energies

MDPI

Article

Short-Term Load Forecasting Using Encoder-Decoder WaveNet: Application to the French Grid

Fernando Dorado Rueda [1], Jaime Durán Suárez [1] and Alejandro del Real Torres [2,*]

[1] IDENER, 41300 Seville, Spain; fernando.dorado.rueda@idener.es (F.D.R.); jaime.duran@idener.es (J.D.S.)
[2] Department of Systems and Automation, University of Seville, 41092 Seville, Spain
* Correspondence: adelreal@us.es

Abstract: The prediction of time series data applied to the energy sector (prediction of renewable energy production, forecasting prosumers' consumption/generation, forecast of country-level consumption, etc.) has numerous useful applications. Nevertheless, the complexity and non-linear behaviour associated with such kind of energy systems hinder the development of accurate algorithms. In such a context, this paper investigates the use of a state-of-art deep learning architecture in order to perform precise load demand forecasting 24-h-ahead in the whole country of France using RTE data. To this end, the authors propose an encoder-decoder architecture inspired by WaveNet, a deep generative model initially designed by Google DeepMind for raw audio waveforms. WaveNet uses dilated causal convolutions and skip-connection to utilise long-term information. This kind of novel ML architecture presents different advantages regarding other statistical algorithms. On the one hand, the proposed deep learning model's training process can be parallelized in GPUs, which is an advantage in terms of training times compared to recurrent networks. On the other hand, the model prevents degradations problems (explosions and vanishing gradients) due to the residual connections. In addition, this model can learn from an input sequence to produce a forecast sequence in a one-shot manner. For comparison purposes, a comparative analysis between the most performing state-of-art deep learning models and traditional statistical approaches is presented: Autoregressive-Integrated Moving Average (ARIMA), Long-Short-Term-Memory, Gated-Recurrent-Unit (GRU), Multi-Layer Perceptron (MLP), causal 1D-Convolutional Neural Networks (1D-CNN) and ConvLSTM (Encoder-Decoder). The values of the evaluation indicators reveal that WaveNet exhibits superior performance in both forecasting accuracy and robustness.

Keywords: time series forecasting; energy consumption forecasting; deep learning; machine learning; convolutional neural networks; artificial neural networks; causal convolutions; dilated convolutions; encoder-decoder

Citation: Dorado Rueda, F.; Durán Suárez, J.; del Real Torres, A. Short-Term Load Forecasting Using Encoder-Decoder WaveNet: Application to the French Grid. *Energies* **2021**, *14*, 2524. https://doi.org/10.3390/en14092524

Academic Editor: Fernando Morgado-Dias

Received: 31 March 2021
Accepted: 25 April 2021
Published: 28 April 2021

1. Introduction

An accurate load forecasting is the basis of energy investment planning. It plays a vital role in the decision making and operation of the energy market. While the overestimation of energy consumption leads to wasted financial resources, the underestimation causes power outages and failures throughout the electrical grid. Authors in [1] conclude that a 1% increase in load forecasting error can increase about a 10 $ million in annual operation costs.

Given the previous context, the authors aim to introduce and propose a novel Deep-Learning algorithm in order to improve the current results achieved by other state-of-art solutions. Specifically, an encoder-decoder WaveNet model, usually applied to othertime-series forecasting fields, is proposed. To such an end, this paper is focused on demonstrating the performance of the proposed approach on a macroscopic level (French National Consumption) of power forecasting in the short term (one day ahead).

Short-Term load forecasting is attracting widespread interest among researchers due to its increased importance in smart-grids and micro-grids [2]. Classical short-term time series

forecasting includes a number of techniques to create parameterised models of different types of stochastic processes, whose performance is highly sensitive to the forecast horizon and the quality of the available data. H. Al-Hamadi et al. proposes the use of Kalman filter for short-term load forecasting to a forecast horizon of 24 h ahead [3]. P. Vähäkyla et al. presented a Box-Jenkins time series analysis using Auto-regressive Integrated Moving Average (ARIMA) model fort load short-term forecasting [4]. The presented method led to a simple, fast and accurate forecasting algorithm. P. Sen et al. proposes an ARIMA model for forecasting energy consumption and GHG emissions in Indian [5]. In [6], the usage of other statistical methods (such as linear regression or Seasonal Auto-Regressive Moving Averaged (SARIMA) have also been contemplated and studied. Some authors report the limitations of ARIMA deling with outliers and long-terms forecast horizons [7].

In previous years, Artificial Intelligence (AI) methods have received considerable attention from researches because of their powerful modelling abilities. As a result of this growing expectation, different traditional Machine-Learning algorithms (e.g., Feed-Forward Networks (FFNs), Support Vector Machines (SVM), Random Forest, Multi-Layer Perceptrons (MLP), etc.) have been widely investigated in the last decade within the energy load forecasting field. However, although the main outcomes of such studies were promising, it was found that one of the major drawbacks of using these methods was their low performance when managing time-series problems. On the one hand, the network treats the data as a bunch of data without any specific indication of time. On the other hand, the traditional ML algorithms suffer when processing long sequences which results on degradation problems in the network (e.g., vanishing or exploding gradients) [8]. Consequently, the research in traditional Machine-Learning algorithms remained relatively limited because of the aforementioned bottlenecks produced when using large datasets [9].

In such a context, the search for alternative and more powerful methods to address the energy load forecasting problems has become crucial. In addition, the latest advancements in computing resources (e.g., GPUs and TPUs) have paved the way for the implementation of Deep-Learning methods in order to outperform the results obtained by the traditional algorithms. A clear evidence of such growing expectation about the applications of AI algorithms can be found in the huge impact that Deep Neural Networks (DNN), including Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) or Transformers, have recently had in multiple fields (computer vision, natural language processing, etc.). An example of a successful application could be GPT-3 (an autoregressive language model with 175 billion parameters based on Transformers), which is able to perform multiple NLP-related tasks [10]. These recurrent models can be understood as Feed-Forward networks whose architecture has changed between one sequence to another and their main advantage is that they have specifically been designed for working with sequence problems. However, the use of deep learning methods for energy load forecasting has remained relatively limited and, only in recent years, the interest in such application is growing [9].

As a proof of this interest, the bibliography reports several state-of-the-art Deep-Learning approaches related to the field of energy demand forecasting. For instance, Zeng. et al. proposes a back-propagation neural network for energy consumption prediction, which reported better performance with respect to other methods [11]. H. Hu. et al. presented DeepESN for forecasting energy consumption and wind power generation by introducing the deep learning framework into the basic echo state network [12]. DeepESN shows better results than Back-Propagation Neural-Networks (BPNN) and Echo State Networks (ESN).

