Article

Deep Learning for Short-Term Load Forecasting—Industrial Consumer Case Study

Stefan Ungureanu ^{1,*} , Vasile Topa ² and Andrei Cristinel Cziker ¹

¹ Department of Electric Power Systems and Management, Technical University of Cluj-Napoca, 400027 Cluj-Napoca, Romania; andrei.cziker@enm.utcluj.ro

² Department of Electrotechnics and Measurements, Technical University of Cluj-Napoca, 400027 Cluj-Napoca, Romania; vasile.topa@ethm.utcluj.ro

* Correspondence: stefan.ungureanu@enm.utcluj.ro; Tel.: +40-741038948

Abstract: In the current trend of consumption, electricity consumption will become a very high cost for the end-users. Consumers acquire energy from suppliers who use short, medium, and long-term forecasts to place bids in the power market. This study offers a detailed analysis of relevant literature and proposes a deep learning methodology for forecasting industrial electric usage for the next 24 h. The hourly load curves forecasted are from a large furniture factory. The hourly data for one year is split into training (80%) and testing (20%). The algorithms use the previous two weeks of hourly consumption and exogenous variables as input in the deep neural networks. The best results prove that deep recurrent neural networks can retain long-term dependencies in high volatility time series. Gated recurrent units (GRU) obtained the lowest mean absolute percentage error of 4.82% for the testing period. The GRU improves the forecast by 6.23% compared to the second-best algorithm implemented, a combination of GRU and Long short-term memory (LSTM). From a practical perspective, deep learning methods can automate the forecasting processes and optimize the operation of power systems.

Keywords: machine learning; deep learning; short-term forecasting; industrial electricity load

Citation: Ungureanu, S.; Topa, V.; Cziker, A. Deep Learning for Short-Term Load Forecasting—Industrial Consumer Case Study. *Appl. Sci.* **2021**, *11*, 10126. <https://doi.org/10.3390/app112110126>

Academic Editor: Edris Pouresmaei

Received: 4 October 2021

Accepted: 25 October 2021

Published: 28 October 2021

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

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Types of electric load forecasting techniques fall into three main categories regarding the forecast horizon: STLF (short-term load forecasting), MTLF (medium-term load forecasting), and LTLF (Long term load forecasting). The authors in [1] present a classification based on the data frame used before the forecast, including very short-term load forecasting and a framework based on several points predicted into the future. In our study, we implement short-term forecasting for 24 steps (hours). Load forecasting (LF) can offer great value if the process can be automated and operated without human intervention.

The approach presented in the article for 24 h ahead forecast facilitates the access to the DAM (day ahead market) and Intra-day market to minimize the difference between real and forecasted values. This difference is mandatory to be balanced on the balancing market, which represents a financial problem for the supplier. Most large non-residential consumers have a single tariff (price/MWh) and few electricity suppliers offer time differentiated prices and time of use tariffs. This single tariff is calculated to cover the expenses with portfolio balancing. Every supplier needs to balance the portfolio of clients by being a balancing responsible party (BRP) or by submitting these responsibility to other parties. Either way, balancing a portfolio is a challenging task for every supplier.

In literature, the classification of forecasting determines two main categories illustrated in Figure 1: qualitative and quantitative. Work developed by [2,3] is a comprehensive and detailed review on selecting forecasting methods from a business standpoint. Regarding the analogy for electricity load, forecasting has the same foundation principles with few particularities. Qualitative techniques apply empirical knowledge of LF experts to make a

judgment call on the forecast. Qualitative techniques used for medium-long term prediction are primarily expert opinions, such as the Delphi Method, Market Research, or Panel consensus [4].

Figure 1. Classification of time series forecasting techniques.

Quantitative techniques are better for short-term forecasting and consist of (i) time series forecasting and (ii) causal forecasting. In time series forecasting, the historical data is a set of chronologically ordered observation points where X_t is recorded in time, $(X_t)_{t=0, \pm 1, \pm 2, \dots}$, observed at times $t = 1, 2, \dots, n$ ($X_1, X_2, \dots, X_n$). Contrasting CF, TSF is the natural ordering of the data points. TSF help understand components such as patterns, trends, peak values and any irregularity or variation in the series of data.

The importance of load forecasting is highlighted by many papers such as [5,6] which presents a detailed literature review. In Table 1, electricity market participants who can benefit from forecasting are highlighted.

Table 1. Impact of load forecasting.

Market Entity	Advantages	Disadvantages/Obstacles
DSO	optimized power flow; RES integration; lower grid losses	big data; high and inconsistent errors by similar models and methods (depending on the period of time, load category, forecasting window); additional resources required (time, qualified employees, software, training)
TSO	lower transmission losses, better planning	big data processing; high errors; inconsistent forecasting;
Energy Supply	better trade, lower costs for portfolio balancing	client commitment; communication about activity planning; high errors or inconsistency of load forecasting;
End-user	lower bills, increased energy efficiency	low interest, as main activities are more important than electricity usage; lack of transparency;

2. Literature Review

A plethora of approaches consisting of time series analysis, regression, smoothing techniques, artificial intelligence, artificial neural networks, machine learning, deep learning, reinforcement learning and various hybrid methods can make this area of research overwhelming. Some authors suggest that established models are better [7–9] presents evidence that complexity harms accuracy. Authors in [10] propose the Golden Rule to provide a unifying theory of forecasting, while others embed multiple algorithms to build hybrid methods combining characteristics of traditional statistics and machine learning. There is truth on both sides; some algorithms will work better or worse depending on historical data or applied period. The forecasting objectives are to minimize errors and improve economic activity: revenue, profit, and higher customer satisfaction. Low error forecasts are of no inherent value if ignored by the industry or otherwise not used to improve organizational performance. Forecasting competitions presented in [11,12] is one of the best ways to compare algorithms on reliable historical data and point out

