Dariusz RUCIŃSKI[1]
ORCID: 0000-0001-5458-9170

[1] PhD Student at Institute of Computer Science
Siedlce University of Natural Sciences and Humanities
Faculty of Exact and Natural Sciences
Institute of Computer Science
ul. 3 Maja 54, 08-110 Siedlce, Poland

Prediction capabilities of the LSTM and Perceptron models based on the Day-Ahead Market on the Polish Power Exchange S.A.

DOI: 10.34739/si.2023.28.04

Abstract. The main purpose of the research was to examine the properties of models for two kinds of neural networks, a deep learning models in which the Long Short-Term Memory was chosen and shallow neural model in which the Perceptron Neural Network was chosen. The subject of the examination was the Day-Ahead Market system of PPE S.A. The article presents the learning results of both networks and the results of the predictive abilities of the models. The research was conducted based on data published on the Polish Stock Exchange for the 2018 year. The MATLAB environment was chosen as a tool for providing the examinations. The determination index (R^2) and the mean square error (MSE) was adopted as the network evaluation criterion for the learning ability and for the prediction ability of both networks.

Keywords. Shallow networks, Deep networks, Day-Ahead Market, MATLAB and Simulink environment, Neural Modeling, Prediction Time, Electricity Prices.

1. Characteristics of the research tools

Artificial intelligence methods constitute a wide range of tools used in many fields of economy and science, their description and application can be found, for example, in [11]. One of the methods is artificial neural networks, which, starting from the work of McCulloch and Pitts [2], and from that time experiencing dynamic development. Since 1943, many types of neural networks have been created, which are used depending on the properties of the system that is modeled with them. They differ depending on the type of data processed, architecture, learning method (with a teacher or without a teacher), etc.

Recently we observe the rapid development of deep networks, which was initiated by the work [3]. For this paper, the classic model of neural networks, which is now called the shallow learning method, was compared with a model based on the Long Short-Term Memory (LSTM) network, considered as a one of the deep networks.

The Multi-Layer Perceptron (MLP) model considered as a shallow learning model is characterized by a shorter learning time compared to deep networks. Longer time of learning of deep networks is a result of the concept of the construction of such networks. Both models are trained on the principle of minimizing the error between the values obtained from the model and the Day Ahead Market (DAM) system.

The architecture of deep networks is more complex, it consists of more layers and individual layers have their specific functions in a such network, they play a specific role. The networks have a modular structure, i.e., they can be built from individual layers that play a specialized role, e.g., input layer, normalization layer, convolutional layer, LSTM, etc. Deep networks are currently used for pattern recognition, text processing and error localization, regression, etc.

The aim of the work was to examine the ability to model the Day-Ahead Market (DAM) on the Polish Power Exchange (PPE) based on a shallow network, in this case, MLP, and a deep network, in this case, the Long Short-Term Memory network. The MLP network and various aspects of its construction, operation, and properties have already been the subject of the author's earlier works [9-10].

The Long Short-Term Memory (LSTM) network first time was proposed in [4], it is an extension of the Recursive Neural Network (RNN) network and proposes to solve the known problem of these networks, i.e. 'vanish' - when the weights of the feedback values are smaller than the values of one mean their next powers are smaller and smaller, so its influence on the output values is minimized, or 'explosion' - when the values of the feedback weights are greater than the value of one, their next powers are getting bigger, so its influence on the output values is too big.

The LSTM network consists of two 'buses' carrying two types of values, Long-Term Memory (LM) and Short-Term Memory (SM), which interact through the so-called goals.

The first Long-Term Memory (LM) has no weights or biases to modify its values, resulting in no known negative RNN 'vanish' and 'explosion' phenomena. The LM values are modified in two places, the forget gate and the input gate. The second Short-Term Memory (SM) is directly connected to the weights and biases of individual networks, which modify its values, however, in the output gate, the value of the LM memory has a significant impact on the value of the SM bus.

LSTM is a structure consisting of several specialized networks. Depending on the role played by a given network, two types of sigmoidal activation functions are used:

– logsig(), which returns values in the range (0÷1):

$$logsig(n) = \frac{e^n}{1 + e^n}, \tag{1}$$

where:

n- argument of the activation function,
e – Euler’s number.

– tanh() which returns values (-1 ÷1):

$$tanh(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}}. \tag{2}$$

Figure 1 shows the idea of the LSTM network operation. The general concept of the network boils down to the principle that the values for individual gates are determined by activation functions determined as hyperbolic tangent and then proportionally corrected by activation functions determined as logistic functions.