H. Li et al. combine a CNN, LSTM, and GRU algorithms to ultra-short-term load forecasting using historical data [13]. The major defect in this study is that the significant hyperameter's combination possibilities to consider to find the best network architecture. Seq2seq approach has been demonstrated to be a robust model in other Deep-Learning fields (e.g., Natural Language Processing (NLP)). Various approaches have been put forward seq2seq in the time series forecasting field, demonstrating promising results. Authors

in [14] present a seq2seq approach based on RNN to make medium-to-long term predictions of electricity consumption in a commercial and residential building at hourly resolution. D. L. Marino et al. investigates in [15] two LSTM based architectures (standard and seq2seq architecture) to predict the electricity consumption from one residential customer. The results show that the seq2seq approach performed well in a minutely and hourly forecast, while LSTM shows a poor performance at one-minute resolution data. Convolutional neural networks were employed in time series forecasting in [16] and time series classification in [17]. However, thanks to the advances in Natural Language Processing (NLP) field has appeared novel networks that could be successfully applied to the time series forecasting field, e.g., Temporal Dilated Causal Convolutional Neural Networks (TDCCNN). The causality refers to there is no information "leakage" from the future to the past, and the output sequence has the same length as the input sequence. In the case that it desired to forecast an output length different from an input length using this architecture, it is needed to implement a seq2seq approach. Dilated convolutions increase the receptive field by uniformly sampling from the receptive fields the examples, helping in dealing with data sparsity, long-term sequence relationships, and multi-resolution [18]. Using the encoder and decoder to enhance the performance is now a key trend in the designing process in CNN architecture [19].

In the last few years, the use of dilated temporal convolutional networks as forecast model has been increasing in the field of time series forecasting for numerous applications. However, the energy sector applications of such technology are still barely studied. R. Wan et al. presents in 2019 a novel method based on Temporal Convolution Network (TCN) for time series forecasting, improving the performance of LSTM and ConvLSTM in different datasets. S. Rizvi et al. presented in 2019 a time series forecasting method of Petrol Retail using Dilated Causal Convolutional Networks (DCCCN), outperforming the results by LSTM, GRU, and ARIMA [18]. O. Yazdanbakhsh et al. propose in [20] a dilated convolutional neural network for time series classification, showing that the used architecture can be effective as a model working with hand-crafted features such as spectrogram and statistical features. Yan et al. presented in 2020 a TCN model for weather predictions task [21]. In this research, a comparative experiment was carried out between LSTM and TCN, where TCN outperforms LSTM using various time-series datasets.

Nevertheless, recent studies reveal promising results using WaveNets in the field of time series forecasting, where such novel ML algorithms are improving the results obtained by the aforementioned Deep Learning methods. WaveNet is a complex deep convolutional architecture based on dilated temporal convolution and skip-connections. A. Borovykh et al. presented in [22] an adaptation of WaveNet for multiples time series forecasting problems. WaveNet consistently outperforms the autoregressive model and LSTM in metrics errors and time in the training process. D. Impedovo presents TrafficWave, a generative Deep Learning architecture inspired by Google DeepMind' Wavenet network for traffic flow prediction [23]. The proposed approach shows better results than other state-of-the-art techniques (LSTM, GRU, AutoEncoders). S. Pramono et al. presented a novel method for short-term load forecasting that consists of a combination of dilated causal residual convolutional neural network and LSTM to forecast four hours-ahead [24]. The long sequences are fed to the convolutional model while the short data sequences are fed to the LSTM layer.

Given the previous context and based on the promising results achieved in other areas, the authors propose to apply these novel techniques to the energy load forecasting field by developing a cutting-edge Deep-Learning architecture, for a forecast horizon of 24 h-ahead, with the main objective of outperforming other state-of-art methods. Specifically, the proposed architecture consists of an Encoder-Decoder approach using WaveNets, which are based on dilated temporal convolutions. The list of advantages that this kind of algorithm presents can be found below:

- There is no "leakage"from the future to the past.

- Employing dilated convolutions enable an exponentially large receptive field to process long input sequences.
- As a result of skip-connections, the proposed approach avoids degradation problems (explosions and vanishing gradients) when the depth of the networks increases.
- Parallelisation on GPU is also possible thanks to the use of convolutions instead of recurrent units.
- Potential performance improvement.

The rest of this document is organised as follows: Section 2 details the elements that comprises the proposed method, data preparation and the steps performed during the training stage. Section 3 presents the results of the proposed approach and comparison methods. Section 4 concludes this research work with future research directions.

2. Materials and Methods

2.1. Proposed Approach: Encoder-Decoder Approach Using WaveNets

WaveNet (Figure 1) is a deep generative model originally proposed by DeepMind which was designed for generating raw audio waveforms [25]. The WaveNet model can be extrapolated beyond the audio in order to be applied to any other type of time series forecasting problem, providing an excellent structure for capturing the long-term dependencies without an excessive number of parameters.

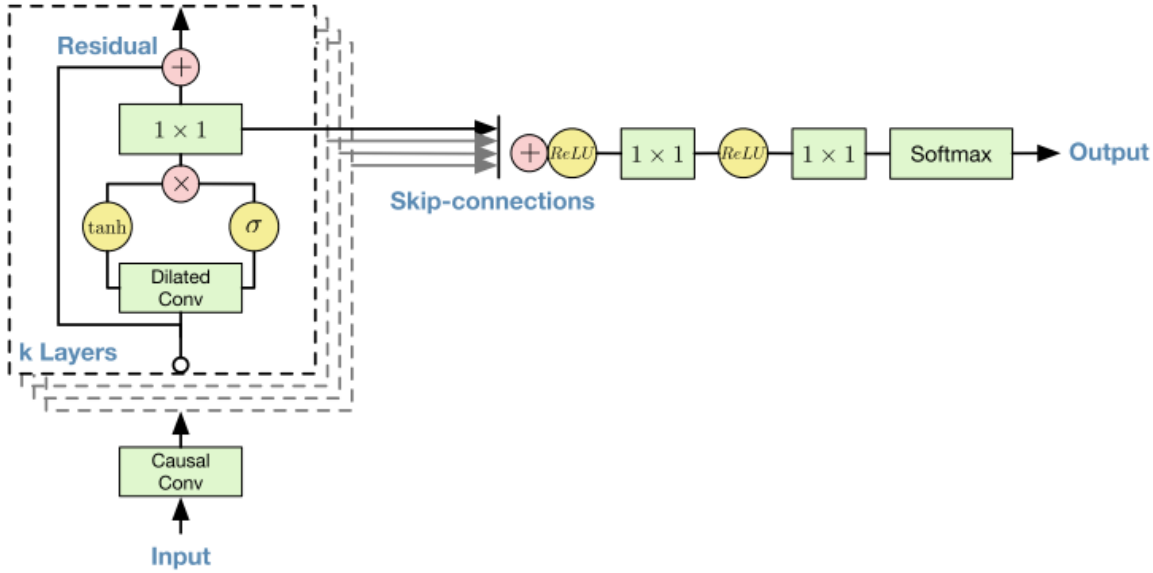

Figure 1. Original WaveNet architecture presented by DeepMind in [26].

WaveNet contains stacks of temporal dilated causal convolutions (TDCCN) that allow it to access a broad range of history of past values when forecasting; By using causal convolution networks (TCNs), the model cannot violate the order in which the data is fed to the algorithm. One of the problems of the basic design of TCNs is that they required many layers or larges filters to process long sequences. To address this problem, the authors of WaveNets suggest the use of a dilated causal convolutional network (DCCN) in combination with causal convolutions [25]. A dilated convolution is a convolution where the filter is applied over an area large than its length by skipping input values with a certain step. This network architecture effectively allows the network to operate on a coaster scale than with traditional convolutions [25]. One of the advantages of dilated convolutions is that they reduce the number of deep neural networks' parameters. Furthermore, it enables a faster training process and, consequently, lower hardware and energy consumption derived cost. ReLU is used as activation function applying multiple convolutional filters in parallel. The dilated convolution splits the input into two different branches which are recombined later via element-wise multiplication. This process depicts a gated activation

unit, where the *tanh* activation branch acts as the learning filter and the *sigmoid* activation branch works as a learning gate that controls the information flow through the network. It is a very similar process that the one explained above for the internal gates of the RNNs. Some of the biggest limitations of deep neural networks are degradations problems (exploding/vanishing gradient) during training process. This occurs as a consequence of either too small or too large derivatives in the first layers of the network, causing the gradient to increase or decrease exponentially, which leads to a rapidly declining performance of the network. In order to address the degradation problem, WaveNets make use of residual connections between layers. This type of deep neural network is based on short-cut connections, which turn the network into its counterpart residual versions. Two options could be considered in Residual Networks (ResNets) [26]: adding new layers would not impact on model's performance (these layers would skip over them if those layers were not useful) and if the new layers were useful, the layer's weights would be non-zero, and the model's performance could increase slightly.

