

Article

Not peer-reviewed version

Modeling Short-Term Energy Consumption Forecasting with External Influences: A Comprehensive Study

[Leonardo Amaral](#)*, [Gustavo Araújo](#), [Ricardo Moraes](#)*, [Francisco Vasques](#), [Paulo Portugal](#)

Posted Date: 21 June 2024

doi: 10.20944/preprints202406.1550.v1

Keywords: electricity forecasting model; deep learning; computational intelligence



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Modeling Short-Term Energy Consumption Forecasting with External Influences: A Comprehensive Study

Leonardo Amaral ¹, Gustavo Araújo ¹, Ricardo Moraes ^{1,*}, Francisco Vasques ² and Paulo Portugal ³

¹ Universidade Federal de Santa Catarina UFSC; leonardo.amaral@unimontes.br (L.A.); gustavo.araujo@ufsc.br (G.A.)

² Faculdade de Engenharia do Porto (FEUP); vasques@fe.up.pt

³ SYSTEC-ARISE, Faculty of Engineering, University of Porto; pportugal@fe.up.pt

* Correspondence: ricardo.moraes@ufsc.br

Abstract: Short-term energy consumption forecasting crucial for the operation of new electrical energy grids, not only due to their own characteristics, but also due to the new elements present in the energy matrix. Many existing models are limited by their methodologies or by considering only a narrow set of factors influencing energy consumption. This study introduces a pipeline for developing an energy forecasting model and evaluates the significance of factors affecting energy consumption and, consequently, forecast accuracy. The study utilizes a dataset from ISO NE (Independent System Operator New England), spanning the total electric load of various cities in New England from January 2017 to December 2019. This dataset comprises 23 independent variables, including weather data, economic indicators, and market information. The results outline the steps involved in constructing energy forecast models using time series analysis. By carefully selecting variables and representing external factors, the study demonstrates the feasibility of generating more accurate predictions with reduced computational resources.

Keywords: electricity forecasting model; deep learning; computational intelligence

1. Introduction

The traditional structure of the Electric Power System is characterized by the presence of large power generation plants (such as hydroelectric and thermal power plants) located far from the consumption areas. In this kind of system, decisions are centralized and the flow of energy is unidirectional, meaning it goes from the operators to the end consumers.

Changes in the energy matrix, driven by a growing proportion of clean and sustainable sources and spurred by the deregulation of the sector, are reshaping the terrain of electric power grid management. We are moving from a scenario with stable predictions and controlled power generation to a more volatile and less predictable scenario, where the flow of energy is bidirectional, traversing from operators to consumers and vice versa [1].

Modern electric grids, referred to as "Smart Grids," represent an evolution designed to address contemporary challenges by leveraging advanced communication and information technologies. These grids encompass not only the transmission of energy from generation to substations but also the distribution of electricity from substations to individual end-users [2].

In this new networks, electrical load forecasting and energy generation forecasting play fundamental roles in maintaining network stability, preventing overloads and to ensure a smooth transition to more ecologically sustainable energy use, all depending on the ability to understand and predict energy supply and demand [3].

To predict energy generation, the procedure proposed in [4] can be utilized. For predicting electrical load demand, the pipeline developed in this study is applicable. The sparse interconnection of alternative energy sources and the loads they supply necessitates distributed control and monitoring of energy flow. Distributed control needs higher availability of information for effective

decision-making [5]. Decision-making could be of two types: a) involving unit commitments with response times of several hours and b) load dispatch decisions with response times ranging from a few minutes to several hours [6]. The pipeline proposed in this study primarily aims to address load dispatch, so the focus is on suitable methods for short-term electric energy prediction.

The main contributions of this paper can be summarized as follows: (1) Emphasizing the growing significance of short-term energy consumption prediction; (2) Offering a comprehensive framework detailing the construction of energy prediction models; (3) Addressing critical considerations in the development of energy consumption prediction models; (4) Evaluating model effectiveness through the analysis of its representation of consumption-affecting factors and the efficacy of relevant prediction methodologies.

The remainder of this paper is organized as follows: Section 2 elaborates on the applicable techniques, implementation challenges, and the proposed prediction model pipeline, while Section 3 outlines the conducted experiments and their results. The concluding section presents the study's findings, followed by the inclusion of bibliographic references.

2. Energy Prediction

2.1. Energy Consumption

Electric energy consumption is subject to a multitude of influencing factors, including user behaviors, weather conditions, seasonal variations, economic indicators, and more [7]. The nonlinear nature of the relationship between exogenous factors and energy consumption add complexity to energy prediction processes.

The complexities of energy prediction stem from the granularity of the demand being forecasted. Predicting aggregate consumer demand proves to be more manageable than individual consumer demand due to stronger correlations with external factors like temperature, humidity, and internal factors such as historical trends. Aggregation mitigates the inherent volatility of individual consumers, resulting in a smoother curve shape and greater homogeneity [8].

In [9] the two situations are compared, one where the prediction of individual users is made and the aggregate prediction is generated from them and another where the prediction of the electrical load is made directly from the aggregate demand.

With regard to reducing energy consumption, it is important to take into account the use of strategies to control the temperature of buildings and reduce costs by taking advantage of surplus photovoltaic energy, as proposed in [10].

2.2. Selection of the prediction method

2.2.1. Prediction Technique based on time series

One of the possible ways to predict energy consumption is through time series analysis. Time series data allows for predicting future values based on past values with a certain margin of error [11].

Traditional techniques for energy demand prediction encompass statistical time series models, such as ARIMA (autoregressive integrated moving average), SARIMA (seasonal ARIMA), and exponential smoothing. For [12], these techniques are benchmarks in the field of energy demand forecasting.

In recent decades, however, computational intelligence methods have gained prominence for better describing the trends of nonlinear systems and taking into account the effects of exogenous factors. As examples, we can mention, according to the same author, the Artificial Neural Network (ANN), the Support Vector Machine (SVM) and others.

Recent works published in the last five years indicate a growing adoption of deep learning techniques for several reasons, including their potential to generate more accurate predictions, advancements in processing capabilities, and the greater availability of information (large datasets). Examples of these techniques include Long Short-Term Memory (LSTM), Convolutional Neural

Network (CNN), as well as combinations thereof. As previously mentioned, while they offer more accurate predictions, they also entail higher computational costs [9, 14, 15, 16].

2.2.2. Deep learning techniques

According to [13], “neural networks represent a subfield of machine learning and deep learning constitutes a subfield of neural networks that has revolutionized various fields of knowledge in recent decades.”. Deep learning employs deep architectures or hierarchical learning methods consisting of multiple layers between the input and output layers, followed by various stages of non-linear processing units. These models are attractive because it explores well the learning characteristics and classifying patterns [17]. Since deep learning was introduced, some disruptive solutions have been developed, showcasing its success in various application domains. Below, the characteristics of the deep learning techniques considered in the study are briefly described.

Artificial Neural Networks (ANNs), or general neural networks, consist of a perceptron that contains few or no internal layers with multiple units (neurons). A Dense Neural Network (DNN) is a perceptron with multiple internal layers capable of modeling complex nonlinear relationships with fewer processing units compared to the general neural network while achieving similar performance.