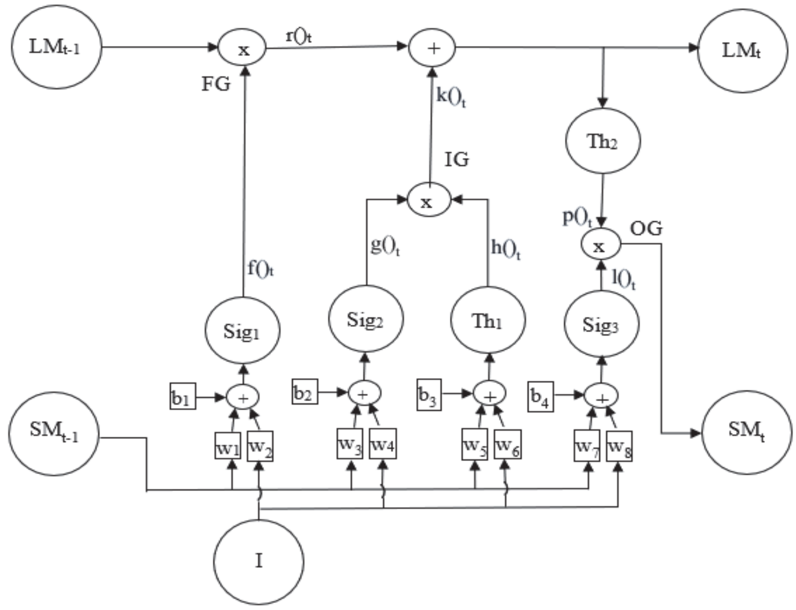


Figure 1. Construction diagram of one LSTM network module. Markings in the text. Own source based on [25].

The forget gate (FG) - is the first block represented in the LSTM architecture. The values from the current input (I) and the previous hidden state (SM_{t-1}) are multiplied by their weights (**w_1** and **w_2**), summed and bias (**b_1**) is added, their sum is the argument for the first sigmoidal activation function (Sig_1). The output values ($f()_t$) are multiplied by the Hadamard product with the input values (LM_{t-1}). If the network output values from the network represented by ($f()_t$) are closer to zero then the result of Hadamard product decried the LM bus (which corresponds to the phenomenon of forgetting), if the network output values represented by ($f()_t$) are closer to one, then on the LM bus change is a little. So, the output value ($f()_t$) determines how much of the LM_{t-1} memory will be forgotten.

$$f()_t = logsig(\boldsymbol{w}_1\boldsymbol{I} + \boldsymbol{w}_2 SM_{t-1} + \boldsymbol{b}_1), \tag{3}$$

$$r()_t = LM_{t-1} \odot f()_t. \tag{4}$$

Input gate (IG) - works as a mechanism that takes into account the state of the ($r()_t$) value from and the current input value (I) and the previous the previous hidden state (SM_{t-1}) value. It also consists of two modules:

First: the values from the current input (**I**) and the previous hidden state (SM_{t-1}) are multiplied by their weights (**w_3** and **w_4**) the bias (**b_2**) is added, their sum is the argument for the second sigmoidal activation function (Sig_2), and in result we get the values ($g()_t$).

Second: the values from the current state (**I**) and the previous hidden state (SM_{t-1}) are multiplied by their weights (**w_5** and **w_6**) the bias (**b_3**) is added, and their sum is the argument for the first activation function (Th_1) as a result we get values ($h()_t$).

The value ($h()_t$) is a potential value that is added to the LM bus, while the value ($g()_t$) determines (similarly to the FG gate) what part of the value ($h()_t$) will be taken into account when changing the LM value ($k(\,)_t$) in gate (IG). The value of ($k()_t$) can be negative or positive depending on the value of the function ($h()_t$), the activation function ($g()_t$) is always positive.

$$g()_t = logsig(\boldsymbol{w}_3 I + \boldsymbol{w}_4 SM_{t-1} + \boldsymbol{b}_2), \tag{5}$$

$$h()_t = tanh(\boldsymbol{w}_5 \boldsymbol{I} + \boldsymbol{w}_6 SM_{t-1} + \boldsymbol{b}_3), \tag{6}$$

$$k()_t = g()_t \odot H()_{t,} \tag{7}$$

$$LM_t = r()_t + k()_t. \tag{8}$$

Output gate (OG) - works as a mechanism that takes into account the state of the SM value from the previous epoch and the current value of the input to the current LM value. It consists of two parts:

First: the values from the current input (**I**) and the previous hidden state (SM_{t-1}) are multiplied by their weights ($\mathbf{w_7}$ and $\mathbf{w_8}$) the bias ($\mathbf{b_4}$) is added, and their sum is the argument to the third sigmoidal activation function (Sig_3) resulting in the values ($l()_t$).

Second: the value from the current one (LM_t) is the argument for the second activation function (Th_2) resulting in we have the values ($p()_t$).