Some modifications were performed in order to adapt the architecture, so the solution is able to generate predictions for the desired forecast horizon and reduce the noise at the output. Originally, WaveNet was trained using the next step prediction, so the errors were accumulated as the model produces a long sequence in the absence of conditioning information [27]. The encoder performs dilated temporal convolutions with a filter bank to generate a set of features of the input sequences. These learned features are comprised into a fixed-vector, also known as context-vector. The decoder network generates an output sequence forecast based on the learned feature of the encoder input sequence, allowing us to generate a direct output forecast for the desired forecast horizon. This modification permits the decoder to handle the accumulative noise when producing a long sequence as output. A generic scheme of the proposed architecture is as follows (Figure 2):

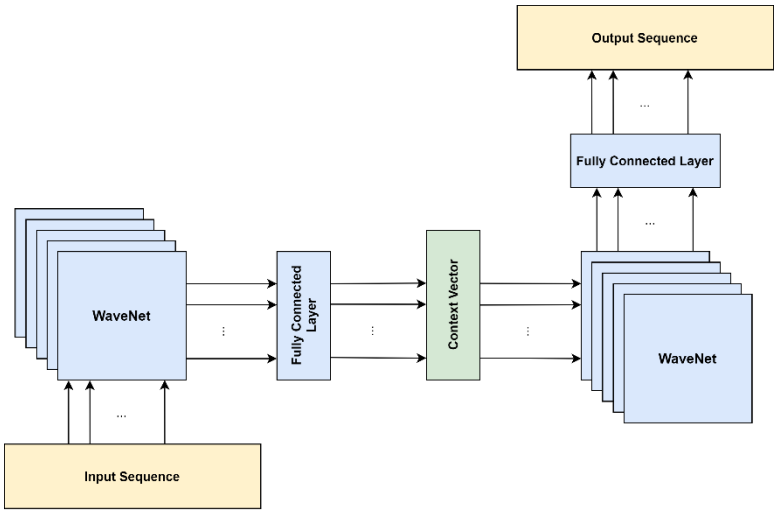

Figure 2. Generic scheme of the proposed approach.

2.2. Data Analysis

As mentioned above, the main objective of the proposed approach is to forecast the power consumption a day ahead by using historical past values. That is, both the input and the output are the same parameter: the power consumption. To this end, macroscopic power consumption data of French has been downloaded from the official RTE website [28]. The data uses for this problem ranges from 2016 to 2020. The data analysis and visualisation were performed by using Pandas, Numpy, and Matploblib Python's libraries. The first visualisation of the available data is illustrated in Figure 3, which shows a seasonal pattern on power consumption.

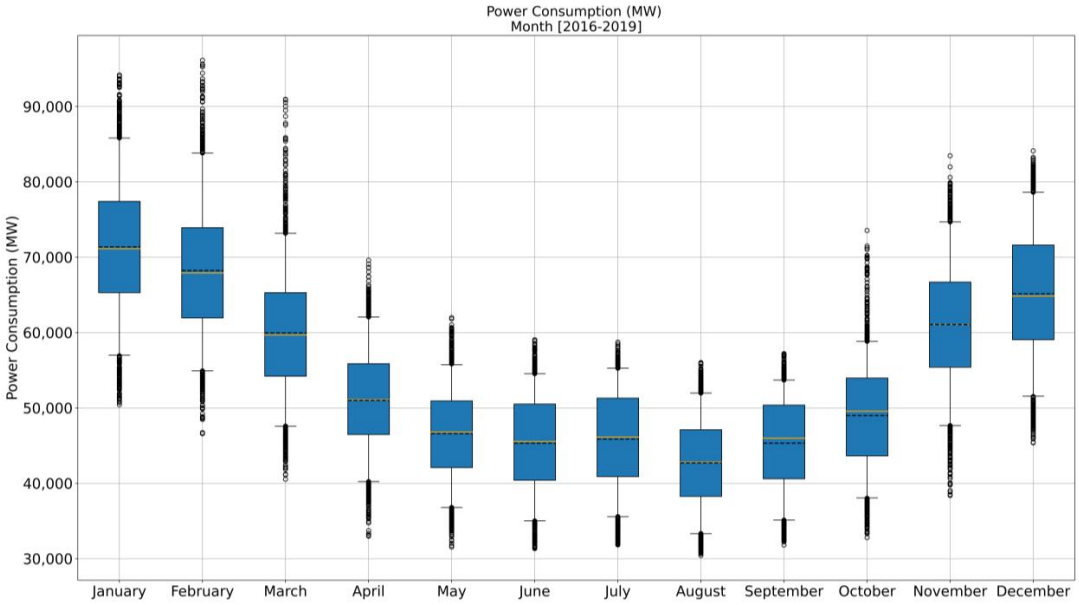

Figure 3. Monthly French power consumption for the period 2016–2019. The blue box indicates the Interquartile Range (IQR), while the lines to the ends correspond with the 95th and 5th percentile, respectively. The coloured lines within the IQR represent the mean (dashed black) line and the median (orange) line.

2.3. Data Preparation

The process of data preparation is a key aspect when training Deep-Learning models since the quality of data affects many parameters: it reduces the modelling errors, speeds up the training process, and simplifies the whole system [29]. In order to carry out the data preparation process for the proposed case, different steps have been carried out, as detailed below.

First, it is interesting to pre-process the data in order to delete outliers, clean unwanted characters and filter null values. As a result of this analysis, it was found that there are no outliers nor any NaN value present in the RTE data, which indicates that RTE provides the data after processing it. Regarding the size of the data, there are 26.280 rows (8.700 elements per year), but it is needed to transform this data into a sequence format.

It is also important to stress that the input and output data sequences have different dimensions. While the input data has a horizon of one week in the past, the prediction horizon for the output data is 24 h ahead. Thus, the dimensions for each one depend on the following parameters: [number of time steps, sequence length, number of features].

In addition, it is well known that machine learning algorithms usually benefit from scaling all their inputs variables to the same range, e.g., normalise all variables into the range [0, 1]. As a result, the network will have a more stable and quick learning process. Besides, it is useful to avoid floating-point number precision due to the different scales of input data. On a time series forecasting problem, where a numerical sequence must be predicted, it can also help to perform a transformation on the target variable. The method that has been used to normalise the data is known as MinMax normalisation:

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Different statistical parameters were calculated to obtain more information about the given data. The data shows a Skewness of 0.53 that indicates that the distribution has more weight in the left tail of the distribution. Some authors suggested in [30] that better predictions are achieved if the target data is linearised through a logarithmic transform. While normalising the target data does not affect the model performance, it helps to reduce their Skewness. In the proposed approach, a log-transformation has been applied to the

target data. The Skewness of the dataset after the log-transformation is 0.08, which indicates that the data is now much closer to a normal distribution.