A Recurrent Neural Network (RNN) is fundamentally different from a conventional neural network because it is a sequence-based model that establishes a relationship between previous information and the current moment. This means that a decision made by the RNN at the current time can impact future time steps.

As a successful RNN model, LSTM (Long Short Term Memory) presents a cell memory that can retain its state over time and non-linear gate units to control the flow of information into and out of the cell [17]. Fig. 1 illustrates these elements.

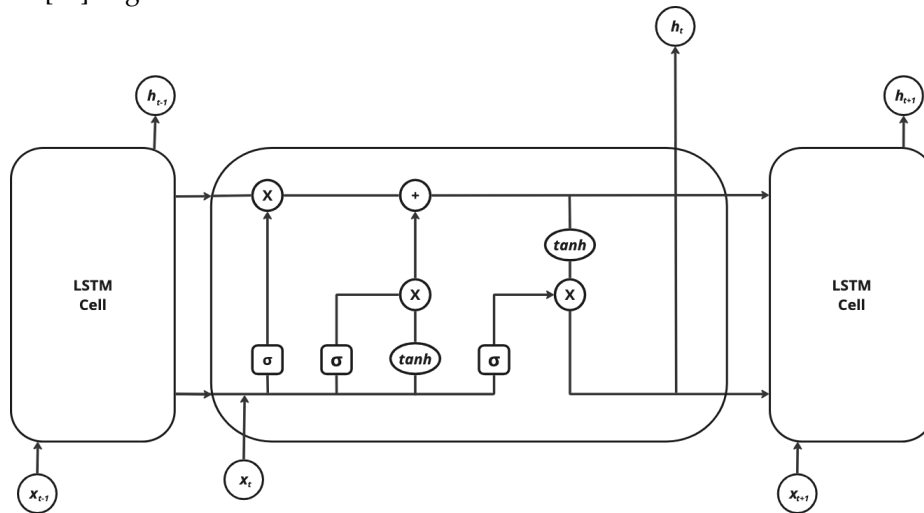


Figure 1. LSTM Model Structure.

(Source: [18])

The operations performed by the elements identified in the previous figure are represented by the equations below [18].

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + b_f), \quad (1)$$

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + b_i), \quad (2)$$

$$\tilde{C}_t = \tanh(W_c \times [h_{t-1}, x_t] + b_c), \quad (3)$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t, \quad (4)$$

$$o_t = \sigma(W_o \times [h_{t-1}, x_t] + b_o), \quad (5)$$

$$h_t = o_t \times \tanh(C_t), \quad (6)$$

where W_i , W_f , W_o e b_i , b_f , b_o are the weights and biases that govern the behavior of the it (input gate), ft (forget gate) e ot (output gate), respectively, and where W_c and b_c are the weights and bias of the \tilde{C}_t (memory cell candidate) and C_t is the cell state. Consider that x_t is the input to the network, h_t is the output of the hidden layer and σ denotes the sigmoidal function.

A Convolutional Neural Network (CNN) is a particular type of neural network that also acts as a feature extractor [17]. This neural network utilizes a combination of convolution and pooling layers applied to input data to extract new features. These features are then used as new inputs and passed to a classic MLP neural network, also known as a fully connected layer. Convolution involves extracting feature maps from each input using a filter with a specific size (typically square filters) and a stride (a left-to-right and top-to-bottom step). The feature maps extracted by the filters (kernels) are subsequently smaller than the input matrix and have different values since each filter is distinct. The feature maps are then used as inputs for the next layer, which can be another convolution layer or a pooling layer; a pooling layer shortens the feature map using a filter with a specific size and stride and the reduction performed by the pooling layer can be either max pooling or average pooling.

After the convolution and pooling layers, the extracted feature maps are flattened and passed to a fully connected layer, as indicated in Fig. 2.

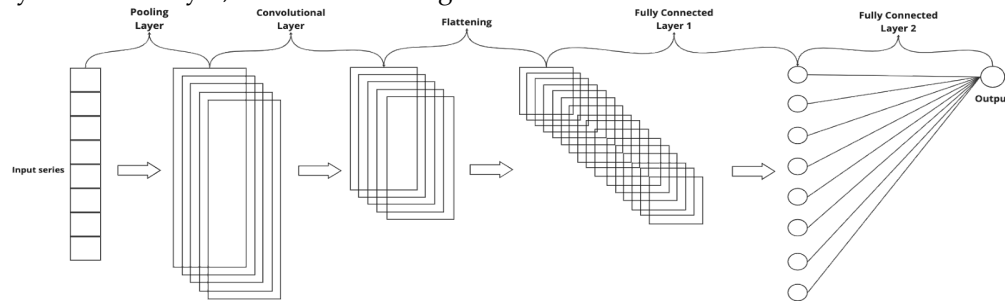


Figure 2. CNN Model Structure. (Source: [17]).

The hybrid model that combines a CNN and an LSTM represents an alternative to enhance prediction accuracy and stability by leveraging the advantages of each. The CNN preprocesses the data and extracts features from the variables that affect consumption, while the LSTM handles modeling temporal information with irregular trends in the time series and uses this information for future estimates. In some situations, a model that employs a single machine learning algorithm may not have as much complexity, but its results can be unrealistic and imprecise [13,15].

2.3. External factors

External factors that affect energy consumption, as mentioned in Section 3.1, are represented by adding or generating new variables derived from the combination of available variables.

2.3.1. Cyclical seasonality

It's intuitive to envision energy consumption curves displaying annual, weekly, and daily cyclic variations. Across the year, consumption fluctuates with the changing seasons. Weekly variations arise from differing consumption patterns on weekdays versus weekends. Similarly, consumption experiences daily fluctuations, often peaking when individuals return home from work.

To replicate these phenomena, [20] introduced several encoding techniques such as binary encoding, linear relationships, and trigonometric functions. These encoding methods serve to distinguish between days of the week and hours of the day, thereby capturing weekly and daily seasonal behaviors. In [21], all these approaches were evaluated, with optimal outcomes observed when using day of the week (dc) coding set to 3, equivalent to employing a 7-bit coding scheme, and time of day (hc) coding set to 2, corresponding to the utilization of trigonometric functions.

2.3.2. Calendar

Energy consumption is influenced by the calendar, which establishes holidays, workdays and weekends [12]. Additionally, the calendar allows us to identify the working hours during which consumption is higher.

Holidays exhibit an atypical behavior in terms of both daily energy consumption and the duration of the consumption curve. The reference for energy consumption on a holiday is the value consumed on the same day of the previous year. For some holiday dates, it is also necessary to consider the induced holiday, which is the bridge day between the holiday and the next weekend. On these days, energy consumption is lower than on workdays.

2.3.3. Thermal discomfort index

According to [23], buildings are responsible for 30% of global final energy consumption and 26% of global energy-related emissions. A significant portion of this energy is used to ensure the thermal comfort of the facilities. Thermal discomfort is related to the field of building ambience studies, which aims to design people's well-being, meaning to provide a bodily state in which there is no sensation of cold or heat. Fig. 3 illustrates the boundaries of a thermal comfort zone.

Within the survival zone, one can discern the temperature thresholds crucial for human survival. These limits encompass hypothermia, characterized by a body temperature below 35°C, and hyperthermia, denoting a significant rise in body temperature. At the outermost edges of the thermal comfort zone lie the critical temperatures, divided into lower and upper categories.