As in the case of (IG), the value of the function ($p()_t$) is a potential value that is corrected by the proportionality coefficient resulting from the function ($l()_t$) through the Hadamard product. The output value ($q()_t$) is the output SM value denoted as (SM_t).

$$l()_t = logsig(\boldsymbol{w}_7 \boldsymbol{I} + \boldsymbol{w}_8 SM_{t-1} + \boldsymbol{b}_4), \tag{9}$$

$$p()_t = tanh(SM_t), \tag{10}$$

$$SM_t = l()_t \odot p()_t. \tag{11}$$

The concept of such a network structure was proposed in [3], where exemplary calculations based on the proposed network were presented. LSTM network was also use in [5] to forecast the price of electricity of the Day Ahead for the Australian market. An interesting approach of this work was to take into account many factors such as day of the week, time of day, weather conditions, oil prices and price from previous periods. The LSTM network was also used for price forecasting in [1], where it was train to forecast electricity demand, taking into account long-term historical dependencies.

Interesting results were published in [12], where deep learning with several specialized layers was proposed, i.e., the first three modules were used to detect outliers and identify correlated features of electricity price series, the second group of layers was used to extract non-linear features, the third one was used to eliminate errors between forecasted prices and actual prices, the fourth one was used as a probabilistic estimator of the assessment. The usage of the LSTM network was also proposed in [7] where the so-called mechanism of attention was introduced.

The mechanism of attention boils down to the assumption that each learning pair in the network also has an associated hidden vector that represents its hidden features. Presented research was carried out on real data from DAM, showed a difference in the predictive ability of neural networks (LSTM and Perceptron) used to model the system. The predictive abilities of the LSTM network turned out to be better.

As you can see, the use of deep networks in forecasting electricity prices is the subject of many studies and various approaches. In this work, the learning and prediction capabilities of two networks, i.e., the Perceptron ANN and the LSTM network, were compared.

2. Subject of research

The subject of the study is the stock exchange market system, where energy is traded on the power exchange (TGE S.A.) mainly on the Day-Ahead Market (DAM). At TGE S.A. there is also trade in Intraday Market. The DAM system consists of 24-hour quotation settlement periods in which Exchange Members can buy and sell electricity. DAM participants send buy or sell orders for individual hours, based on which the supply curve based on sell orders and the demand curve based on purchase orders are created, respectively.

The research aimed to compare the acceptable time horizon that could be used in prediction based on learned ANN neuronal models of the DAM system. The research was carried out based on data covering the period of operation of DAM in the period from 01/01/2018 to 31/12/2018. The criterion for assessing the ability of individual models to predict volume-weighted average electricity prices was mean square error (MSE) and the regression index (R^2).

The research consisted in evaluating the learning level of both models and checking successive prognostic periods in a rolling manner, i.e., neural networks (Perceptron and LSTM) trained in the system based on the first half of 2018 were the basis for making predictions for subsequent time intervals. Then, the obtained estimation results were subjected to statistical analysis in terms of the repeatability of the obtained results.

3. Research environment

MATLAB and Simulink environment with Deep Learning Toolbox were used for research purposes. Deep Learning Toolbox (DLT) [8] – is a tool that allows for modular building of ANNs by declaring successive layers of learning. These layers are designed to create deep learning networks and are used, among others, for image classification, regression, sequences and training of time series data.

The layers are divided into certain functional groups:

1) Input Layers – In which the sequence input layer is used to input data to a neural network
2) Convolution and Fully Connected Layers - In which the fully connected layer multiplies is used to multiply the input by a weight matrix and then adds a bias vector.
3) Sequence Layers – In which the LSTM layer is used.
4) Activation Layers – In which the hyperbolic tangent (tanh) activation layer applies the tanh function on the layer inputs is used.
5) Normalization Layers – were not used in this research.
6) Utility Layers – were not used in this research.
7) Resizing Layers – was not used in this research.
8) Pooling andUn pooling Layers – were not used in this research.
9) Combination Layers – were not used in this research.
10) Object Detection Layers – were not used in this research.
11) Output Layers – In which the regression layer is used it returns as an output layer the data for the neural network as a RegressionOutputLayer object.

Important from the point of view of computational efficiency in deep networks is also the introduction of the stochastic learning method Adam (Adaptive Moment Estimation) [6]. ADAM method in the learning process requires only the first derivative to estimate the error gradient, which significantly speeds up the computational process in compare to the classical methods where Hessian (second derivative) is required.

4. Assumptions adopted for the research.

Assumptions for the LSTM network

The training data was divided into training and testing data in the following ratio: 80% teaching data, and 20% testing data). The 'mini-batch' variable, which specifies the number of pairs whose output values from the model will be the basis for determining the error value (difference between the values from the model and the training ones) for the backpropagation algorithm, was set to the size of the entire training set, i.e., in this case 182. In the teaching method, a commonly used ADAM learning algorithm was adopted. The LSTM network consists of the input layer 'sequenceInputLayer' with a 24-element input vector representing the normalized volume of energy sold on a given hour in the given day, the next layer is the proper 'LstmLayer' for which the output is the 'fullyConnectedLayer' layer, which in turn is the input to the 'sigmoidlayer'.