In order to train the proposed model, a training, a validation and a testing set have been extracted from the original data. On the one hand, the training set is the sample of data used to fit (adjust the weigths) the model. On the other hand, the validation set is the sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyperparameters. Finally, the testing set is the sample of data used to provide an unbiased evaluation of the final model configuration.

The problem under modelling model consists of a time series problem. Accordingly, the data must be divided sequentially. Taking this criterion, the test set only provides the error metrics that correspond with a number of weeks in a year. It does not provide robust metrics to make conclusions about the model's performance. Thus, different training sessions for each data division have been performed, each with different training, validation, and test sets corresponding with different seasons for each year, in order to obtain representative and reliable metrics. The starting and end date for each training session are presented below. Figures 4–6 shows the power consumption in France for data division 1,2,3, respectively.

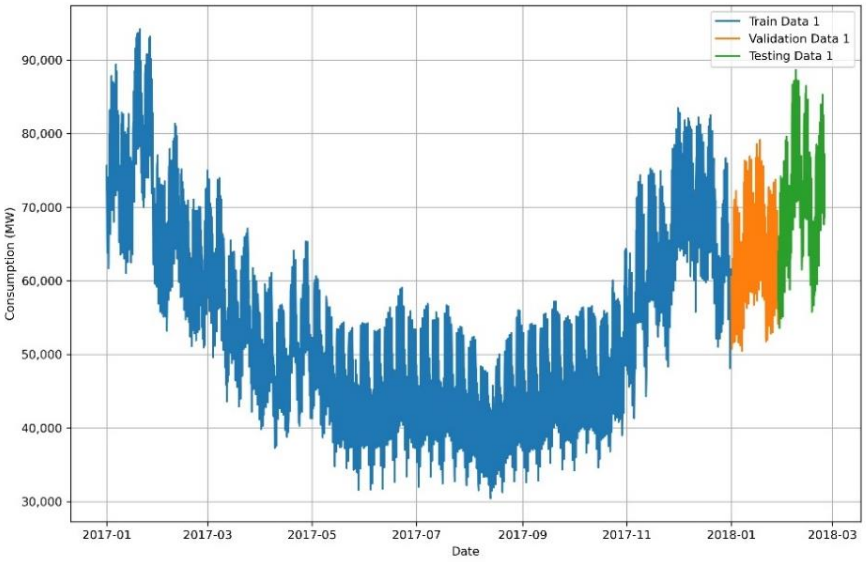

Figure 4. Power consumption for the data division 1.

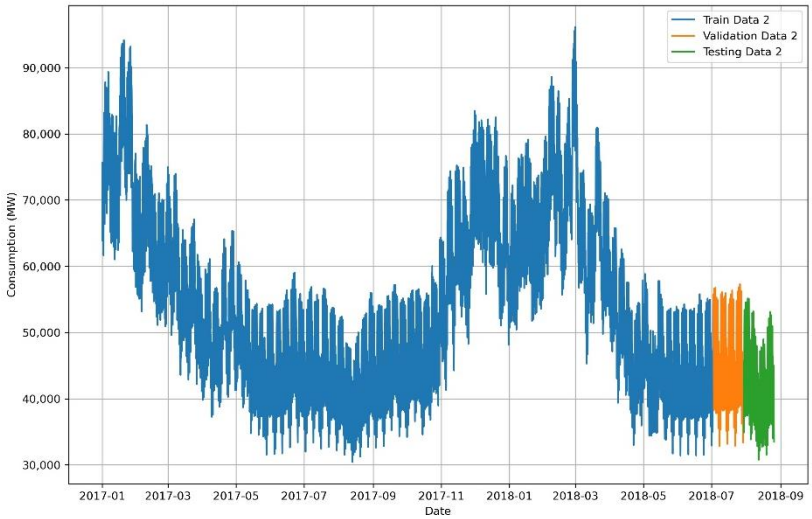

Figure 5. Power consumption for the data division 2.

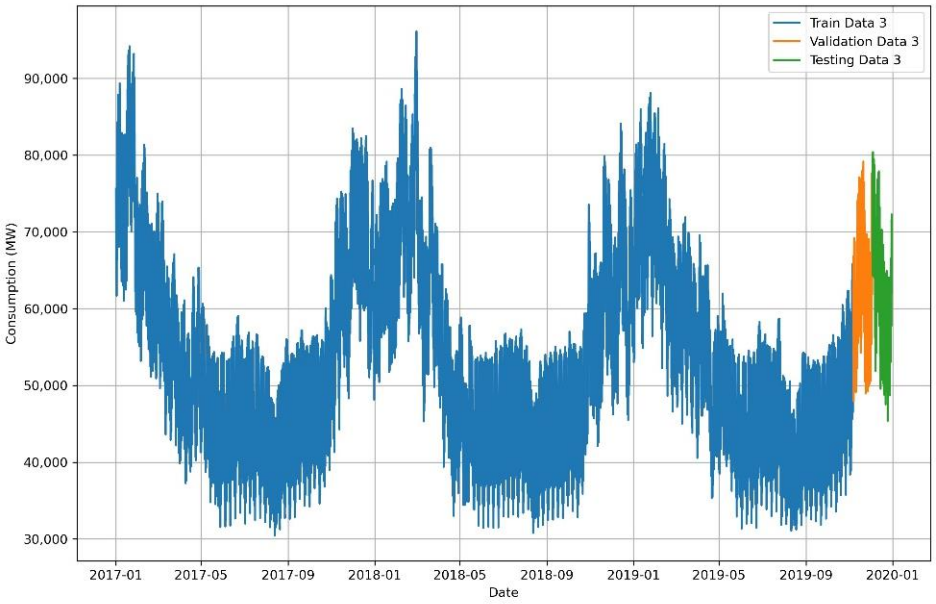

Figure 6. Power consumption for the data division 3.

- Data division 1:
  - Training set: 2017-01-01 00:00:00 to 2017-12-31 23:00:00.
  - Validation set: 2018-01-01 00:00:00 to 2018-01-28 07:00:00.
  - Testing set: 2018-01-28 08:00:00 to 2018-02-24 15:00:00.
- Data division 2:
  - Training set: from 2017-01-01 00:00:00 to 2018-07-02 11:00:00.
  - Validation set: from 2018-07-02 12:00:00 to 2018-07-29 19:00:00.
  - Testing set: from 2018-07-29 20:00:00 to 2018-08-26 03:00:00.
- Data division 3:
  - Training set: from 2017-01-01 00:00:00 to 2019-11-06 07:00:00.
  - Validation set: from 2019-11-06 08:00:00 to 2019-12-03 15:00:00.
  - Testing set: from 2019-12-03 16:00:00 to 2019-12-31 23:00:00.

2.4. Deep Learning Architecture

Design of the Architecture

A grid search process was performed in different training sessions to find the optimal hyperparameter's configuration (Table 1). To such an end, several network architecture configurations were trained using the same data division for training, validation, and testing. These experiments were performed using an i7-8770 processor with 16 GB RAM and Nvidia 1080 RTX with Nvidia CUDA drivers and Tensorflow GPU (2.2.3) as Deep-Learning library. As reported in [31], the use of GPU to train models can be up to 4 train faster than CPU. For this reason, the different training stages for the proposed approach and the comparison techniques have been carried out on GPU Tensorflow is an open-source software library developed by Google brain team for numerical computation [32]. Different models were trained repeatedly, changing the learning parameters (as the learning rate) to find the optimal ones. The following table shows the different tests that have been done to find the best hyper-parameters configuration.