the results. In multiple cases, recurrent neural networks (RNN) architecture stands out as a stable algorithm. The work done by [13] presents an extensive experimental study using seven popular DL architectures and found that LSTM is the most robust type of recurrent network, and while LSTM provides the best forecasting accuracy, convolutional neural networks (CNN) are more efficient and suffer less variability of results. In this paper, various RNN networks are applied for industrial load forecasting and analyzed to establish the best architecture for deep recurrent neural networks. In the article [14] authors point out that the difference between simple RNN to GRU and LSTM is that the number of parameters increases, a conclusion also presented in our article. For 24 h ahead forecasting commercial building data, authors concluded that the DNN model achieved worse results than the sequence to sequence RNN models. The authors in [15] present a simple recurrent neural network for the one-hour prediction of residential electric load. The model takes as inputs weather data as well as data related to electricity consumption. The percentage error calculated for a week test is 1.5% for the mean error and 4.6% for the maximum error. The difference between industrial load and residential usage is that the latter is highly dependent on weather data and daily patterns are more repetitive. In our article, exogenous variables such as temperature, humidity, and dew point are used in forecasting, because the industrial processes analyzed are influenced by these variables. Day-ahead forecasting of hourly large city load based on deep learning is studied by [16] with a novel flexible architecture that integrates multiple input features processed using different types of neural network components according to their specific characteristics. The authors have implemented multiple parallel CNN components with different filter sizes to introduce parallel structure into the DNN model instead of stacking DNN layers. The proposed architecture (MAPE: 1.405%) outperformed the CNN-LSTM (MAPE: 1.475%) and the DNN (MAPE: 1.665%). Another approach based on RNN and CNN is proposed by [17], consisting of convolutional layers and bidirectional LSTM and GRU recurrent layers to predict the next hour utility load. The results of experiments on two datasets (0.67% and 0.36% MAPE) demonstrate that the proposed model outperforms the conventional GRU and LSTM models. In this article, we found that the deep GRU network performs better than the combined GRU + LSTM network. A comprehensive comparison performed by authors in [18] concludes that RNNs require more resources than traditional models, but perform better. We reached similar findings in our article, the GRU unit is simpler than the LSTM unit, as well as faster in computations. The article presents that overall the LSTM performs better than GRU, which contradicts the results for short-term load forecasting presented in our article. The authors in [19] compare different variations of the LSTM algorithm and conclude that the longer the historical data available for training, the better the load forecasting accuracy would be. For building loads, the day-ahead forecasting errors show up to 45% improvement using RNNs (LSTM, LSTM with attention, BiLSTM, BiLSTM with attention) in comparison with other states of the art forecasting techniques.

3. Materials and Methods

The forecasting methods are implemented in this paper use hourly data (Figure 2) from an industrial company active in the wood processing industry for an entire year (2019). The power supply for the factory is provided through twelve power transformers summing 12.6 MVA. The following technological processes, machinery, and equipment determine the electricity consumption forecasted in this article:

- Installations that serve the equipment for cutting and exhaust;
- Installations that serve the cooling system to ensure the necessary cold to keep in optimal conditions the substances used in the foaming process;
- Installations that serve the processing and cutting of sponges;
- The installations that serve the different subsections when making the mattresses;
- Equipment used for making upholstery and assembling all subassemblies;
- Interior lighting installations located in the physical perimeter of all production halls;
- Robots for packing finished products;

- Conveyors for the transport of products in the logistics warehouse;
- Specific facilities for food preparation in the canteen;
- other installations are specific to the universal processes which take place within this undertaking.

Figure 2. Hourly industrial load curve used for forecasting.

For the implementation Tensorflow [20] was used for deep learning applications. Keras [21] is a high-level API, open-source library for machine learning that works on top of Tensorflow. For the data preparation and visualization of the results, Scikit-learn [22], Numpy [23], and Seaborn [24] were used. The simulations computed on a PC Intel(R) Core(TM) i5-4690K CPU@3.5 GHz, RAM 16 GB, 64-bit operating system, x64-based processor.

The industrial consumer analyzed is a furniture factory consisting of all the technological processes necessary to manufacture furniture starting from raw wood, mainly electric drives. The consumer energy needs are electricity and wood scraps. Production of heat and hot water relies on burning the remaining wood from the technological processes. The heating in the winter period for the office building and factory production facilities is achieved with electric heaters which influence the consumption in the winter period together with the lightning systems (work schedule is in three shifts). High electricity consumption is driven by large ventilated storage halls used for the thermal preparation of the raw wood. Correlation between electric load and outdoor temperature, dew point, and humidity is observed. Working/non-working days load patterns are not the same because factory planning is highly dependent on production quota. A Dickey–Fuller test [25] made for the yearly load time series points to the null hypothesis and the non-stationarity of the time series. Reliable linear dependencies between exogenous variable and consumption could not be establish and deep learning became an option to explore for nonlinear dependencies. From all the algorithms implemented in this article, variations of RNN (LSTM, GRU, GRU-LSTM), the GRU algorithm offered the best result for forecasting. Given this reason, we tried to analyze which is the best structure for the GRU for our particular problem.

3.1. Deep Learning (DL)

There is a vast spectrum of terminology that tends to be confusing because of the interchangeability of utilization: artificial intelligence, machine learning, deep learning, artificial neural networks, or reinforcement learning. Machine learning is considered a subdomain of artificial intelligence [26]. Deep learning is a subdomain of machine learning, and neural networks are at the core of deep learning algorithms. The dissimilarity between a simple neural network and a deep learning algorithm is the number of neurons and structure of hidden layers (deep learning must have more than two hidden layers). ML techniques can be broadly grouped in two large sets—supervised and unsupervised. The methods related to the supervised learning paradigm classify objects in a pool using a set of known annotations/attributes/features. The unsupervised learning techniques

form groups among the objects in a batch by identifying similarities and then use them for classifying the unknowns. Reinforcement learning is a behavioral algorithm similar to supervised learning, not using sample data for training but by trial and error. A sequence of successful outcomes will develop the best recommendation or policy for a given problem.