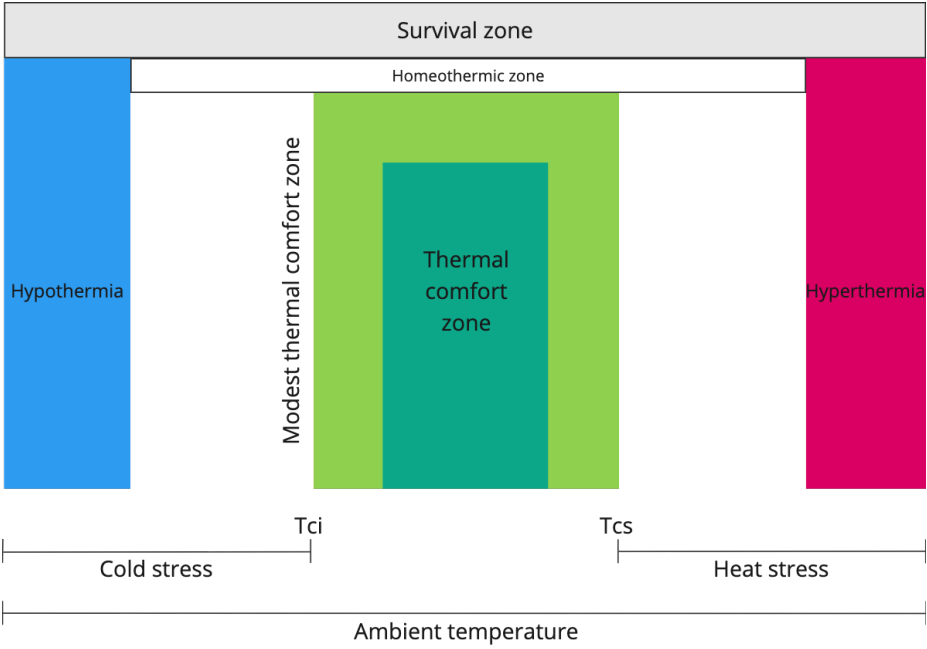


Figure 3. Thermal Comfort Zone.m(Source:[26]).

The method used to obtain the thermal discomfort index is based on the formula adapted by [24] and cited by [25]. In this formula, T_a represents the air temperature in degrees Celsius (°C) and T_o represents the dew point temperature, also in degrees Celsius (°C).

$$I_d = 0,99 \times T_a + 0,36 \times T_o + 41,5, \tag{5}$$

where:

- Id - Discomfort Index
- Ta - Ambient temperature
- To - Dew point temperature

The thermal comfort condition is determined by referencing Table 1, as presented by [26], which provides the calculated discomfort index (I_d).

Table 1. Relationships with Thermal Comfort Conditions.

Interval	Condition
$I_d > 80$	Heat stress
$75 < I_d < 80$	Uncomfortable due to heat
$60 < I_d < 75$	Comfortable
$55 < I_d < 60$	Uncomfortable due to cold
$I_d < 55$	Stress due to cold

When I_d values fall within the extreme ranges, exceeding 75 or dropping below 60, individuals tend to activate air conditioning devices.

2.3.4. Market and economic indicators

Econometric models indicate that Gross Domestic Product (GDP), energy prices, gross production and population influence energy consumption [27].

2.4. Modeling methodology

Essentially, the models utilized in electric prediction employing computational intelligence can be categorized into two types: one treats consumption data purely as a time series, with the impact of exogenous factors implicitly embedded within the data, making it univariate. The other type is multivariate, explicitly incorporating all factors influencing consumption into the model.

Regarding the prediction horizon, the model can be either one-step or multi-step, allowing the user to specify how many time steps ahead they wish to forecast demand. For [28], building neural network models is more like an art than a science, as there is a lack of a clear methodology for parameter determination. The author highlights the challenge in this domain as developing a systematic approach to assembling a suitable neural network model for the specific problem at hand.

2.4.1. Pipeline for creating energy prediction models

Constructing a neural network prediction model for a specific problem is not a trivial task. According to [29], a critical decision is determining the appropriate architecture. While there are tools that assist in this task, parameter selection also depends on the problem and/or the information available in the dataset.

A forecast of a county’s energy demand, for example, may rely on the recent history of demand (endogenous factors) as well as weather forecasts, which are exogenous factors. Many time series methods, such as autoregression with exogenous variables (ARX), have extensions that accommodate these variables. However, it is often necessary to prepare the exogenous data before use, ensuring it matches the temporal resolution of the variable being forecast (e.g., hourly, daily)[30].

The creation of models is complicated by the numerous choices that must be made prior to defining the model itself. Consequently, designers often face uncertainty regarding the decisions to be made. Therefore, instead of an exhaustive exploration of the subject, the main goal here is to provide support to streamline the process of creating load prediction models.

It is worth noting that the most accurate models are the ones that capture the behavior of input variables and replicate their impact on the output. Therefore, it is important to develop an understanding of which factors are most relevant to the problem and how to represent them in the model.

To identify these factors, questions related to the problem and the chosen prediction techniques were considered [31].

- a) Questions related to prediction techniques
 - 1. How many neurons should the input layer have?

The number of input nodes or neurons is probably the most critical decision for time series-based modeling since it defines which information is most important for the complex non-linear

relationship between inputs and outputs. The number of input neurons should match the number of relevant variables for the problem [29].

2. How many hidden layers should the model have?

As emphasized by [29], a network with no hidden layers represents a model with linear outputs, equivalent to linear statistical prediction models. Some authors, such as [32], suggest that typically no more than two hidden layers are necessary in a network to effectively address most prediction problems.

This study assess two architectures: one shallow and the other deep. They vary in the number of intermediate layers, with the the deep model incorporating additional intermediate layers. The specific attributes of shallow and deep models are detailed in Appendix A.

3. How many neurons should the hidden layers have?

According to [33], an excessive number of neurons per hidden layer can degrade model generalization and lead to overfitting. Therefore, it is suggested that the number of hidden nodes should be correlated with the number of inputs. The following expression was proposed to define it:

$$Nh = (4n^2 + 3)/(n^2 - 8), \quad (5)$$

where n is the number of input variables and Nh is the number of neurons in the intermediate layers. According to [17], a deep neural network allows you to model complex data with fewer processing units compared to a conventional neural network with similar performance.

4. What hyperparameters to use and how to select them?

Hyperparameters are attributes that control machine learning model training. Deep neural network models have several hyperparameters. The strategy used to define them was to process the model using GridsearchCV from Sklearn [34]. By processing the dataset with GridsearchCV, the best parameter adjustments were identified. Refer to Appendix B for further details.

5. Which training algorithm to use?

Training a neural network entails minimizing a non-linear function, adjusting the network's weights to minimize the total mean squared error between desired and actual values. The backpropagation algorithm, a gradient descent algorithm, is widely utilized for this purpose [29]. In this method, the magnitude of each step is referred to as the learning rate. A small learning rate typically results in slow learning, whereas a high learning rate may induce network oscillation. The standard backpropagation technique is favored by most researchers, as it supports both online and batch updates.

6. Which prediction algorithm to use?

As mentioned in section 3.2, deep learning techniques are the most promising for demand prediction applications. Therefore, they were prioritized in this study.