Table 1. Tested hyperparameter's combination. The highlight parameters are the parameters combined with the lowest error in the validation set.

Input Length	Learning Rate	Epochs	Batch Size	Optimiser	Dilation Rates	Kernel Sizes	Filters	Fully Connected Layer's Neurons
168	0.01	100	8	RMSprop	[1,2,4,8]	**2**	12	16
256	**0.001**	150	16	**Adam**	**[1,2,4,8,16]**	4	**32**	**32**
	0.0001	200	**32**		[1,2,4,8,16,32,64,64,128]		64	64
		1000			[1,2,4,8,16,32,64,64,128,128,256]		128	

It is important to remark that the dilation rates exponentially increase as the depth n of the network increases.

$$d_{rates} = 2^i,\ 1 \leq i \leq n \tag{2}$$

where i represents the *i-th* layer and n represents the total number of dilated causal convolution layers.

Both Encoder and Decoder share the same hyperparameters configuration. Nevertheless, while the encoder transforms the past time series values into a feature vector, the decoder uses the feature vector as an input to generate an output sequence for 24 h-ahead. Once all the results were obtained, the objective was to find the model with the least bias error (error in the training set) and a low validation error. Accordingly, the selected model has the following configuration:

- Input length: 168 h
- Learning rate: 0.001
- Epochs: 1000
- Batch size: 32
- Optimiser: Adam
- Dilation Rates: [1,2,4,8,16]
- Kernel sizes: 2
- Number of filters: 32
- Fully connected neurons: 32

In Section 3, the performance of the previous configuration with the testing set is presented and analysed.

2.4.1. Training

The training process consists of learning the best values of the weights and biases in the training set, so the structure was able to correlate its output sequence with respect to the input sequence. The difference between a Deep-Learning time series approach from a traditional one is the handling of the generalisation error. This error is defined as the error when new input sequences are fed into the trained model. Typically, the usual approach consists of estimating the generalisation error by the metric error in the validation set which is independent of the training set. Two facts determine how well the deep learning model performs: its capabilities to reduce the training error to the lowest level and to keep the gap between the training error and the validation error as smaller as possible.

The rest of the training parameters were selected as follows:

- β1 parameter: 0.9. The exponential decay rate for the first moment estimates (momentum).
- β2 parameter: 0.99. The exponential decay rate for the first moment estimates (RMSprop).
- Loss function: Mean Squared Error (MSE).

The learning curves on the three data division are presented below.

When analysing the learning curves (Figure 7), it can be appreciated that the error of the proposed solution is slightly higher in validation sets 1 and 3. These datasets correspond to the winter months, where the error is usually a bit higher as can be observed in the results presented below. Regarding the learning curve corresponding to data division 2,

which elapses the summer months, the validation error is even smaller than the training error, since it encompasses a lot of data combining several months.

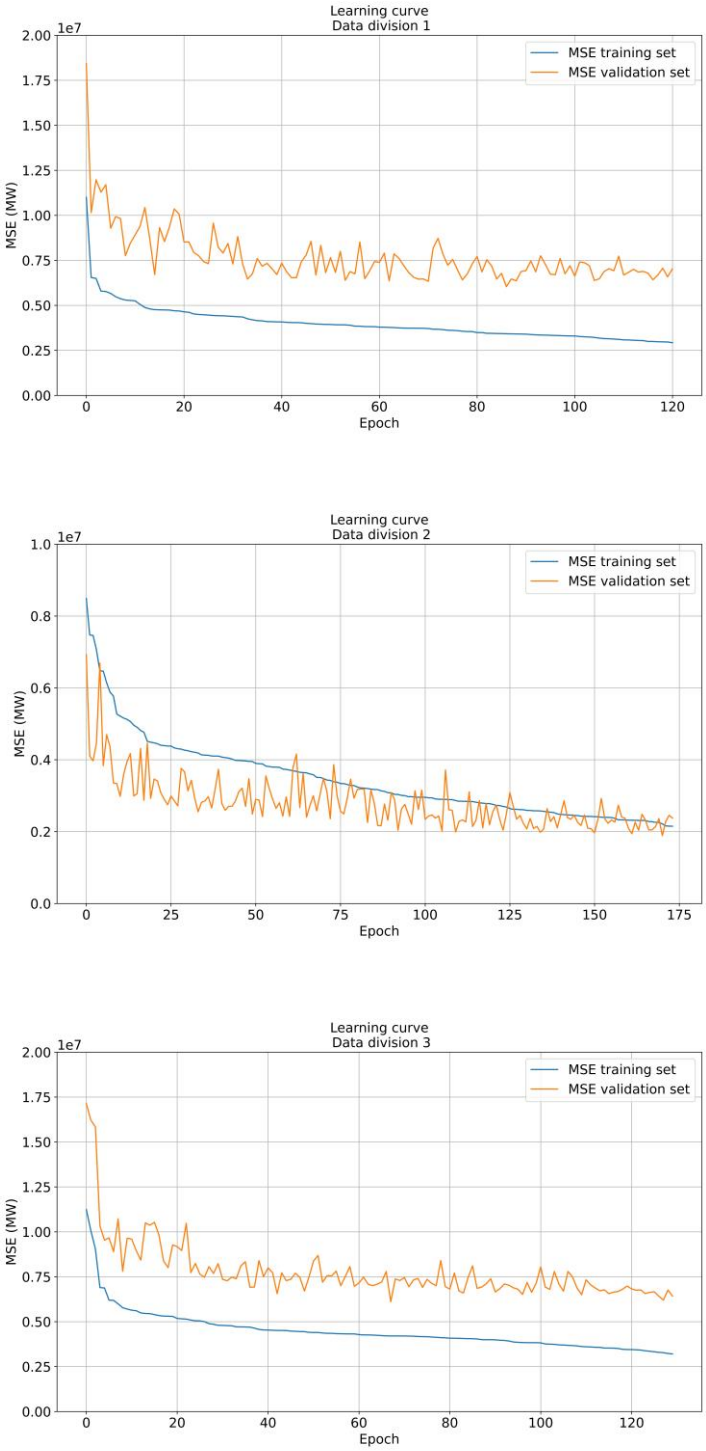

Figure 7. Learning curves for each data division.

3. Results

Different error metrics were computed in order to evaluate the model's performance for a forecast horizon of 24 h ahead. These metrics have been calculated by comparing the actual power consumed in France in MW with the power predicted by each of the proposed models. Training, validation, and testing sets have been extracted from the three data partition divisions. Specifically, Mean Squared Error (MSE), Root Mean Squared Error

(RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), Mean Bias Error (MBE), Mean Bias Percentage Error (MBPE) were calculated for each trained model.

$$\text{MSE (MW)} = \frac{1}{n}\sum|y - \hat{y}|^2 \tag{3}$$

$$\text{RMSE (MW)} = \sqrt{\frac{1}{n}\sum|y - \hat{y}|^2} \tag{4}$$

$$\text{MAE (MW)} = \frac{1}{n}\sum|y - \hat{y}| \tag{5}$$

$$\text{MAPE (\%)} = \frac{100}{n}\sum\left|\frac{y - \hat{y}}{y}\right| \tag{6}$$

$$\text{MBE (MW)} = \frac{1}{n}\sum y - \hat{y}, \tag{7}$$

$$\text{MBPE (\%)} = \frac{100}{n}\sum\frac{y - \hat{y}}{y} \tag{8}$$

where y is the real measure of power consumption and $\hat{y}$ is the estimated measurement of power consumption. Table 2 reports the results obtained on the three data division for training, validation, and testing datasets.