DL models were developed to map a complex function between the last “n” hours (timesteps—also called lag) and predict how the time series can continue in the future, as presented in Figure 3. Most machine learning algorithms have hyperparameters; by setting the parameters, the ML algorithm can offer the desired results. The values of hyperparameters should not be calculated in the learning stage (because of the overfitting problem). To evaluate the generalization of the DL methods on the training data, we use a testing set of time series that the built network in the training stage did not experience prior.

Figure 3. Supervised learning with neural networks for forecasting.

In our work we use deep recurrent neural networks (DRNN) and variations of the algorithm. RNN is a sequential data neural network processor because it has internal memory to update the state of each neuron in the network with the previous input. Because RNN train with backpropagation, this can fail because of vanishing gradient descent. Deep networks combine multiple layers into the architecture and provide more significant benefits.

Neural networks build functions by multiplying a weight matrix to the input vector, add bias, and then apply the activation function to obtain non-linearity in the output. To calculate the current state, we can use the following Formula (1).

$$h_{(t)} = f(h_{(t-1)}, x_{(t)}; \theta) = \tanh(W h_{(t-1)} + U x_{(t)} + b) \quad (1)$$

In the equation above, the parameters θ include W , U , and b . The W and U are parameters representing weight matrices, and b is the bias vector. Hyperbolic tangent is the activation function $\tanh$ for the hidden state; other activation functions could be used. The output of the RNN cell is:

$$o_{(t)} = g(h_{(t)}; \theta') = V h_{(t)} + c \quad (2)$$

where V and c denote the weight and bias, the parameters θ' of the output function g . Matrix V and vector c are multidimensional outputs. The same set of parameters is applied at each time step for every RNN-cell [27].

LSTM was developed to improve the vanishing or exploding gradient problem and has become one of the most popular RNN architectures to date and was introduced by [28]. GRUs were later introduced by [29] as a simpler alternative and have also become quite popular. We will use both architectures in the context of the vanishing or exploding gradient problem. Many variants of LSTM and GRUs exist in the literature, and even the default implementations in various deep learning frameworks often differ. Performance is often similar, but this can confuse when reproducing results. The study proposed by [30] ranked MLP first in terms of forecasting performance, better than Support Vector Regression, RF,

ARIMA and RNN. The RNN is among the less accurate ML-based methods in this study, but the authors have not tried a variation of RNN.

The results presented in [31] for air quality prediction show that the LSTM and the CNN-LSTM generally perform better in multi-hour forecasting than other ML algorithms. For residential load with spatial and temporal features authors in [32] obtained the highest performance of 0.37 MSE (mean square error) with CNN-LSTM better than LSTM, GRU, Bi-LSTM, and Attention LSTM. In [33] authors showcase that CNN-LSTM model gives the lowest values of MAE, RMSE and MAPE compared to LSTM, RBFN and XG-boost models. An average MAPE of 3.22% is obtained for 24 hours ahead forecast for national consumption.

3.2. Gated Recurrent Units (GRU)

Many articles such as [34] present a general literature review of ML and work done by [35–37] details a comprehensive review for DL algorithms used for forecasting. The GRU combines input gate and forget gate of LSTM into an update gate Z_t and the output gate in LSTM is called a reset gate R_t in GRU [38] as shown in Figure 4.

Figure 4. Structure of the GRU cell.

The difference between GRUs and simple RNN is the implementation of gating for the hidden state, which uses a few steps to determine when the hidden state needs to be updated and when to reset.

The input is a sequence of data, for a given time step t , $X_t \in \mathbb{R}^{n \times d}$ (n is the number of sequences, i is the number of inputs) and the hidden state of the previous time step is $H_t \in \mathbb{R}^{n \times h}$ (h is the number of hidden units). Then, the reset gate $R_t \in \mathbb{R}^{n \times h}$ and update gate $Z_t \in \mathbb{R}^{n \times h}$ are implemented as follows in Equation (3):

$$\begin{aligned} R_t &= \sigma(X_t W_{xr} + H_{t-1} W_{hr} + b_r) \\ Z_t &= \sigma(X_t W_{xz} + H_{t-1} W_{hz} + b_z) \end{aligned} \quad (3)$$

where $W_{xr}, W_{xz} \in \mathbb{R}^{i \times h}$ and $W_{hr}, W_{hz} \in \mathbb{R}^{h \times h}$ represent weight parameters and $b_r, b_z \in \mathbb{R}^{1 \times h}$ are biases. In Equation (4) the reset gate R_t updates the candidate hidden state $\tilde{H}_t \in \mathbb{R}^{n \times h}$ at time step t :

$$\tilde{H}_t = \tanh(X_t W_{xh} + (R_t \odot H_{t-1}) W_{hh} + b_h) \quad (4)$$

where $W_{xh} \in \mathbb{R}^{d \times h}$ and $W_{hh} \in \mathbb{R}^{h \times h}$ are weight parameters, $b_h \in \mathbb{R}^{1 \times h}$ is the bias, and the symbol $\odot$ is the Hadamard (elementwise) product operator. For the nonlinearity of the values in the candidate hidden state, $\tilde{H}_t$ uses the tanh to maintain the values in the interval $(-1, 1)$. Hidden state at time step t , $H_t \in \mathbb{R}^{n \times h}$, is a combination of previous hidden state H_{t-1} as presented in Equation (5) and current time step candidate hidden state:

$$H_t = Z_t \odot H_{t-1} + (1 - Z_t) \odot \tilde{H}_t \quad (5)$$

Activation functions used in the GRU cell are sigmoid and hyperbolic tangent Equation (6):

$$\begin{aligned} \tanh(x) &= \frac{e^{2x} - 1}{e^{2x} + 1} \\ \sigma(x) &= \frac{1}{1 + e^{-x}} \end{aligned} \quad (6)$$

RNN model was implemented by [39] for forecasting non-residential loads (catering, electronic, and hotel industry), concluded that predicting each consumer is not as accurate as of the forecasts for substation loads, and obtained the best results using LSTM with MAPE ranging from 15.45% to 19.57%. Extreme gradient boosting regressor (XGBoost) for STLTF was implemented by [40] and achieved MAPE of 3.74% for a horizon of one week for substation loads, with hourly steps, which is a total of 168 h. For one-step forecasting [41] are presenting a model based on FFNN and LSTM for air compressor electricity usage. The authors in [42] have also been used LSTM for nonlinear, non-stationary, and nonseasonal univariate electric load time series over 96 steps ahead with a MAPE of 5.35%, but it does not mention what type of load is forecasting. The RNN model fails on vanishing gradient decent as mentioned above, so LSTM and GRU are designed to compensate for its failure by using gates such as the ones in Figure 4. GRU is designed to provides a longer-term memory [43].