7. How many output nodes to use?

The number of output nodes is relatively easy to specify because it is related to the problem under study. For a time series-based problem, the number of output nodes corresponds to the number of time intervals ahead that you want to predict [29].

b) Questions related to problem

1. Why preprocess the data?

Data cleaning is necessary whenever outliers or out-of-pattern data are identified. Normalization is important because it ensures that all input data for the model has the same weight and range of variation [35].

2. Which input variables to consider?
Some variables are dominant compared to others and need to be prioritized for inclusion in the model. To select them, algorithms that quantify the correlation between these variables and the target variable were used. Examples of these algorithms are available in the open-source software package Waikato Environment for Knowledge Analysis (WEKA). It should be noted that the selection of input variables affects the training speed and the model's response [36].
3. How to reproduce seasonality?
Some studies have proposed encoding methods to represent seasonality. For example, [20] suggested encoding methods to represent annual, weekly and daily seasonality. These methods were tested in [21] and considered in the models evaluated in this study.
4. How to consider the calendar?
The calendar allows differentiating, from the perspective of energy consumption, holidays, workdays and hours of work throughout the day when energy consumption changes. The coding used is shown in Table 2.

Table 2. Date and Time of Day Encoding Methods.

Name	Description	Value Range
Holiday	Holiday or not	[0,1]
Workday	Workday or not	[0,0.5,1]
Working	Working or not	[0,0.5,1]

Source: [12].

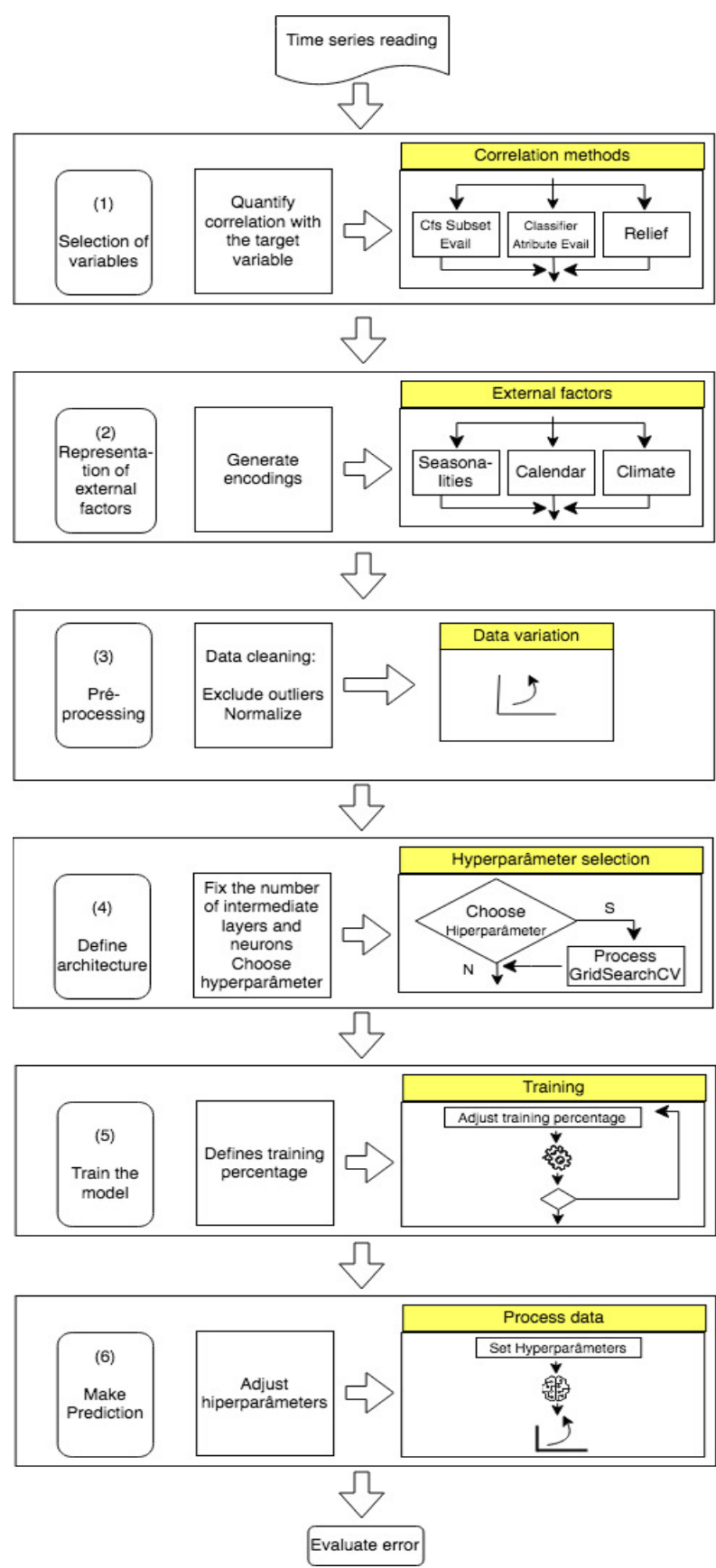


Figure 4. Pipeline for creating an electricity consumption forecast models. (Source: Author).

5. How to represent the discomfort index?

The discomfort index is important because it indicates the user's predisposition to turn on the air conditioning or other climate control equipment in their environment, which can be energy-consuming. It is calculated as indicated in section 3.3

Based on the previous analyses, it was possible to define the essential steps required for constructing more precise and well-fitted prediction models. This is illustrated in the pipeline depicted in Fig. 4.

Finally, to measure the accuracy of the models (Evaluate Error block), several metrics can be employed. In this study, we use the Mean Absolute Percentage Error (MAPE) metric, a widely accepted measure of prediction accuracy in time series analysis. MAPE is calculated by taking the average of the absolute percentage errors between the predicted and actual values. This metric is particularly useful because it expresses the error as a percentage, making it easier to interpret the model's performance across different scales and datasets. It is determined by:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - F_i}{F_i} \right| \times 100, \quad (1)$$

where, n represents the number of observed instances, A_i is the current consumption value and F_i is the predicted value for each point. It shows that the total absolute error value is determined by summing up the absolute values of each instance divided by the number of evaluated points, n . The conclusions are based on MAPE values, where a lower MAPE value indicates higher model accuracy [19].

3. Implementation and results

The subsequent discussion and analysis are anchored in the initial premise that constructing an energy consumption forecast model is inherently complex, primarily due to the multitude of decision-making processes involved.

To enhance clarity in comprehending the construction of load prediction models, a structured pipeline has been introduced. This pipeline encompasses the stages essential for developing multivariate and multistep models utilizing deep learning techniques. Each stage is meticulously justified, emphasizing its necessity and specific contribution.

Thus, what follows is a systematic evaluation of the fundamental components essential to composing accurate load forecasting models. To demonstrate the pipeline's applicability and evaluate its efficacy, a comprehensive array of experiments has been conducted. These experiments encompass various input conditions (number of variables) and output conditions (time horizon), as detailed in the subsequent tables. Notably, they encompass different configurations of input variables, prediction algorithms, and model architectures (shallow or deep).

3.1. Dataset

The dataset used in this study can be accessed on the ISO NE (Independent System Operator New England) website at <http://www.iso-ne.com>. It includes the total electric load of several cities in England for a period between January 2017 and December 2019 and comprises 23 independent variables of the following nature: weather information, economic indicators and market data.

