866%	1.992%	2.126%	2.245%

Except for the ninth day in advance, the algorithm always offers better accuracy. The Optimized Algorithm has an average improvement of 0.024% compared to the current schedule.

5.4. Computational Burden

The Spanish electricity system operator requires future load of 19 electrical regions. Nowadays, the entire forecasting horizon spans up to 240 h, thus the total of future loads that can be predicted extends to 4560, which require 10.16 min. However, if the quarter-hour system is employed, the number of future intervals to forecast will multiply by four. This new system will entail 18,240 numbers to be calculated in 40.62 min, which is unfeasible since REE requires results before 7 min have passed.

Table 5 shows the execution time for every scheduling in the worst-case scenario. That is forecasting 19 electric regions, during the hours of the day with the maximum number of future intervals to forecast. Execution times have been estimated employing Equation (2).

Table 5. Maximum execution times.

Scheduling	Run Hour with Maximum Intervals to Forecast	Number of Intervals to Forecast	Execution Time (Minutes)
Current planning	9:00 a.m. Look at Table 1.	3933 (207 × 19)	8.77
Optimized algorithm	Any hour. Every time it is executed it has a constant number of future intervals to forecast.	1349 (71 × 19)	3.01
Complete schedule	0:00 a.m. The system forecasts the entire current day and the following nines.	4560 (240 × 19)	10.16
Last-day selection	0:00 a.m. The system forecasts the entire current day and one of the following days.	912 (48 ×19)	2.04

Last-day selection and optimized algorithm manage to compute predictions under 7 min and the first one requires less time. However, the optimized algorithm offers better accuracy in most cases as indicated in Figure 9 and Table 4.

According to Equation (2) and quarter-hour intervals, the maximum number of forecasts to compute in 7 min is 3140. So, there is time to forecast 165 intervals in every electrical region. Therefore 165 is the maximum value that can be used on the algorithm as number N of forecasts.

6. Conclusions

European electricity operators need to develop systems with quarter-hour intervals and work in coordination with the rest of the operators in Europe. Figure 8 shows that accuracy is not proportional to the number of calculations, therefore a criterion is necessary to decide which predictions will be calculated. The increase in frequency and granularity can impose an extra constraint in forecasting models that can be detrimental to accuracy. In order to avoid any accuracy losses to the Spanish TSO, this research has developed an algorithm that organizes the calculation schedule of a STLF system throughout the day. The schedule obtained is adapted to the computational capacity of the computer while actually increasing the system accuracy from 1.089% and 1.276% to 1.066% and 1.239% for the current-day and next-day forecast respectively, with an average improvement of 0.023% for all previous days. The methodology described can be applied to any forecasting technique, even if computational burden is not an issue because it has been proven that limiting the number of forecasts can be beneficial for accuracy, as it has been demonstrated for the case of REE.

On the other hand, according to results, the main contribution of the work is to reduce the computational load of a predictive system without sacrificing accuracy. This will allow a transition to the quarter-hour system with an optimal execution schedule.

Furthermore, the error analysis provides a deeper understanding of how each temperature forecast update reduces the error on distant days but not so much on proximate ones, and how recent loads affect forecasting error drastically in the next 3 h but only marginally beyond hours 8 or 9.

This paper offers a first approach to improving forecasting systems through calculation planning, not only for electricity STLF but also for other time series forecasting systems, as well as other forecasting intervals such as quarter-hour intervals. Applying a similar study to other time series prediction systems could improve them in a similar way.

As future work, it is proposed to use the algorithm to plan the new quarter-hour system of the Spanish TSO. Each advance has a different impact on the operator's management, so forecasting errors have different impacts. To take this into account, each error can be multiplied by a weight factor according to the advance. In this way, with slight changes on the algorithm, it is possible to calculate schedules based on costs according to errors.

Author Contributions: Conceptualization, A.C.E. and M.L.G.; methodology, A.C.E. and M.L.G.; software, A.C.E., M.L.G. and C.S.B.; validation, M.L.G. and S.V.V.; formal analysis, A.C.E.; investigation, A.C.E. and M.L.G.; resources, S.V.V.; data curation, C.S.B. and M.L.G.; writing—original draft preparation, A.C.E.; writing—review and editing, M.L.G.; visualization, A.C.E.; supervision, S.V.V.; project administration, S.V.V.; funding acquisition, S.V.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partly funded by Red Eléctrica de España, TSO for the Spanish system as part of the project: Contrato para la Ampliación de los Trabajos de Mejora de la Previsión de Demanda Eléctrica a Corto Plazo. (REE1.19SW).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are openly available at https://www.esios.ree.es/es/analisis/1293?compare_indicators=545,544&start_date=10-07-2021T00:00&geoids= (accessed on 15 May 2022) for historical demand records and https://opendata.aemet.es (accessed on 15 May 2022) for historical temperature records.