Multiple hidden RNN layers can be stacked on each another. The main reason for stacking is to allow for greater model complexity. Deep RNN work better than shallower networks, as presented by [44], a multiple-layer deep architecture was better for machine-translation in an encoder-decoder framework and authors in [45] also showed improved results by using an architecture with several stacked recurrent layers for RNN.

3.3. Proposed Methodology

This work uses DRNN for hourly variations of the electricity consumption prediction using weather, type of day, day of week and an autoregressive variable (AR) as input in the training and testing dataset. The historical data used as input in the networks is selected based on the observed results. The best-obtained results use the past two weeks of hourly consumption as input in the neural networks. Shorter lag periods increased the MAPE. This aspect means that daily patterns exist and repeat weekly. For the AR method, the lag was selected based on the p -value analysis.

Autoregressive method (AR) is a regression implementation of time series data to predict future values based on past correlations. Two weeks of hourly data was analysed and based on the p -value the relevant past data was used in the forecast Equation (7). To forecast hours ($h_{1..24}$) in day ($d + 1$) we consider hours ($h_{1..24}$) in days ($d - 1, d - 2 \dots d - 14$)

$$\hat{Y}_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \dots + \beta_{14} y_{t-14}. \quad (7)$$

Based on the regression p -value scoring presented in Table 2, we keep the past hours that are relevant for the regression Equation (7).

Table 2. Regression Statistics.

		Coeff.	Std. Error	t Stat	p-Value	Lower 95%	Upper 95%	Lower 95.0%	Upper 95.0%
	β_0	0.41	0.039	6.823	0.0000000	0.157	0.31	0.17	0.401
T-1	β_1	0.311	0.0121	31.86	0.0000000	0.46	0.402	0.3	0.32
T-2	β_2	−0.016	0.012	−1.277	0.2015198	−0.040	0.008	−0.040	0.008
T-3	β_3	0.039	0.012	3.220	0.0012885	0.015	0.063	0.015	0.063
T-4	β_4	0.071	0.012	5.791	0.0000000	0.047	0.095	0.047	0.095
T-5	β_5	−0.008	0.012	−0.653	0.5135898	−0.032	0.016	−0.032	0.016
T-6	β_6	0.025	0.012	2.092	0.0364606	0.002	0.048	0.002	0.048
T-7	β_7	0.576	0.011	51.288	0.0000000	0.554	0.598	0.554	0.598
T-8	β_8	−0.295	0.012	−25.119	0.0000000	−0.318	−0.272	−0.318	−0.272
T-9	β_9	−0.013	0.012	−1.087	0.2769461	−0.037	0.011	−0.037	0.011
T-10	β_{10}	−0.040	0.012	−3.244	0.0011830	−0.064	−0.016	−0.064	−0.016
T-11	β_{11}	−0.071	0.012	−5.792	0.0000000	−0.095	−0.047	−0.095	−0.047
T-12	β_{12}	0.006	0.012	0.497	0.6189813	−0.018	0.030	−0.018	0.030
T-13	β_{13}	−0.023	0.012	−1.920	0.0548434	−0.046	0.000	−0.046	0.000
T-14	β_{14}	0.303	0.011	28.200	0.0000000	0.282	0.324	0.282	0.324

All the previous steps with a p -value score greater than 0.05 are removed from the regression equation. The forecast $\hat{Y}_t$ generated by AR(9) is given as input in the training of the GRU network. Recurrent neural networks consisting of more than two hidden layers are called deep recurrent neural networks (DRNN). The main feature of DRNN is that each hidden state is continuously passed to the next time step of the current layer and the next layer of the current time step. The hidden state of the hidden layer $H^\ell \in \mathbb{R}^{n \times h}$, the output layer variable $Y_t \in \mathbb{R}^{n \times o}$, and the hidden layer activation. The hidden state of the hidden layer ℓ is computed with Equation (8) and the output of the DRNN with Equation (9).

$$H_t^\ell = \tanh(H_t^{\ell-1}W_{xh}^\ell + H_{t-1}^\ell W_{hh}^\ell + b_h^\ell) \quad (8)$$

$$Y_t = H_t^{(L)}W_{ho} + b_o \quad (9)$$

o : number of outputs

$W_{xh}^\ell \in \mathbb{R}^{h \times h}$, $W_{hh}^\ell \in \mathbb{R}^{h \times h}$ and $W_{ho} \in \mathbb{R}^{h \times o}$: weight parameters

$b_h^\ell \in \mathbb{R}^{1 \times h}$ and $b_o \in \mathbb{R}^{1 \times o}$: bias parameter

Considering the mentioned equations, we present in Figure 5 the implementation framework for the deep learning algorithms. All the training variables are used to learn long-term dependencies. In the testing phase, the DRNN model takes as inputs the past three days of hourly consumption, AR(9) hourly forecast for the day ($d + 1$), and the exogenous variables for the day ($d + 1$). The output represents the hourly forecast for the next day (24 h).

Forecasting based on the test dataset is implemented on a day ahead approach with the variables presented in Figure 5. We forecast each day once for 24 h and compare it with the actual consumption. This process is repeated for the entire test data without turning the DRNN algorithms or other ML algorithms used in this work.

The historical data used as input in the networks is selected based on the observed results. The best-obtained results use the past two weeks of hourly consumption as input in the neural networks. Shorter lag periods increased the MAPE. This aspect means that daily patterns exist and repeat weekly. For the AR method, the lag was selected based on the p -value analysis. The exogenous variable considered in forecasting have direct influence in electricity consumption.