Graphical representations displaying annual, weekly, and daily variations in total energy consumption for the aforementioned dataset are available in [22]. The Annual Variation chart reveals peaks during the summer months and another, albeit less intense, peak in the winter months. The Weekly Variation chart illustrates higher consumption levels on Mondays, while the Daily Variation chart depicts consumption peaks during the day, particularly between 6 and 7 pm.

3.2. Experiments

The conducted experiments are summarized in Tables 3 to 6, encompassing various combinations of input variables, algorithms, and prediction horizons. The implementation was carried out using the Python programming language along with machine learning libraries, Sklearn

and Keras [34]. In order to underscore the significance of step (1) of the pipeline, experiments detailed in Table 3 were conducted, focusing on variations in input variables and comparing processing times.

It is important to highlight that the processing detailed in Table 3 includes seasonality and calendar representation. Additionally, it's noteworthy that the variable selection algorithms employed originate either from the Weka tool or were custom-built in Python (refer to the source column). The experimental results were compared based on precision (MAPE) and processing time (refer to the magnitudes column).

Table 3. Experiments with Variation in Input Variables (Horizon: One Step Ahead).

Technique	Source	Selected variables	Magnitudes	Dnn		Cnn		Lstm		Cnn+Lstm	
				Shallows	Deep	Shallows	Deep	Shallows	Deep	Shallows	Deep
Cfs Subset Eval		RT demand, DACC, DA MLC, RT MLC, MIN_5MIN_RSP	Maape	0,22	0,18	1,8	0,74	0,37	0,24	0,21	0,18
		MAX_5MIN_RSP	t(s)	152	185	276	340	887	1798	739	848
		MAX_5MIN_RSP									
Classifier attribute Eval	Weka	MAX_5MIN_RSP, DA EC, DA CC, RT demand, DA LMP	Maape	0,27	0,23	1,6	0,48	0,21	0,16	0,26	0,24
			t(s)	153	187	342	384	994	1813	761	790
Principal Componentes		RT LMP, RT EC, DA LMP, DA EC, RT MLC	Maape	7,8	6,84	9,67	6,5	10,95	9,79	8,06	7,57
			t(s)	87	130	150	219	594	912	996	386
Relief		RT demand, DA demand, DA EC, DA LMC, Reg Service Price	Maape	0,52	0,33	3,82	0,22	0,23	0,16	0,16	0,17
			t(s)	112,8	123	274	339	906	1792	692	823
Mutual information	Python	DA MLC, DA LMP, MIN_5MIN_RSP, DA EC, Dew Point	Maape	5,9	5,76	8,15	5,03	10,21	7,82	7,23	6,11
			t(s)	155	186	270	339	581	1827	725	792
-	-	All	Maape	0,86	0,38	2,27	1,05	0,29	0,2	0,25	0,20
			t(s)	118	123	158	153	591	920	695	879

Source: Author.

- Based on these experiments, the following observations can be made:
- Among the tested variable selection algorithms, notable ones include CFS Subset Eval, Classifier Attribute Eval, and Relief. These algorithms selected the 5 or 6 variables most correlated with the target variable, yielding predictions with error rates similar to or lower than those obtained when

- considering all variables in the database (as indicated in the last line of Table 3). This underscores the relevance of the selected variables.
- Regarding the performance of models with different prediction algorithms, LSTM stands out for requiring the highest computational cost. It necessitates 10 times more processing time than DNN, 5 times more than CNN, and 2.5 times more than the combined CNN+LSTM model. However, the models employing LSTM and a combination of CNN+LSTM achieved the highest accuracy.
 - Comparing the performance of shallow and deep models, it was observed that the latter incur a higher computational cost. Nevertheless, in most scenarios, they demonstrate superior accuracy compared to shallow models.

To demonstrate the importance of integrating external factors into the model, as outlined in step (2) of the pipeline, some experiments were conducted, and their results are presented in Tables 4 and 5.

In these tables, the columns indicate the presence of external factors, with the distinction based on the prediction horizon. The acronyms and elements included in the columns are defined as follows: Sh-Shallow; D-Deep; SI-System Load; W-Weather; S-Sazonality; Fs-Feature Selection; Av-All Variable, Id- Discomfort index, @- Percentage variation of error. For instance, the column SI+W presents the results incorporating both System Load and Sazonality external factors for different prediction algorithms.

Table 4. MAPE of Processes with Different Sets of Input Variables.

Horizon : One step ahead											
T e c h n i q u e	M o d e l	SI	SI+W	SI+S+ C+ Id	@ (%)	Fs	Fs+S+ C+ Id	@ (%)	Av	Av+S+ C+ Id	@ (%)
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Dnn	Sh	11,3	6,76	5,38	52,4	0,29	0,19	34,5	0,19	0,18	5,3
	D	10,85	5,98	8,57	44,9	0,37	0,17	54,1	0,22	0,17	22,7
Cnn	Sh	12,13	8,46	8,0	34	2,85	2,23	21,8	3,40	2,26	33,5
	D	11,04	4,95	4,29	61,1	0,18	0,18	0	0,26	0,16	38,5
Lstm	Sh	12,37	8,31	7,86	36,5	0,65	0,25	61,5	0,47	0,17	63,8
	D	11,74	7,59	9,0	23,3	14,1	7,07	49,8	14,96	10,45	30,1
Cnn+ Lstm	Sh	10,38	6,56	5,67	45,4	0,31	0,21	32,2	0,42	0,19	54,8
	D	11,11	5,09	7,60	31,6	0,25	0,15	40	0,25	0,24	4

Source: Author.

Table 5. MAPE of Processes with Different Sets of Input Variables.

Horizon: Twelve step ahead											
T e c h n i c s	M o d e l	Sl	Sl+ W	Sl+ S+ C+ Id	● (%)	Fs	Fs+ S+ C+ Id	● (%)	Av	Av+ S+ C+ Id	● (%)
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Dnn	Sh	33,2	23,44	27,98	15,7	36,58	29,77	18,6	29,8	29,76	0,13
	D	48,1	27,63	37,01	23,1	45,03	32,4	28,1	44,0	40,58	7,77
Cnn	Sh	12,01	15,37	4,92	59,0	4,91	4,83	1,6	4,92	4,68	4,9
	D	10,6	9,19	3,77	64,4	4,79	3,96	17,3	4,39	3,63	17,7
Lstm	Sh	13,00	14,36	9,82	24,5	4,39	4,30	2,0	4,29	3,66	14,7
	D	15,58	15,65	15,27	2,0	15,8	11,34	4,46	15,4	15,15	1,62
Cnn+ Lstm	Sh	15,00	8,04	4,58	69,5	4,42	3,93	11,0	4,12	4,09	0,7
	D	11,24	8,65	10,7	4,8	15,94	10,82	32,1	15,6	15,36	1,5

Source: Author.

Considering the experiments detailed in Tables 4 and 5, the following observations can be made:

- The distinguishing factor between the tables is the prediction horizon. It is evident that increasing the prediction horizon leads to a reduction in accuracy.
- Experiments involving only the target variable (Sl), with or without external factors, exhibited the highest errors. This implies that deep learning algorithms face limitations in their generalization capacity when operating with a very limited number of input variables, resulting in elevated error rates.
- Comparing predictions based on only the target variable (Sl), the selected variables (Fs), and all variables in the dataset (Av), it is notable that errors in the Fs and Av columns are closely aligned. This suggests that despite variable selection reducing the number of input variables from 23 to 6 or 7, this reduction did not lead to an increase in prediction errors.