Acknowledgments: Part of this research has been financed thanks to the R&D project of Red Eléctrica de España (REE) and the Miguel Hernández University (UMH) for the development of short-term load forecasting on quarter-hour data.

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

References

1. Zhang, Z.; Dou, C.; Yue, D.; Zhang, B. Predictive Voltage Hierarchical Controller Design for Islanded Microgrids Under Limited Communication. *IEEE Trans. Circuits Syst. Regul. Pap.* **2022**, *69*, 933–945. [CrossRef]
2. Han, L.; Peng, Y.; Li, Y.; Yong, B.; Zhou, Q.; Shu, L. Enhanced Deep Networks for Short-Term and Medium-Term Load Forecasting. *IEEE Access* **2019**, *7*, 4045–4055. [CrossRef]
3. Hong, T.; Wilson, J.; Xie, J. Long Term Probabilistic Load Forecasting and Normalization With Hourly Information. *IEEE Trans. Smart Grid* **2014**, *5*, 456–462. [CrossRef]
4. Kong, W.; Dong, Z.Y.; Hill, D.J.; Luo, F.; Xu, Y. Short-Term Residential Load Forecasting Based on Resident Behaviour Learning. *IEEE Trans. Power Syst.* **2018**, *33*, 1087–1088. [CrossRef]
5. Chen, Q.; Xia, M.; Lu, T.; Jiang, X.; Liu, W.; Sun, Q. Short-Term Load Forecasting Based on Deep Learning for End-User Transformer Subject to Volatile Electric Heating Loads. *IEEE Access* **2019**, *7*, 162697–162707. [CrossRef]
6. Niu, D.-X.; Wanq, Q.; Li, J.-C. Short term load forecasting model using support vector machine based on artificial neural network. In Proceedings of the 2005 International Conference on Machine Learning and Cybernetics, Guangzhou, China, 18–21 August 2005; Volume 7, pp. 4260–4265.
7. Singh, S.; Hussain, S.; Bazaz, M.A. Short Term Load Forecasting Using Artificial Neural Network. In Proceedings of the 2017 Fourth International Conference on Image Information Processing (ICIIP), Shimla, India, 21–23 December 2017; p. 5.
8. Chen, K.; Chen, K.; Wang, Q.; He, Z.; Hu, J.; He, J. Short-Term Load Forecasting With Deep Residual Networks. *IEEE Trans. Smart Grid* **2019**, *10*, 3943–3952. [CrossRef]
9. Wang, P.; Liu, B.; Hong, T. Electric load forecasting with recency effect: A big data approach. *Int. J. Forecast.* **2016**, *32*, 585–597. [CrossRef]
10. Yang, L.; Yang, H. A Combined ARIMA-PPR Model for Short-Term Load Forecasting. In Proceedings of the 2019 IEEE Innovative Smart Grid Technologies—Asia (ISGT Asia), Chengdu, China, 21–24 May 2019; pp. 3363–3367.
11. López, M.; Valero, S.; Rodriguez, A.; Veiras, I.; Senabre, C. New online load forecasting system for the Spanish Transport System Operator. *Electr. Power Syst. Res.* **2018**, *154*, 401–412. [CrossRef]
12. Bianchi, F.M.; De Santis, E.; Rizzi, A.; Sadeghian, A. Short-Term Electric Load Forecasting Using Echo State Networks and PCA Decomposition. *IEEE Access* **2015**, *3*, 1931–1943. [CrossRef]
13. Ma, Y.; Zhang, Q.; Ding, J.; Wang, Q.; Ma, J. Short Term Load Forecasting Based on iForest-LSTM. In Proceedings of the 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), Xi'an, China, 19–21 June 2019; pp. 2278–2282.
14. Abedinia, O.; Amjady, N.; Zareipour, H. A New Feature Selection Technique for Load and Price Forecast of Electrical Power Systems. *IEEE Trans. Power Syst.* **2017**, *32*, 62–74. [CrossRef]
15. Mamun, A.A.; Sohel, M.; Mohammad, N.; Haque Sunny, M.S.; Dipta, D.R.; Hossain, E. A Comprehensive Review of the Load Forecasting Techniques Using Single and Hybrid Predictive Models. *IEEE Access* **2020**, *8*, 134911–134939. [CrossRef]
16. 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]
17. Hong, T.; Fan, S. Probabilistic electric load forecasting: A tutorial review. *Int. J. Forecast.* **2016**, *32*, 914–938. [CrossRef]