Figure 5. The proposed framework for the implementation of deep learning for hourly load curves.

4. Results

The scope of this work is to identify the best solution for industrial load forecasting. For this reason, several algorithms were analyzed and implemented using the tools mentioned in Section 3 to have a solid comparison standpoint. For the evaluation of the forecast, the metrics applied for the results obtained in the test dataset using the equations in Table 3.

Table 3. Forecast errors metrics.

Mean Absolute Error	Root Mean Square Error	Mean Absolute Percentage Error
$MAE = \frac{\sum_{t=1}^n Y_t - \hat{Y}_t }{n}$	$RMSE = \sqrt{\frac{\sum_{t=1}^n (Y_t - \hat{Y}_t)^2}{n}}$	$MAPE = \frac{\sum_{t=1}^n \frac{ Y_t - \hat{Y}_t }{Y_t} * 100}{n}$

n = number of observations; $\hat{Y}_t$ = forecasted value; Y_t = actual value.

The forecasting methods used in this article and presented in Table 4, together with the hyperparameters for each ML algorithm. The AR method is implemented as well to showcase the load forecast from a traditional perspective. The standalone AR method for forecasting has the same order as the one used in the hybrid method.

Table 4. Parameters used by the forecast methods implemented in this work.

Method	Parameters Considered
AR	Autogressive prediction for each hour is based on the same hour in the past 14 days. Coefficients are presented in Table 2
MLP	Multi-Layer Perceptron. Input matrix [24,11]. Input variable: Past 14 days, Day of week, Working/non-working day and hours, special days, temperature. 2 × hidden layers (300, 200). Output layer: Dense; Activation: Relu; Optimizer: Adam; Loss: MSE; Epochs: 100
Simple RNN	Recurrent neural network. Input matrix [24,11]. Input variable : Past 14 days, Day of week, Working/non-working day and hours, Special days, temperature, humidity, dew point. 3 × hidden layers (100, 100, 96). Output layer: Dense; Activation: Tanh, Sigmoid; Optimizer: Adam; Loss: MSE; Epochs: 100
LSTM	Long short-term memory. Input matrix [24,11]. Input variable: Past 14 days, Day of week, Working/non-working day and hours, special days, temperature, humidity, dew point. 3 × hidden layers (100, 100, 168). Output layer: Dense; Activation: Tanh, Sigmoid; Optimizer: Adam; Loss: mean square error; Epochs: 100
LSTM encoder-decoder	Input matrix [24,11]. Input variable: Past 14 days, Day of week, Working/non-working day and hours, special days, temperature, humidity, dew point. 3 × hidden layers (100, 100, 100), 1 × Repeat Vector, 1 × Time distributed layer (96). Activation: Tanh, Sigmoid; Optimizer: Adam; Loss: mean square error; Epochs: 100
GRU	Gated recurrent unit. Input matrix [24,11]. Input variable: Past 14 days, Day of the week, Working/non-working day and hours, special days, temperature, humidity, dew point. 3 × hidden layers (100, 100, 48). Output layer: Dense; Activation: Tanh, Sigmoid; Optimizer: Adam; Loss: mean square error; Epochs: 100
GRU—LSTM	Combination of LSTM and GRU layers. Input matrix [24,11]. Input variable: Past 14 days, AR(9), Day of week, Working/non-working day and hours, special days, temperature, humidity, dew point. 3 × hidden layers (100, 100, 48). Output: Dense; Activation: Tanh, Sigmoid; Optimizer: Adam; Loss: MSE; Epochs: 100

The calculated errors according to Table 3 are presented in Table 5 for each algorithm implemented, highlighting the best results. The results represent the lowest error obtained from multiple implementations for the ML architectures considered in this study. The errors are calculated on a hourly basis and for the entire testing period. The GRU model that uses the feature mentioned in Table 4 as input obtains the best MAPE of 4.82% better than all the other DL algorithms. This approach improved the GRU method by 6.48% and 11.88% compared to the AR method. The LSTM method scored a higher error in comparison with GRU and close to the AR method. The AR result of 5.53% MAPE illustrates that the hourly load curve has a strong repetitive pattern, and the LSTM is overfitting on the training data. The MLP and simple RNN scored higher error than the AR, and the LSTM encoder-decoder has the highest error of 6.28%.

Table 5. Forecast errors for the test dataset: 15 october–20 december.

	AR(9)	LSTM enc dec	LSTM-GRU	LSTM	MLP	Simple RNN	GRU
MAPE	5.53%	6.28%	5.14%	5.43%	5.71%	6.63%	4.82%
RMSE	0.146	0.193	0.138	0.141	0.165	0.181	0.131
MAE	0.112	0.145	0.1001	0.104	0.129	0.140	0.0998

In Figure 6, a normal working week (Monday to Friday) is presented to highlight the improvement obtained by using the proposed methodology. It can be observed that using

the AR component it helps when the behaviour of the consumer is repetitive and improves the forecast.

Figure 6. Comparison between the actual and the forecasted values with GRU, GRU+LSTM and AR methods.

Other ML algorithms have good results, but none score better than the GRU. In Figure 7 high variations can be observed in the peak hours. Simple RNN scored the highest MAPE 6.63%, even worse than the autoregressive model. The LSTM encoder-decoder is a sequence to the sequence learning algorithm, performed worse than the LSTM by 14.89% because more trainable weights for each of the encoder's time steps are required. The consequence of this situation is a large number of parameters if the input data for the encoder is a long time series, even more parameters than the LSTM. A complex network needs a longer time for training, and the underfitting problem can cause the LSTM encoder-decoder to perform worse than the LSTM. Because we implemented a similar structure for all the RNN networks to offer a solid comparison benchmark, we can conclude that the LSTM encoder-decoder could improve forecasting results if the complex network would be trained for a longer period with a more powerful hardware resource.

Figure 7. Comparison between the actual and the forecasted values with AR, LSTM encoder-decoder, LSTM, MLP and simple RNN methods.