To evaluate the impact of architecture on the model's performance, experiments detailed in Table 6 were carried out. These experiments varied the number of intermediate layers and neurons within those layers. Deep learning models were employed, and variable selection was conducted using the Weka Classifier Attribute Eval. Additionally, the inclusion of external factors such as seasonality and calendar was taken into account during the experiments.

Table 6. Experiments with deep models varying the number of intermediate layers and neurons per intermediate layer (MAPE values for one-step-ahead prediction).

		DNN			CNN			LSTM			CNN+LSTM		
		Number of neurons			Number of neurons			Number of neurons			Number of neurons		
		6	15	32	6	15	32	6	15	32	6	15	32
Number of intermediat e layers	2	0,24	0,15	0,41	0,18	0,15	0,15	0,23	0,16	0,19	0,3	0,15	0,25
	3	0,3	0,54	0,16	0,15	0,26	0,29	0,52	0,15	0,17	0,18	0,43	0,52
	4	0,17	0,44	0,2	0,19	0,18	0,17	2,34	0,46	0,81	0,18	0,17	0,17
	6	0,24	0,15	0,16	0,3	0,18	0,41	10,0	15,4	0,40	15,53	0,40	0,35

Source: Author.

From the point of view of the model structure, Table 6 suggests that utilizing two or three intermediate layers with a recommended number of neurons equal to 6 (Eq. (8)) enables the generation of predictions with lower computational cost and error rates within an acceptable range (<1%).

In summary, the conducted experiments allow for several conclusions:

- Multivariate and multistep models offer flexibility by permitting variations in the number of input variables and prediction intervals, making them appealing for energy consumption forecasts.
- It has been demonstrated that incorporating external factors enhances model accuracy, with experiments achieving up to a 60% increase in accuracy.
- It has been proven that incorporating external factors increases the accuracy of the model, with experiments achieving an accuracy increase of up to 60%.
- Variable selection is a necessary measure when the number of input variables is large. It enables the reduction of the problem's dimensionality while still achieving good accuracy through the model. This reduction in dimensionality promotes the application of deep learning techniques and the utilization of deep models.
- In deep learning models, defining the architecture is a crucial step. The experiments have demonstrated that it's unnecessary to incorporate more than three intermediate layers or introduce an excessive number of neurons within these layers. It has been proven that adhering to the number of neurons defined by Eq. (8) yields satisfactory results.
- From the perspective of model accuracy, it's evident that the lowest error rate achieved was 0.15% when employing the CNN+LSTM deep learning technique, which considered both variable selection and the representation of external factors. In [37], the lowest error using CNN was 0.8%, while in [38], employing LSTM, it was 1.44%. Additionally, [8] suggests that errors ranging from 1% to 5% are typically expected for aggregate consumption.
- Tables 4 and 5 show results for experiments conducted under various input conditions and prediction horizons. The lowest error rates are highlighted in bold within these tables. It is evident that CNN models and composite models (CNN+LSTM) consistently demonstrated the lowest error rates across most scenarios. It occurs due to the advantageous feature extraction capability of CNN from input variables. The presence of CNN in both models significantly contributed to error rate reduction. Moreover, LSTM's proficiency in handling time series data further enhanced the composite model's performance, allowing it to surpass other models in certain experiments.

4. Conclusions and Future Research

This study emphasizes that the most critical stage of the process lies in the selection of input variables because, as it profoundly influences the model's architecture, processing time and accuracy. It was shown that the importance of a variable can be determined by its correlation with the target variable and this criterion was tested. Furthermore, it was illustrated that dimensionality reduction of the problem is achievable without compromising the model's efficacy.

A significant emphasis was placed on incorporating external factors that influence energy consumption. Only the consideration of the target variable and climatic variables, as commonly practiced in existing literature, was deemed insufficient. Through empirical experiments, it was elucidated that integrating external factors such as seasonality, calendar variations, thermal discomfort indices, and economic indicators substantially enhances forecast accuracy.

Moreover, meticulous attention was devoted to defining the model architecture. Various models incorporating deep learning algorithms were considered, with parameters like the number of intermediate layers and the number of neurons in these layers. Results indicated that models constructed with a small number of intermediate layers (up to three) and neurons therein, determined based on the number of input variables (as per Eq. (8)), yield favorable outcomes. The fact that the performance of deep models surpassed that of shallow models in most experiments demonstrates the relevance of the presence of intermediate layers in the model. Through them, the model detects the characteristics of the data, captures patterns and thus maps the complex non-linear relationship between data input and output.

Among the tested models, those employing the CNN algorithm and the combination of CNN and LSTM algorithms were the most robust and accurate for energy consumption prediction applications.

As a proposal for future research, it is suggested to individually evaluate the significance of each external factor and test the models with individual consumer datasets. This approach aims to ascertain potential alterations in the relevance of external variables and further enrich the predictive capabilities of the models.

Author Contributions Conceptualization of the paper and methodology were given by GA, RM, FV and PP; formal analysis, investigation, and writing (original draft preparation) by LA, GA, FV and RM; software and validation by LA and GA, writing—review and editing by FV, PP and RM. All authors read and approved the final manuscript.

Data Availability Data available within the article or its supplementary materials.

Acknowledgements This work was partially supported by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES/Brazil) (PrInt CAPES-UFSC “Automação 4.0”).

Code Availability The code that supports the findings of this study are available from the corresponding author upon reasonable request.

Conflict of interest The authors declare no conflict of interest.

Appendix A - Model Architectures

Two models were created: one called shallow and another called deep, which has two additional intermediate layers. Tables A1 to A4 show the internal structures of each of them.

Table A1. DNN Model Architecture.

Architecture: DNN TIPO 1 : Shallow			Architecture: DNN TIPO 2 : Deep		
Layer	Settings	Activation	Layer	Settings	Activation
FC 1	32	Relu	FC 1	32	Relu
Flatten			FC 2	6	Relu
FC 2	6	Relu	FC 3	6	Relu
Flatten			FC4	6	Relu
Output	1		Flatten		
			Output	1	

Table A2. CNN Model Architecture.

Architecture : CNN TIPO 1 : Shallow			Architecture: CNN TIPO 2 : Deep		
Layer	Settings	Activation	Layer	Settings	Activation
Conv 1	32	Relu	Conv 1	32	Relu
Pooling 1 (max)	2		Pooling 1 (max)	2	
Flatten			Conv 2	32	Relu
FC 1	6	Relu	Pooling 1 (max)	2	
Output	1		Flatten		
			FC 1	6	Relu
			Output	1	

Table A3. LSTM. Model Architecture.

Architecture: LSTM TIPO 1 : Shallow			Architecture: LSTM TIPO 2 : Deep		
Layer	Settings	Activation	Layer	Settings	Activation
LSTM 1	32	Relu	LSTM 1	32	Relu
Flatten			LSTM 2	6	Relu
drouput			Flatten		
Output	1		FC 1	6	Relu
			Flatten		
			drouput		
			Output	1	

Table A4. CNN+LSTM Model Architecture.