18. López, M.; Sans, C.; Valero, S.; Senabre, C. Empirical Comparison of Neural Network and Auto-Regressive Models in Short-Term Load Forecasting. *Energies* **2018**, *11*, 2080. [CrossRef]
19. Sethi, R.; Kleissl, J. Comparison of Short-Term Load Forecasting Techniques. In Proceedings of the 2020 IEEE Conference on Technologies for Sustainability (SusTech), Santa Ana, CA, USA, 23–25 April 2020; pp. 1–6.
20. Jie-sheng, W.; Qing-wen, Z. Short-term electricity load forecast performance comparison based on four neural network models. In Proceedings of the The 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China, 23–25 May 2015; pp. 2928–2932.
21. Mehmood, S.T.; El-Hawary, M. Performance Evaluation of New and Advanced Neural Networks for Short Term Load Forecasting. In Proceedings of the 2014 IEEE Electrical Power and Energy Conference, Calgary, AB, Canada, 12–14 November 2014; pp. 202–207.
22. Sun, X.; Luh, P.B.; Cheung, K.W.; Guan, W.; Michel, L.D.; Venkata, S.S.; Miller, M.T. An Efficient Approach to Short-Term Load Forecasting at the Distribution Level. *IEEE Trans. Power Syst.* **2016**, *31*, 2526–2537. [CrossRef]
23. Rafi, S.H.; Nahid-Al-Masood, N.-A.-M. Highly Efficient Short Term Load Forecasting Scheme Using Long Short Term Memory Network. In Proceedings of the 2020 8th International Electrical Engineering Congress (iEECON), Chiang Mai, Thailand, 4–6 March 2020; pp. 1–4.
24. Kong, W.; Dong, Z.Y.; Jia, Y.; Hill, D.J.; Xu, Y.; Zhang, Y. Short-Term Residential Load Forecasting Based on LSTM Recurrent Neural Network. *IEEE Trans. Smart Grid* **2019**, *10*, 841–851. [CrossRef]
25. Rafi, S.H.; Nahid-Al-Masood; Deeba, S.R.; Hossain, E. A Short-Term Load Forecasting Method Using Integrated CNN and LSTM Network. *IEEE Access* **2021**, *9*, 32436–32448. [CrossRef]
26. Mohammed, J.; Bahadoorsingh, S.; Ramsamooj, N.; Sharma, C. Performance of exponential smoothing, a neural network and a hybrid algorithm to the short term load forecasting of batch and continuous loads. In Proceedings of the 2017 IEEE Manchester Power Tech, Manchester, UK, 18–22 June 2017; pp. 1–6.
27. Veljanovski, G.; Atanasovski, M.; Kostov, M.; Popovski, P. Application of Neural Networks for Short Term Load Forecasting in Power System of North Macedonia. In Proceedings of the 2020 55th International Scientific Conference on Information, Communication and Energy Systems and Technologies (ICEST), Niš, Serbia, 10–12 September 2020; pp. 99–101.
28. Weyermüller, E.; Vermeulen, H.J.; Groch, M. Short-Term Load Forecasting using Minimalistic Adaptive Neuro Fuzzy Inference Systems. In Proceedings of the 2020 International SAUPEC/RobMech/PRASA Conference, Cape Town, South Africa, 29–31 January 2020; pp. 1–6.
29. Liu, S.; Gu, S.; Bao, T. An Automatic Forecasting Method for Time Series. *Chin. J. Electron.* **2017**, *26*, 445–452. [CrossRef]
30. Panapongpakorn, T.; Banjerdpongchai, D. Short-Term Load Forecast for Energy Management Systems Using Time Series Analysis and Neural Network Method with Average True Range. In Proceedings of the 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP), Bangkok, Thailand, 16–18 January 2019; pp. 86–89.
31. Di, S. Power System Short Term Load Forecasting Based on Weather Factors. In Proceedings of the 2020 3rd World Conference on Mechanical Engineering and Intelligent Manufacturing (WCMEIM), Shanghai, China, 4–6 December 2020; pp. 694–698.
32. 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]
33. McIlvenna, A.; Herron, A.; Hambrick, J.; Ollis, B.; Ostrowski, J. Reducing the computational burden of a microgrid energy management system. *Comput. Ind. Eng.* **2020**, *143*, 106384. [CrossRef]
34. Weimar, M.; Somani, A.; Etingov, P.; Miller, L.; Makarov, Y.; Loutan, C.; Katzenstein, W. Benefit Cost Analysis of Improved Forecasting for Day-Ahead Hourly Regulation Requirements. In Proceedings of the 2018 IEEE Power Energy Society General Meeting (PESGM), Portland, OR, USA, 5–10 August 2018; pp. 1–5.
35. López, M.; Sans, C.; Valero, S.; Senabre, C. Classification of Special Days in Short-Term Load Forecasting: The Spanish Case Study. *Energies* **2019**, *12*, 1253. [CrossRef]