In article [46], the authors compare LSTM with GRU on text datasets, and conclude that through empirical research, the advantage of GRU is relevant in the scenario of small datasets. In other scenarios, compared with LSTM, the performance loss of GRU decreases. GRU can forget and choose memory with one gate, and fewer parameters, while LSTM needs to use more gates and more parameters to complete the same task. For our scenario, with short-term industrial load forecasting in Figure 8, the GRU network offers better result than the LSTM and combined GRU and LSTM networks Figure 9. Because in the training stage the same architecture for the RNN networks were used, the LSTM is underfitted in

the training stage. Building larger and training for more epochs did not improve the errors and the network failed due to overfitting on the training data.

Figure 8. Comparison between the actual and the forecasted values with GRU.

LSTM-GRU network improves the overall errors, because of the GRU layer, but there is a clear pattern of high variations for peak values and off-peak period. The authors in [47] could find a clear difference between LSTM and GRU, and suggest that the selection of the type of gated recurrent unit depend on the dataset and corresponding task. In our case, with industrial load forecasting, the results are clearly in favor of GRU.

Figure 9. Comparison between the actual and the forecasted values with LSTM-GRU and LSTM.

For a better understanding of the evolution of the GRU results for the testing period in Figure 10 the actual and forecasted load curve can be correlated with the daily MAPE. The best daily MAPE is 1.58%, but the worst error value is 25.38%. The high error was caused by two legal holiday periods from 30 November to 1 December 2019. The GRU fails to correctly identify the evolution of the actual consumption because in the training dataset, this period is not seen by the network, and the GRU can not generalize on such a situation. On top of this, the previous weekend, an uncommon event occurred due to probable losses in the compress air system.

Figure 10. Forecast evaluation for the GRU implementation.

The process of using DL algorithms is stochastic, and with different MAPE values scored by using the same architecture, is difficult to fine-tune the hyperparameters of the neural networks to generalize efficiently on the training data. For this reason, in Figure 11 various implemented architectures are analyzed for the best method implemented in this article. The results indicate that using a complex architecture can lead to higher errors due to overfitting the training data. The lowest forecasting error scored by the GRU model (MAPE 4.82%) was reached by using a network with three hidden layers (GRU 24|3|100|100|48|24). It can be observed that the training MAPE error keeps decreasing, the test error increasing, indicating that the network is overfitting and fails to generalize on the new data. This situation needs to be quantified to find the structure that offers the lowest errors.

A simple indicator was defined to quantify the complexity of the DL algorithm in Equation (10) and to compare the results of the GRU algorithm.

$$DL_{index(i)} = \frac{E_i \times N_{pi}}{E_{max} \times N_{pmax}} \quad (10)$$

where i is the number of analyzed architectures, E is the number of epochs, and N_p is the number of total parameters used in the DL architecture (weight and biases). E_{max} is the maximum number of epoch used in all the simulations, and N_{pmax} is the maximum number of parameters in all simulations. For each simulation, the (DL_{index}) is compared with the MAPE for both datasets (train and test) and training time to make sure the DL does not overfit on the training data. In Figure 12, the evolution of error concerning epochs, time of training, and the complexity of the DNN (DL_{index}) are highlighted. Training complex architectures with a high number of epochs, besides time and resources, leads to higher errors because of overfitting. A minimal error can determine the selection of the best algorithm.

The (DL_{index}) highlights that increasing the number of hidden layers and neurons in each layer can negatively impact the performance of the DL algorithms. It becomes computationally harder to obtained lower errors and the training time increases unjustifiably. Training a complex deep network will determine that each layer can describe a precise features in the relation between input and output, but the neural network will fail to generalize to new data needed for forecasting and later for electricity acquisition.

Figure 11. The various structures for the analysed GRU deep network.

Figure 12. Comparison of the DL_{index} with the training time, number of epochs and MAPE.

5. Discussion

Suppliers want clients to use more energy rather than less, but the recent development of demand-side management, demand response, and smart grid [48,49] will shift the status quo from irresponsible behavior towards indispensable accurate predictions. According to [50], the industrial and commercial sector represents 63.46% of the world's total electricity consumption. The challenge is to anticipate the stochastic behavior of large consumers.

Industrial load forecast plays an essential role in the cost of electricity, especially for a large consumer such as the one presented in this work. Load forecasting is important because planning for electricity supply is dependent on consumption forecasts. High imbalances generated between actual and forecast consumption will create a higher risk for all the participants in the power market. This work proposes a methodology for industrial load forecasting based on DL and AR. The result and analysis indicate that DL has a high order of stochasticity, and careful tuning of the parameters is needed. The challenge is to anticipate the stochastic behavior of large consumers. This work proves that deep neural networks can successfully forecast hourly industrial load. The obstacle to this approach is the lack of smart-metering and sensors for forecasting consumption in real-time. Without relevant data, efficient and replicable forecasting models do not represent a reliable investment for the private sector. The novelty of the work is the proposed framework applied for industrial load curves, the analysis of the best architecture, and the scalability of the deep neural networks using a simple complexity index. The study compared the forecast performance for seven methods and tested various combinations for forecast variables and lag structures. Our test sample results across 1608 hourly values (15 October–20 December 2019) indicate consistently that: (i) deep recurrent neural networks are suitable for industrial load consumption; and (ii) the best model implemented for is GRU. The work highlights that increasing the number of hidden layers and neurons in each layer (total parameters) can negatively impact the performance of the DL algorithms. Training a complex deep network, each layer can map exactly the relation between input and output,

but the neural network will fail to generalize to new data needed for forecasting and later for electricity acquisition.

6. Conclusions

A compromise is needed to find a practical solution to make electric load forecasting more accessible to the industry sector by implementing algorithms that learn directly from data with little human intervention. The novelty of the work is the proposed framework applied for industrial load curves, the analysis of the best architecture, and the scalability of the deep neural networks using a simple complexity index. The study compared the forecast performance for seven methods and tested various combinations for forecast variables and lag structures. Our test sample results across 1608 hourly values (15 October–20 December 2019) indicate consistently that: (i) deep recurrent neural networks are suitable for industrial load consumption; and (ii) the best model implemented for is GRU. The work highlights that increasing the number of hidden layers and neurons in each layer can negatively impact the performance of the DL algorithms.