Architecture: CNN+LSTM TIPO 1 : Shallow			Architecture: CNN+LSTM TIPO 2 : Deep		
Layer	Settings	Activation	Layer	Settings	Activation
Conv 1	32		Conv 1	32	
LSTM 1	6	Relu	Pooling 1 (max)		
Flatten			Conv 2	32	
drouput			Pooling 1 (max)		
Output	1		LSTM 1	6	Relu
			LSTM 2	6	Relu
			Flatten		
			FC 1	6	Relu
			Flatten		
			Output	1	

Note: Tables A1 to A4 were generated by the author

Appendix B - Hyperparameter Selection

Neural network models have internal parameters known as hyperparameters. Table B1 lists the main hyperparameters of the models. The last column contains the values obtained from the processes using the GridSearchCV function from Sklearn. These values were used in the experiments conducted in this study.

Table B1. Hyperparameter Selection.

Hyperparameter	Interval	Selection
Number of hidden layers	[1..100]	1 to 6
Number of neurons	[1..100]	5
Learning Rating	[0.0001..0.01]	0,01
Dropout rate	[0.1.. 0.5]	0.1
Activation function	Relu, Sigmoid, Tanh	Relu
Optimizer	Adam,RMSprop,SGD	Adam
Batch Size	[16..512]	128
Loss function	Mse, mae, mape	Mse
Kernel	[2..5]	2
Number of epochs	[10..500]	30

Note: Table B1 were generated by the author

References

1. Merce, R.A., Grover, S.E., Conte, J Le. Load and Demand Side Flexibility Forecasting. In: *ENERGY Tenth Int. Conf. Smart Grids, Green Commun. IT Energy-aware Technol.* 2020, p. 1–6. <https://minesparis-psl.hal.science/hal-03520220/>

2. Khalid A. Khan, Md Muzakkir Quamar, Faleh H. Al-Qahtani, Muhammad Asif, Mohammed Alqahtani, Muhammad Khalid, Smart grid infrastructure and renewable energy deployment: A conceptual review of Saudi Arabia, *Energy Strategy Reviews*, Volume 50, 2023, 101247, ISSN 2211-467X, <https://doi.org/10.1016/j.esr.2023.101247>

3. Mystakidis;Koukaras, P.;Tsalikidis, N.; Iosnnidis, D. Compreeensive Review Of Techniques and Technologies. *Energies* 2024,17,1662.

4. Min, Hyunsik, et al. "SolarFlux Predictor: A Novel Deep Learning Approach for Photovoltaic Power Forecasting in South Korea." *Electronics* 13.11 (2024): 2071.

5. Mejía, M.; Villanueva, J.; Hernández, D.; Santos, F.; García, A. *A Review of Low-Voltage Renewable Microgrids: Generation Forecasting and Demand-Side Management Strategies.* *Electronics* 2021,10, 2093. [CrossRef] <https://www.mdpi.com/2079-9292/10/17/2093>

6. Bunn, DW (1982). Short-term forecasting: a review of procedures in the electricity supply industry. *J Opl Res Soc*33:533–545. <https://link.springer.com/article/10.1057/jors.1982.116>

7. Park, H.Y.; Lee, B.H, Son, J.H., et al. A comparison of neural network- based methods for load forecasting with selected input candidates. In: *Proceedings of the 2017 IEEE international conference on industrial technology (ICIT)*, Toronto, ON, Canada, 22–25 March 2017, pp.1100– 1105. New York: IEEE. <https://ieeexplore.ieee.org/abstract/document/7915516>

8. Peñaloza, A.K.A.; Balbinot, A.; Leborgne, R.C. Review of Deep Learn- ing Application for Short-Term Household Load Forecasting. In *Proceedings of the 2020 IEEE PES Transmission & Distribution Conference and Exhibition—Latin America (T&D LA)*, Montevideo, Uru- guay, 28 September–2 October 2020, pp. 1–6. [CrossRef] <https://ieeexplore.ieee.org/abstract/document/9326148>

9. Kong, Weicong, et al. "Short-term residential load forecasting based on LSTM recurrent neural network." *IEEE transactions on smart grid* 10.1 (2017): 841-851.

10. Mannini, Romain, et al. "Predictive Energy Management of a Building-Integrated Microgrid: A Case Study." *Energies* 17.6 (2024): 1355.

11. Elsworth, S., G üttel, S. Time series forecasting using lstm networks: A symbolic approach. 2020. <https://arxiv.org/abs/2003.05672>

12. Glavan, M. Gradišar, D. Moscariello, S. Juričić, D. and Vrancić, D. Demand-side improvement of short-term load forecasting using a pro- active load management–A supermarket use case. *Energy Buildings*, vol. 186, pp. 186–194, Mar. 2019, doi: 10.1016/j.enbuild.2019.01.016.