Table 2. Performance comparison metrics for each data division in training, validation, testing datasets.

Data Division	Set	MSE (MW)	RMSE (MW)	MAE (MW)	MAPE (%)	MBE (MW)	MBPE (%)
1	Training Set	3.815×10^6	1953.212	1366.707	2.488	−558.947	−1.114
	Validation Set	7.432×10^6	2726.292	1979.027	3.096	−907.611	−1.511
	Testing Set	7.743×10^6	2782.677	2119.104	2.953	79.439	0.004
2	Training Set	2.972×10^6	1724.011	1225.774	2.167	327.089	0.573
	Validation Set	2.487×10^6	1577.155	1084.796	2.338	477.346	0.913
	Testing Set	2.685×10^6	1638.709	1117.471	2.626	18.018	0.061
3	Training Set	3.560×10^6	1889.346	1352.111	2.477	395.462	0.656
	Validation Set	6.717×10^6	2591.814	1743.703	2.785	129.694	0.122
	Testing Set	6.666×10^6	2581.991	1889.944	3.064	185.531	0.219

The results reveal similar error metrics for the first two tests, corresponding for the data division 1 and 2. The difference between the testing set and the training set is most significant for the data division 3, where the testing set corresponds to December's month. This increase in error may be due to multiple things: based on data analysis, it has been observed that the standard deviation of consumption is higher in the winter months (Figure 3. This may be due to multiple public holidays, cold days where heating consumption spikes, etc.

Additionally, the error metrics of state-of-art deep learning architectures and other advanced statistical time-series methods:Autoregressive-Integrated Moving Average (ARIMA) [32–34], Multilayer Perceptron Network (MLP) [35,36], Convolutional Neural Network (CNN) [37,38], Long-Short Term Memory (LSTM) [39–46], Stacked LSTM [47], Gated-Recurrent Unit (GRU) [48], and Stacked-GRU [47,49] are also presented in Table 3 in order to make a fair-comparison between approaches. The criteria to select each of the models is the same as the one previously described for the proposed approach. For each deep-learning architecture, different hyperparameters configuration has been tested. The model that provides the lowest error in each training session's validation set has been selected to calculate the testing set's errors. The results confirmed that the proposed model outperforms the other deep learning techniques, achieving the lowest error in the three data division for a forecast horizon of 24 h-ahead.

Table 3. Error metrics of the proposed models and comparison techniques for each data division in the testing set.

Data Division	Model	MSE (MW)	RMSE (MW)	MAE (MW)	MAPE (%)	MBE (MW)	MBPE (%)
1	Proposed Approach	7.743×10^6	2782.677	2119.104	2.953	79.439	0.004
	ARIMA	2.409×10^8	15,524.138	13,701.994	18.170	13,564.961	17.9248
	MLP	1.050×10^7	3240.606	2542.002	3.517	258.806	0.3205
	CausalConv1D	9.700×10^6	3115.109	2336.889	3.230	413.027	0.5484
	ConvLSTM	9.006×10^6	3001.072	2321.472	3.211	283.711	0.356
	LSTM	4.428×10^7	6654.573	5439.153	7.614	237.031	−0.235
	GRU	4.476×10^7	6690.945	5498.788	7.649	935.558	0.603
	StackedLSTM	4.414×10^7	6644.195	5317.568	7.416	617.804	0.326
	StackedGRU	1.976×10^7	4445.943	3513.181	4.859	779.914	0.842
2	Proposed Approach	2.685×10^6	1638.709	1117.471	2.626	18.018	0.061
	ARIMA	3.584×10^7	5987.222	4974.551	12.352	−1745.694	−5.961
	MLP	3.294×10^6	1815.062	1402.886	3.296	−353.854	−1.031
	CausalConv1D	2.809×10^6	1676.236	1139.331	2.644	−147.862	−0.465
	ConvLSTM	2.814×10^6	1677.679	1233.298	2.870	−246.956	−0.707
	LSTM	3.103×10^7	5570.446	4585.232	11.243	−956.642	−3.908
	GRU	2.793×10^7	5284.928	4316.504	10.554	−726.990	−3.297
	StackedLSTM	8.562×10^6	2926.257	2328.889	5.487	−43.796	−0.556
	StackedGRU	2.967×10^6	5447.697	1775.300	4.183	−524.327	−1.480
3	Proposed Approach	6.666×10^6	2581.991	1889.944	3.064	185.531	0.219
	ARIMA	9.082×10^6	9530.107	7598.771	11.525	5884.41	8.136
	MLP	8.566×10^6	2926.846	2211.715	3.673	−659.786	−1.311
	CausalConv1D	7.472×10^6	2733.633	1895.930	3.076	−637.183	−1.167
	ConvLSTM	6.880×10^6	2623.061	1894.860	3.129	−206.603	−0.437
	LSTM	2.458×10^7	4958.222	3883.119	6.366	−442.154	−1.161
	GRU	3.706×10^7	6087.658	4904.714	8.113	−811.686	−2.036
	StackedLSTM	3.415×10^7	5843.829	4719.721	7.760	−176.051	−1.074
	StackedGRU	9.504×10^6	3082.936	2346.052	3.862	−357.712	−0.769

It is interesting to stress that the CNN-based methods yield better results than recurrent methods, indicating that the causal convolution structure effectively extracts the feature of ever-changing power demand patterns. Contrary to the outcomes achieved by other research carried out in this area, any significant recurrent based model was found to provide the closest results with respect to the convolutional approaches. Only the results from a complex architecture stacking more Gated-Recurrent-Unit (GRU) layers are most comparable to the rest of the methods.

The following images depict the power prediction at instant k for the next 24 h. For better visualisation, only one forecast by day is presented in the subsequent illustrations. Additionally, Table 4 shows a better description of the absolute error's statistical error in each test set.

Table 4. Statistics description of the absolute error in each test set.

Test Set	Mean (%)	Median (%)	5th Percentile (%)	25th Percentile (%)	75th Percentile (%)	95th Percentile (%)
1	2.966	2.642	1.328	1.971	3.732	5.616
2	2.997	2.800	1.438	2.017	3.646	5.588
3	3.239	2.848	1.470	2.181	3.910	6.474

4. Discussion

This research has offered a novel Deep-Learning structure commonly used for audio synthesis applied for time series forecasting. Despite this paper's focus on the French

load forecasting, the proposed approach could be replicated for a wide range of fields that involve time series forecasting.

It was demonstrated that the proposed encoder-decoder WaveNet, based on dilated temporal causal convolutions, exhibits better results than the other comparison techniques. Our proposed model shows the best performance in different test sets compared to other state-of-art Deep-Learning models in terms of MSE, RMSE, MAE, MAPE, MBE, MBPE. Specifically, Multi-Layer Perceptron Network (MLP) and Causal Convolution 1-D (1D-CNN), Autoregressive-Integrated Moving Average (ARIMA), Long-Short-Term-Memory (LSTM), Gated-Recurrent-Unit (GRU) and some combination of them have been implemented to make a fair comparison between approaches.