Author Contributions: Conceptualization, S.U.; methodology, S.U.; software, S.U.; validation, A.C.C. and V.T.; formal analysis, A.C.C. and V.T.; investigation, S.U.; resources, S.U.; data curation, S.U.; writing—original draft preparation, S.U.; writing—review and editing, A.C.C.; visualization, S.U.; supervision, A.C.C. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was funded by Technical University of Cluj-Napoca.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Datasets values and exogenous variables are made available at <https://drive.google.com/file/d/1X1P7m3i1FrlrAIGlxNnhZKmgpYVeSi2P/view?usp=sharing> (accessed on 26 October 2021). Hourly climatic data is obtained from the website: [https://rp5.ru/Vremea_%C3%AEn_Baia_Mare_\(aeroport\)](https://rp5.ru/Vremea_%C3%AEn_Baia_Mare_(aeroport)) (accessed on 26 October 2021).

Acknowledgments: This paper was supported by the Project POCU/380/6/13/123927, “Entrepreneurial competencies and excellence research in doctoral and postdoctoral programs- ANTREDOC”, the project co-funded by the European Social Fund.

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

References

- Hernandez, L.; Baladron, C.; Aguiar, J.M.; Carro, B.; Sanchez-Esguevillas, A.J.; Lloret, J.; Massana, J. A Survey on Electric Power Demand Forecasting: Future Trends in Smart Grids, Microgrids and Smart Buildings. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1460–1495. [CrossRef]
- Chambers, J.C.; Mullick, S.K.; Smith, D.D. *How to Choose the Right Forecasting Technique*; Magazine; Harvard University, Graduate School of Business Administration: Cambridge, MA, USA, 1971.
- Armstrong, J.S. Selecting Forecasting Methods. *Princ. Forec. Int. Ser. Oper. Res. Manag. Sci.* **2001**, *30*, 365–386. [CrossRef]
- Archer, B.H. Forecasting demand: Quantitative and intuitive techniques. *Int. J. Tour. Manag.* **1980**, *1*, 5–12. [CrossRef]
- Feinberg, E.A.; Genethliou, D. Load Forecasting. In *Applied Mathematics for Restructured Electric Power Systems: Optimization, Control, and Computational Intelligence*; Chow, J.H., Wu, F.F., Momoh, J., Eds.; Springer: Boston, MA, USA, 2005; pp. 269–285. [CrossRef]
- Nti, I.K.; Teimeh, M.; Nyarko-Boateng, O.; Adekoya, A.F. Electricity load forecasting: a systematic review. *J. Electr. Syst. Inf. Technol.* **2020**, *7*, 13. [CrossRef]
- Gravesteijn, B.Y.; Nieboer, D.; Ercole, A.; Lingsma, H.F.; Nelson, D.; Van Calster, B.; Steyerberg, E.W. Machine learning algorithms performed no better than regression models for prognostication in traumatic brain injury. *J. Clin. Epidemiol.* **2020**, *122*, 95–107. [CrossRef]
- Kuster, C.; Rezgui, Y.; Mourshed, M. Electrical load forecasting models: A critical systematic review. *Sustain. Cities Soc.* **2017**, *35*, 257–270. [CrossRef]
- Green, K.C.; Armstrong, J.S. Simple versus complex forecasting: The evidence. *J. Bus. Res.* **2015**, *68*, 1678–1685.
- Armstrong, J.; Green, K.; Graefe, A. Golden rule of forecasting: Be conservative. *J. Bus. Res.* **2015**, *68*, 1717–1731. [CrossRef]
- Hyndman, R.J. A brief history of forecasting competitions. *Int. J. Forec.* **2020**, *36*, 7–14. [CrossRef]