- https://www.sciencedirect.com/science/article/pii/S0378778818327014?casa_token=wgAT7ZatiCQAAAAA:3j41Epuwvmv2ECqgrQY5lEMSoOzQChy1Vhz5uFLyfAeyqlfgUVH TnX5wmp7etOO4EzHv3oJwwDPpsw
13. Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Hasan, M., Van Essen, B.C., Awwal, A.A., Asari, V.K. (2019). A state-of-the-art survey on deep learning theory and architectures. *Electronics* 8 (3), 292. <https://www.mdpi.com/2079-9292/8/3/292?ref=https://githubhelp.com>
 14. Alzubaidi L, Zhang J, Humaidi AJ, Al-Dujaili A, Duan Y, Al-Shamma O, et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data* 2021;8:53. <https://link.springer.com/article/10.1186/s40537-021-00444-8>
 15. Ungureanu S, Topa V. Czikier AC. Deep learning for short-term load forecasting-industrial consumer case study. *Appl Sci.* 2021, 11(21):10126. <https://www.mdpi.com/2076-3417/11/21/10126>
 16. Jiang, W. Deep learning based short-term load forecasting incorporating calendar and weather information. *Internet Technology Letters*, 2022, 5 (4): e383. https://onlinelibrary.wiley.com/doi/abs/10.1002/itl2.383?casa_token=7XhMJNSPx18AAAAA:EDZSFbFFKD JZ8h2IN8ODnI9weuCICdK9JpZ6pSulCSXo1NbPuDO1OT8My4vYH2PIloBFIt2I85zDn4zALw
 17. Son, N. Comparison of the Deep Learning Performance for Short- Term Power Load Forecasting. *Sustainability*, 2021, 13, 12493. <https://www.mdpi.com/2071-1050/13/22/12493>
 18. Kuo P. H. , Huang C. J.. A high precision artificial neural networks model for short-term energy load forecasting. *Energies* 11(1) (2018) 213
 19. Kim, T.; cho S. (2019) Predicting residential energy consumption using CNN-LSTM neural. *Energy* 2019, 258,114087.https://www.sciencedirect.com/science/article/pii/S0360544219311223?casa_token=YBV0IOSh4UAAAAA:zqpt8i3nUD1zqKuW7UTqtrR4o1nLlh_cozZJ6FB7e1axSvtb_O0SCsoHgd9qrX7VRjUwrkhD8gABPw
 20. Dudek, G. (2020) Multilayer perceptron for short-term load forecasting: From global to local approach. *Neural Comput. Appl.*, vol. 32, no. 8, pp. 3695–3707. <https://link.springer.com/article/10.1007/s00521-019-04130-y>
 21. Amaral, L. S., de Araújo, G. M., & de Moraes, R. A. R. (2022). Analysis of the factors that influence the performance of an energy demand forecasting model. *Advanced Notes in Information Science*, 2, 92-102.
 22. Amaral, L. S.; Araujo, G. M.; Moraes, R. A. An Expanded Study of the Application of Deep Learning Models in Energy Consumption Prediction. *Conference on Data, 2022 - Springer*. https://link.springer.com/chapter/10.1007/978-3-031-22324-2_12
 23. Buildings, (<https://www.iea.org/energy-system/buildings>)
 24. KawBamura, T. Distribution of discomfort index in Japan in summer season. *J Meteorol. Res.*, 17:460-466, 1965.
 25. Ono, H.S.P.; Kawamura, T. Sensible climates in monsoon Asia. *Int.J.Biometeorol.*, Ibaraki, v. 35, p.39-47, jan 1991. <https://link.springer.com/article/10.1007/BF01040962>
 26. Júnior, Arielson José de Oliveira et al. Software para cálculo de desconforto térmico humano. *Tekhne e Logos*, v. 5, n. 1, p. 56-68, 2014. <http://revista.fatecbt.edu.br/index.php/tl/article/view/280>
 27. Suganthi L, Samuel A. (2012) Energy models for demand forecasting – a review. *Renew Sustain Energy Rev*;16:1223–40. https://www.sciencedirect.com/science/article/pii/S1364032111004242?casa_token=Hlayq1Fbxj4AAAAA:cjm0F6jhtVa6OutfgK3QC-5R7B4TgUVCnwy66kHfEFgzJ4FnSSwpeR3ZVGVL9vBtk6VSA0FifUk-g
 28. Curry, B. ;Morgan, P. H. Model selection in neural networks: some difficulties. *European Journal of Operational Research*, vol. 170, no. 2, pp. 567–577, 2006. https://www.sciencedirect.com/science/article/pii/S0377221704004278?casa_token=o5cAXr-RobYAAAAA:bnJwn8Pt50NAQXIy1WkuhgjDPefSfW4wu_rfLXgNn36re9SYyx13E_hkt3V7WpzkRSUS1nGFkpEAd9w
 29. Zhang, G., Eddy Patuwo, B., & Y. Hu, M. Forecasting with artificial neural networks:: The state of the art. *International Journal of Forecasting*, 1998, 14(1), 35–62 https://www.sciencedirect.com/science/article/pii/S0169207097000447?casa_token=AHgCjtEjCf8AAAAA:j2A1qd9xE1W7QHh4BHKbHNXtpMbRVK49eUzJBWoWmaw3uRgPpISrQV5ughT7SQRWivtzBdnMi-Y6QA
 30. Fotios Petropoulos, Daniele Apiletti, Vassilios Assimakopoulos, Mohamed Zied Babai, Devon K. Barrow, Souhaib Ben Taieb, Christoph Bergmeir, Ricardo J. Bessa, Jakub Bijak, John E. Boylan, Jethro Browell, Claudio Carnevale, Jennifer L. Castle, Pasquale Cirillo, Michael P. Clements, Clara Cordeiro, Fernando Luiz Cyrino Oliveira, Shari De Baets, Alexander Dokumentov, Joanne Ellison, Piotr Fiszeder, Philip Hans Franses, David T. Frazier, Michael Gilliland, M. Sinan Gönül, Paul Goodwin, Luigi Grossi, Yael Grushka-Cockayne, Mariangela Guidolin, Massimo Guidolin, Ulrich Gunter, Xiaojia Guo, Renato Guseo, Nigel Harvey, David F. Hendry, Ross Hollyman, Tim Januschowski, Jooyoung Jeon, Victor Richmond R. Jose, Yanfei Kang, Anne B. Koehler, Stephan Kolassa, Nikolaos Kourentzes, Sonia Leva, Feng Li, Konstantia Litsiou, Spyros Makridakis, Gael M. Martin, Andrew B. Martinez, Sheik Meeran, Theodore Modis, Konstantinos Nikolopoulos, Dilek Önköl, Alessia Paccagnini, Anastasios Panagiotelis, Ioannis

- Panapakidis, Jose M. Pavía, Manuela Pedio, Diego J. Pedregal, Pierre Pinson, Patrícia Ramos, David E. Rapach, J. James Reade, Bahman Rostami-Tabar, Michał Rubaszek, Georgios Sermpinis, Han Lin Shang, Evangelos Spiliotis, Aris A. Syntetos, Priyanga Dilini Talagala, Thiyanga S. Talagala, Len Tashman, Dimitrios Thomakos, Thordis Thorarinsdottir, Ezio Todini, Juan Ramón Trapero Arenas, Xiaoqian Wang, Robert L. Winkler, Alisa Yusupova, Florian Ziel, Forecasting: theory and practice, *International Journal of Forecasting*, Volume 38, Issue 3, 2022, Pages 705-871, ISSN 0169-2070, <https://doi.org/10.1016/j.ijforecast.2021.11.001>).
31. Cogollo, M. R, Velasquez J.D. Methodological advances in artificial neural networks for time series forecasting. *IEEE Lat Am Trans* 2014;12(4):764–71. <https://ieeexplore.ieee.org/abstract/document/6868881>
 32. Lippmann, R.P. An introduction to computing with neural adaptively trained neural network. *IEEE Transactions on neural networks*, IEEE ASSP Magazine, 1987, April, 4–22.
 33. Sheela K. G. and Deepa, S. N. Review on methods to fix number of hidden neurons in neural networks. *Math. Problems Eng.*, vol. May 2013, Art. no. 425740. doi: 10.1155/2013/425740. <https://www.hindawi.com/journals/mpe/2013/425740/>
 34. Chollet, f. (2015). Keras: The python deep learning API. Disponível em <https://keras.io/>
 35. Ribeiro, A.M.N.C.; do Carmo, P.R.X.; Endo, P.T.; Rosati, P.; Lynn, T. Short- and Very Short-Term Firm-Level Load Forecasting for Warehouses: A Comparison of Machine Learning and Deep Learning Models. *Energies* 2022, 15, 750. [CrossRef] <https://www.mdpi.com/1996-1073/15/3/750>
 36. Gnanambal, S.; Thangaraj, M.; Meenatchi, V.; Gayathri, V. Classification algorithms with attribute selection: An evaluation study using WEKA. *Int. J. Adv. Netw. Appl.* 2018, 9, 3640–3644. <https://www.proquest.com/openview/8f1f0774ec448384012948adb2af16e3/1?pq-origsite=gscholar&cbl=886380>
 37. Bendaoud, N.M.M.; Farah, N. Using deep learning for short-term load forecasting. *Neural Comput. Appl.* 2020, 32, 15029–15041. <https://link.springer.com/article/10.1007/s00521-020-04856-0>
 38. Torres, J., Martínez-Álvarez, F. & Troncoso, A. A deep LSTM network for the Spanish electricity consumption forecasting. *Neural Comput. Appl.* 1–13 (2022). <https://link.springer.com/article/10.1007/s00521-021-06773-2>

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