The errors were uniformly distributed through the different test sets, which corresponds with different seasons in 2018–2019 (Figure 8). Although the error metrics are uniformly distributed, it can be appreciated that it is slightly higher in test set 3, corresponding to December of 2019. This variation in the error metrics also occurs in the other comparison techniques, probably due to the holidays and non-school period this month. Although the improvements of the proposed model over other approaches may seem to be slight, it is more than enough to improve the energy usage efficiency allowing significant savings inmoney and resources since, as D. Bunn et al. claim in [1], a 1% of improvement of an electricity forecasting model can result in up to 10M £ of savings.

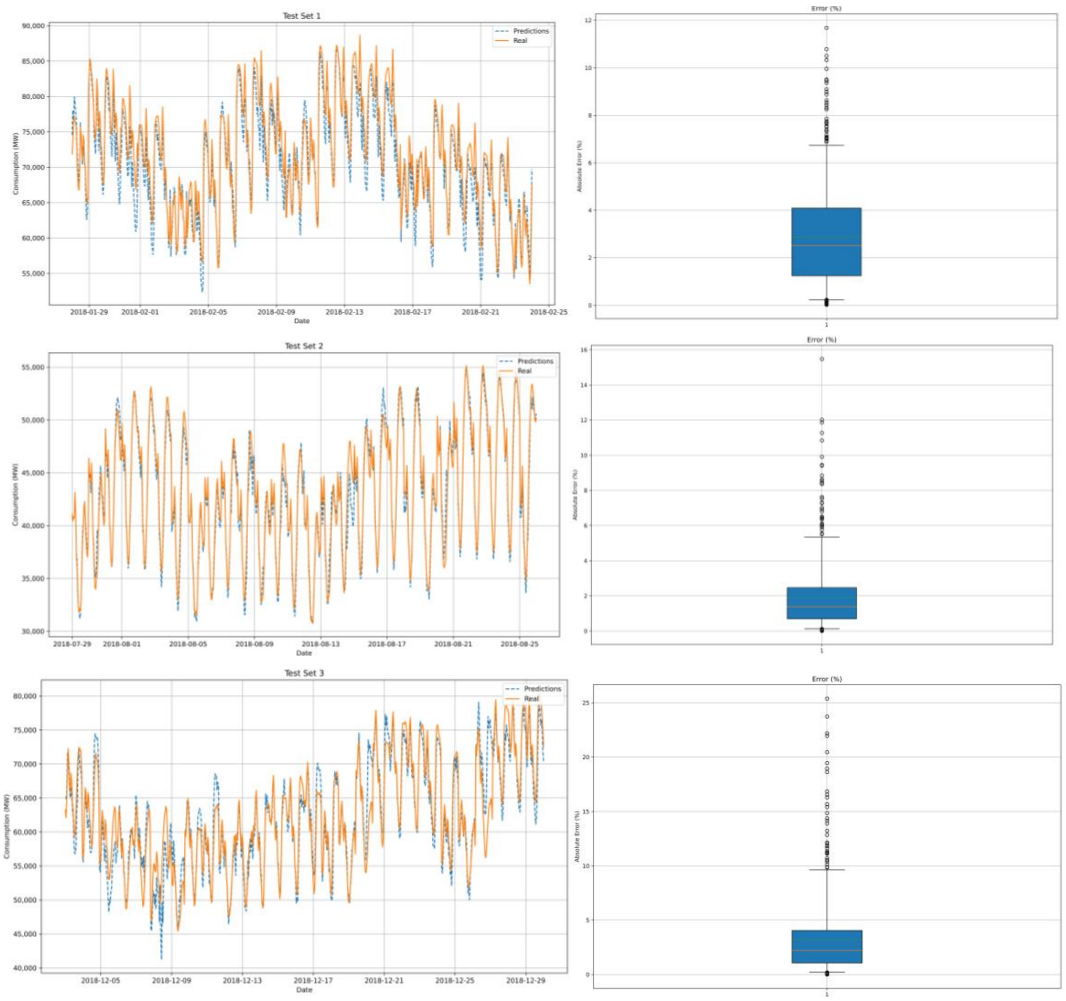

Figure 8. Performance of the power foecasting model on the three different test sets. (**Left** column) Real and prediction of the power consumption in the three different testing sets. (**Right** column) Box-Plot of the absolute error in each test set. The coloured lines within the box represent the mean (dashed green) line and the median (orange) line.

The performance achieved in this research represents promising results for those researches interested in time series forecasting. This study proved the good learning capabilities of dilated temporal convolutions networks. One of the significant advantages of the proposed method, respectively, traditional recurrent methods, is the possibility of parallelising the training process in GPUs, enabling the training process in less time even if the amount of data increases a lot.

Further research will focus on considering multiples sources of data to improve the proposed network's performance, e.g., weather data and calendar information. Additionally, it can be helpful a better adjustment of the hyperparameter configuration, performing more training sessions. Another possible aspect is ensemble forecasting, where the final prediction forecast will be a linear or non-linear combination of the predictions from different models. A pre-processing stage could be further added to decompose the input sequence, e.g., Wavelet Transforms (WT) or Empirical Mode Decomposition (EMD), whereas each decomposed sequence will be treated as a new feature of the model. Another possible neural network modification consists of replacing the context vector with an attention mechanism to focus on the most important parts in the input sequence with respect to the output sequence.

Anyone interested in the code can write an email to the addresses of the first page in order to be provided with the code.

Author Contributions: Conceptualisation, F.D.R.; software, F.D.R.; validation, A.d.R.T. and J.D.S.; writing—original draft preparation, F.D.R.; writing—review and editing, A.d.R.T. and J.D.S.; supervision, A.d.R.T. and J.D.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data have been extracted from the official RTE website [27]. The data is available for research purposes free of charge.

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

References

1. Bunn, D.; Farmer, E. *Comparative Models for Electrical Load Forecasting*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 1985.
2. Sadaei, H.; de Lima Silva, P.; Guimarães, F.G.; Lee, M.H. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. *Energy* **2019**, *175*, 365–377. [CrossRef]
3. Al-Hamadi, H.M.; Soliman, S.A. Short-term electric load forecasting based on Kalman filtering algorithm with moving window weather and load model. *Electr. Power Syst. Res.* **2004**, *68*, 47–59. [CrossRef]
4. Vähäkyla, P.; Hakonen, E.; Léman, P. Short-term forecasting of grid load using Box-Jenkins techniques. *Int. J. Electr. Power Energy Syst.* **1980**, *2*, 29–34. [CrossRef]
5. Sen, P.; Roy, M.; Pal, P. Application of ARIMA for forecasting energy consumption and GHG emission: A case study of an Indian pig iron manufacturing organization. *Energy* **2016**, *116*, 1031–1038. [CrossRef]
6. Box, G.E.P.; Jenkins, G.M.; Reinsel, G.C. *Time Series Analysis Forecasting and Control*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 2013.
7. Luceño, A.; Peña, D. *Autoregressive Integrated Moving Average (ARIMA) Modeling*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 2007.
8. Brownlee, J. Machine Learning Mastery. 2017. Available online: https://machinelearningmastery.com/handle-long-sequences-long-short-term-memory-recurrent-neural-networks/ (accessed on 16 October 2020).
9. Chitalia, G.; Pipattanasomporn, M.; Garg, V.; Rahman, S. Robust short-term electrical load forecasting framework for commercial buildings using deep recurrent neural networks. *Appl. Energy* **2020**, *278*, 115410. [CrossRef]
10. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyvam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *arXiv* **2020**, arXiv:2005.14165.
11. Zeng, Y.; Zeng, Y.; Choi, B.; Wang, L. Multifactor-influenced energy consumption forecasting using enhanced back-propagation neural network. *Energy* **2017**, *127*, 381–396. [CrossRef]
12. Hu, H.; Wang, L.; Lv, S.X. Forecasting energy consumption and wind power generation usingdeep echo state network. *Renew. Energy* **2020**, *154*, 598–613. [CrossRef]
13. Li, H.; Liu, H.; Ji, H.; Zhang, S.; Li, P. Ultra-Short-Term Load Demand Forecast Model Framework Based on Deep Learning. *Energies* **2020**, *13*, 4900. [CrossRef]