12. Makridakis, S.; Spiliotis, E.; Assimakopoulos, V. The M4 Competition: 100,000 time series and 61 forecasting methods. *Int. J. Forec.* **2020**, *36*, 54–74. [\[CrossRef\]](#)
13. Lara-Benítez, P.; Carranza-García, M.; Riquelme, J.C. An Experimental Review on Deep Learning Architectures for Time Series Forecasting. *Int. J. Neural Syst.* **2021**, *31*, 2130001. [\[CrossRef\]](#)
14. Sehovac, L.; Grolinger, K. Deep Learning for Load Forecasting: Sequence to Sequence Recurrent Neural Networks With Attention. *IEEE Access* **2020**, *8*, 36411–36426. [\[CrossRef\]](#)
15. Marvuglia, A.; Messineo, A. Using Recurrent Artificial Neural Networks to Forecast Household Electricity Consumption. *Energy Procedia* **2012**, *14*, 45–55. [\[CrossRef\]](#)
16. He, W. Load Forecasting via Deep Neural Networks. *Procedia Comput. Sci.* **2017**, *122*, 308–314. [\[CrossRef\]](#)
17. Eskandari, H.; Imani, M.; Moghaddam, M. Convolutional and recurrent neural network based model for short-term load forecasting. *Electr. Power Syst. Res.* **2021**, *195*, 107173. [\[CrossRef\]](#)
18. Hewamalage, H.; Bergmeir, C.; Bandara, K. Recurrent Neural Networks for Time Series Forecasting: Current status and future directions. *Int. J. Forec.* **2021**, *37*, 388–427. [\[CrossRef\]](#)
19. 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. [\[CrossRef\]](#)
20. Learning Libraries. Tensorflow. Available online: <https://www.tensorflow.org/> (accessed on 10 September 2021).
21. Learning Libraries. Keras. Available online: <https://keras.io/> (accessed on 16 September 2021).
22. Learning Libraries. Scikit-Learn. Available online: <https://scikit-learn.org/stable> (accessed on 8 September 2021).
23. Learning Libraries. Numpy. Available online: <https://numpy.org/> (accessed on 8 September 2021).
24. Learning Libraries. Seaborn. Available online: <https://seaborn.pydata.org/> (accessed on 8 September 2021).
25. Dickey, D.A.; Fuller, W.A. Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *J. Am. Stat. Assoc.* **1979**, *74*, 427–431. [\[CrossRef\]](#)
26. Ray, S. A Quick Review of Machine Learning Algorithms. In Proceedings of the 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, India, 14–16 February 2019; pp. 35–39. [\[CrossRef\]](#)
27. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: <http://www.deeplearningbook.org> (accessed on 28 July 2021).
28. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)
29. Cho, K.; van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:cs.CL/1406.1078.
30. Gonçalves, J.N.; Cortez, P.; Carvalho, M.S.; Frazão, N.M. A multivariate approach for multi-step demand forecasting in assembly industries: Empirical evidence from an automotive supply chain. *Dec. Support Syst.* **2021**, *142*, 113452. [\[CrossRef\]](#)
31. Yan, R.; Liao, J.; Yang, J.; Sun, W.; Nong, M.; Li, F. Multi-hour and multi-site air quality index forecasting in Beijing using CNN, LSTM, CNN-LSTM, and spatiotemporal clustering. *Expert Syst. Appl.* **2021**, *169*, 114513. [\[CrossRef\]](#)
32. Kim, T.Y.; Cho, S.B. Predicting residential energy consumption using CNN-LSTM neural networks. *Energy* **2019**, *182*, 72–81. [\[CrossRef\]](#)
33. Rafi, S.H.; Nahid-Al-Masood; Deeba, S.R.; Hossain, E. A Short-Term Load Forecasting Method Using Integrated CNN and LSTM Network. *IEEE Access* **2021**, *9*, 32436–32448. [\[CrossRef\]](#)
34. Cioffi, R.; Travagliani, M.; Piscitelli, G.; Petrillo, A.; De Felice, F. Artificial Intelligence and Machine Learning Applications in Smart Production: Progress, Trends, and Directions. *Sustainability* **2020**, *12*, 492. [\[CrossRef\]](#)
35. Almalag, A.; Edwards, G. A Review of Deep Learning Methods Applied on Load Forecasting. In Proceedings of the 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), Cancun, Mexico, 18–21 December 2017; pp. 511–516. [\[CrossRef\]](#)
36. Zor, K.; Timur, O.; Teke, A. A state-of-the-art review of artificial intelligence techniques for short-term electric load forecasting. In Proceedings of the 2017 6th International Youth Conference on Energy (IYCE), Budapest, Hungary, 21–24 June 2017; pp. 1–7. [\[CrossRef\]](#)
37. Zhang, L.; Wen, J.; Li, Y.; Chen, J.; Ye, Y.; Fu, Y.; Livingood, W. A review of machine learning in building load prediction. *Appl. Energy* **2021**, *285*, 116452. [\[CrossRef\]](#)
38. Lu, K.; Meng, X.R.; Sun, W.X.; Zhang, R.G.; Han, Y.K.; Gao, S.; Su, D. GRU-based Encoder-Decoder for Short-term CHP Heat Load Forecast. *Mater. Sci. Eng.* **2018**, *392*, 062173. [\[CrossRef\]](#)
39. Jiao, R.; Zhang, T.; Jiang, Y.; He, H. Short-Term Non-Residential Load Forecasting Based on Multiple Sequences LSTM Recurrent Neural Network. *IEEE Access* **2018**, *6*, 59438–59448. [\[CrossRef\]](#)
40. Aguilar Madrid, E.; Antonio, N. Short-Term Electricity Load Forecasting with Machine Learning. *Information* **2021**, *12*, 2005. [\[CrossRef\]](#)
41. Wu, D.C.; Bahrami Asl, B.; Razban, A.; Chen, J. Air compressor load forecasting using artificial neural network. *Expert Syst. Appl.* **2021**, *168*, 114209. [\[CrossRef\]](#)
42. Zheng, J.; Xu, C.; Zhang, Z.; Li, X. Electric load forecasting in smart grids using Long-Short-Term-Memory based Recurrent Neural Network. In Proceedings of the 2017 51st Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 22–24 March 2017; pp. 1–6. [\[CrossRef\]](#)

43. Nguyen, H.; Tran, K.; Thomassey, S.; Hamad, M. Forecasting and Anomaly Detection approaches using LSTM and LSTM Autoencoder techniques with the applications in supply chain management. *Int. J. Inf. Manag.* **2021**, *57*, 102282. [[CrossRef](#)]
44. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to Sequence Learning with Neural Networks. *CoRR* **2014**, *2014*, 3104–3112.
45. Irsoy, O.; Cardie, C. Deep Recursive Neural Networks for Compositionality in Language. In *Advances in Neural Information Processing Systems*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q., Eds.; Curran Associates, Inc.: New York, NY, USA, 2014; Volume 27.
46. Yang, S.; Yu, X.; Zhou, Y. LSTM and GRU Neural Network Performance Comparison Study: Taking Yelp Review Dataset as an Example. In Proceedings of the 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI), Shanghai, China, 12–14 June 2020; pp. 98–101. [[CrossRef](#)]
47. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:cs.NE/1412.3555.
48. Tazi, K.; Abdi, F.; Abbou, M.F. Demand and Energy Management in Smart Grid: Techniques and Implementation. In Proceedings of the 2017 International Renewable and Sustainable Energy Conference (IRSEC), Tangier, Morocco, 4–7 December 2017; pp. 1–6. [[CrossRef](#)]
49. Liu, R.; Liu, Y.; Jing, Z. Impact of industrial virtual power plant on renewable energy integration. *Glob. Energy Interconn.* **2020**, *3*, 545–552. [[CrossRef](#)]
50. Agency, I.E. *World Electricity Final Consumption by Sector, 1974–2018*; IEA: Paris, France. Available online: <https://www.iea.org/data-and-statistics/charts/world-electricity-final-consumption-by-sector-1974-2018> (accessed on 6 October 2021).