14. Rahman, A.; Srikumar, V.; Smith, A.D. Predicting electricity consumption for commercial and residential buildingsusing deep recurrent neural networks. *Appl. Energy* **2018**, *212*, 372–385. [CrossRef]
15. Marino, D.L.; Amarasinghe, K.; Manic, M. Building Energy Load Forecasting using Deep Neural. *IECON* **2016**. [CrossRef]
16. Mittelman, R. Time-series modeling with undecimated fully convolutional neural networks. *arXiv* **2015**, arXiv:1508.00317.
17. Wang, Z.; Yan, W.; Oates, T. Time series classification from scratch with deep neural networks: A strong baseline. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 14–19 May 2017.
18. Rizvi, S.; Syed, T.; Qureshi, J. Real-Time Forecasting of Petrol Retail using Dilated Causal CNNs. *J. Ambient. Intell. Smart Environ.* **2019**. [CrossRef]
19. Yasrab, R.; Gu, N.; Zhang, X. An Encoder-Decoder Based Convolution Neural. *Appl. Sci.* **2017**, *7*, 312. [CrossRef]
20. Yazdanbakhsh, O.; Dick, S. Multivariate Time Series Classification using Dilated Convolutional Neural Network. *arXiv* **2019**, arXiv:1905.01697.
21. Yan, J.; Mu, L.; Wang, L.; Ranjan, R.; Zomaya, A.Y. Temporal Convolutional Networks for the Advance Prediction of ENSO. *Nature* **2020**, *10*, 8055. [CrossRef] [PubMed]
22. Borovykh, A.; Bohte, S.; Oosterlee, C. Conditional time series forecasting with convolutional neural networks. *arXiv* **2018**, arXiv:1703.04691.
23. Impedovo, D.; Dentamaro, V.; Pirlo, G.; Sarcinella, L. TrafficWave: Generative deep learning architecture for vehicular traffic flow prediction. *Appl. Sci.* **2019**, *9*, 5504. [CrossRef]
24. Pramono, S.H.; Rohmatillah, M.; Maulana, E.; Hasanah, R.N.; Hario, F. Deep Learning-Based Short-Term Load Forecasting. *Energies* **2019**, *12*, 3359. [CrossRef]
25. Oord, A.D.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Kavukcuoglu, K. WaveNet: A Generative Model for Raw Audio. *arXiv* **2016**, arXiv:1609.03499.
26. Raghunandepu. Understanding and Implementation of Residual Networks (ResNets). 2019. Available online: https://medium.com/analytics-vidhya/understanding-and-implementation-of-residual-networks-resnets-b80f9a507b9c (accessed on 5 August 2020).
27. RTE. RTE France. Available online: https://rte-france.com/ (accessed on 6 October 2020).
28. Koval, S.I. Data preparation for neural network data analysis. In Proceedings of the IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus), Moscow and St. Petersburg, Russia, 26–29 January 2018.
29. Scikit-Learning. Available online: https://scikit-learn.org/stable/auto_examples/compose/plot_transformed_target.html#sphx-glr-auto-examples-compose-plot-transformed-target-py (accessed on 7 October 2020).
30. Buber, E.; Diri, B. Performance Analysis and CPU vs GPU Comparison for Deep Learning. In Proceedings of the 2018 6th International Conference on Control Engineering & Information Technology (CEIT), Istanbul, Turkey, 25–27 October 2018.
31. Brain, G. TensorFlow. Available online: https://www.tensorflow.org/ (accessed on 18 October 2020).
32. Box, G.; Jenkins, G.M.; Reinsel, C. *Time Series Analysis: Forecasting and Control*; J. Wiley and Sons Inc: Hoboken, NJ, USA, 1970.
33. Zhang, G.P. Time series forecasting using a hybrid ARIMA and neural network model. In *Neurocomputing*; Elsevier: Amsterdam, The Netherlands, 2003; pp. 159–175.
34. Kotu, V.; Deshpande, B. Time Series Forecasting. In *Data Science*; Morgan Kaupmann: Burligton, MA, USA, 2019; pp. 395–445.
35. Goodfellow, I.; Bengio, Y.; Courville, A. Deep FeedForward Networks. In *Deep Learning*; MIT Press: Cambridge, MA, USA, 2017; pp. 166–228.
36. Sun, J. Feedforward Neural Networks. 2017. Available online: https://www.cc.gatech.edu/~{}san37/post/dlhc-fnn/ (accessed on 17 September 2020).
37. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. In *Neural Computation*; MIT Press: Cambridge, MA, USA, 1989; pp. 541–551.
38. Goodfellow, I.; Bengio, Y.; Courville, A. Convolutional Neural Networks. In *Deep Learning*; MIT Press: Cambridge, MA, USA, 2017; pp. 330–373.
39. Goodfellow, I.; Bengio, Y.; Courville, A. Recurrent Neural Networks. In *Deep Learning*; MIT Press: Cambridge, MA, USA, 2017; pp. 373–425.
40. Rumelhart, D.; Hinton, G.; Williams, R. Learning representations by back-propagation errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
41. Lim, B.; Zohren, S. Time Series Forecasting with Deep Learning: A Survey. *arXiv* **2020**, arXiv:2004.13408.
42. Perez, P.T. Deep Learning: Recurrent Neural Networks. 2018. Available online: https://medium.com/deeplearningbrasilia/deep-learning-recurrent-neural-networks-f9482a24d010 (accessed on 18 September 2020).
43. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
44. Hewanakage, H.; Bergneir, C.; Bandara, K. Recurrent Neural Networks for Time Series Forecasting: Current Status. *arXiv* **2020**, arXiv:1909.00590.
45. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
46. Olah, C. Understanding LSTM Networks. 2015. Available online: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed on 21 September 2020).

47. Cho, K.; Bahdanau, D.; Bougares, F.; Bengio, H.S.Y. Learning Phrase Representations using RNN Encoder–Decoder. *arXiv* **2014**, arXiv:1406.1078.
48. Anderson, R. RNN, Talking about Gated Recurrent Unit. Available online: https://technopremium.com/blog/rnn-talking-about-gated-recurrent-unit/ (accessed on 21 September 2020).
49. Inkawhich, M. Deploy Seq2seq Model. Available online: https://pytorch.org/tutorials/beginner/deploy_seq2seq_hybrid_frontend_tutorial.html (accessed on 21 September 2020).