The output layer from the network is a 'regressionlaer', which has a 24-element vector at the output. The connection diagram of individual layers is shown in Fig. 1. A brief description of these layers can be found in Chapter 3.

Assumptions for the Perceptron Network

The training data was divided into training, testing, and validation in proportion: 70% training data, 15% testing data, and 15% validation data. The Levenberg-Marquad algorithm was adopted as the network learning method. The Perceptron Network consists of a 24-element hidden layer with a sigmoidal tansig activation function and a 24-element output layer with a linear purelin activation function. Comparing the architecture of Perceptron ANN and Deep Learning Network is shown in Fig. 2. For both networks, the maximum number of learning epochs was set at 1000. Both networks were trained on the same data sets, i.e., from 01.01.2018 to 30.06.2018. Then, the learned networks were used to make predictions for subsequent periods, i.e., from 01.07.2018 to 31.12.2018, according to the same algorithm. i.e., the semi-annual prediction period was shifted successively every 10 days, which in total for the semi-annual period (second half of 2018) gave 18 prediction periods. For example, if the semi-annual training data set was from 01.01.2018 to 30.06.2018. is the first prediction set being in the range from January 10.01. 2018, to 10.07.2018, the next one from 21.01.2018, to 20.07.2018, until 10.08.2018, to 30.12. 2018.

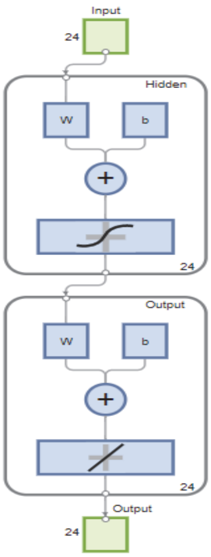
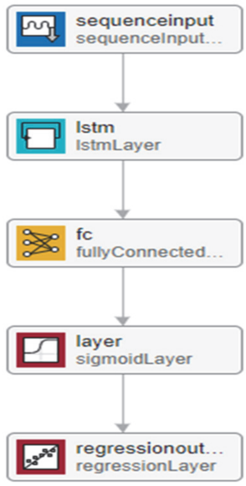
Deep Learning Network	Perceptron Neural Network
	

Figure 2. Architecture of Perceptron ANN and Deep Learning Network. Symbols: Input (24) - input signals to the APS, here: the volume of electricity supplied and sold in each hour of the day (24 signals), Output (24) - output signals from the APS (24 signals, here the volume-weighted average price obtained from the electricity in each hour of the day Description of DLN modules in chapter 3. Source: Own elaboration in MATLAB and Simulink environment [8].

5. Network learning results

As a result of learning the LSTM network, a model representing the RDN TGE system was obtained. Figure 3 shows the network learning process. As you can see, the decrease in the RMSE error (Root Square Mean Error) in the first learning iterations was significant from 0.9 to 0.2, and in subsequent epochs, it was not so significant.

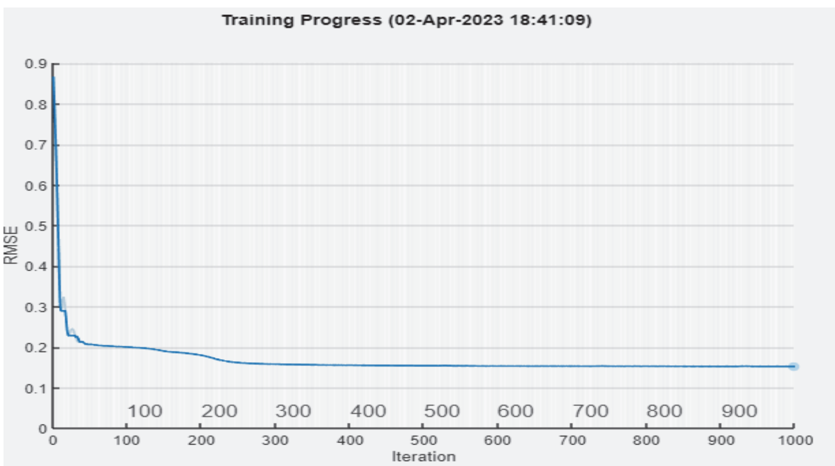


Figure 3. The course of the deep network learning process. Symbols x(Iteration) axis - successive network training iterations, y-axis - RMSE error value for successive training phases. Source: Own elaboration in MATLAB and Simulink environment [8].

The quality of model fitness at the training stage was checked using randomly selected test data. Figure 4 shows the quality of fit of the model to real data based on test data

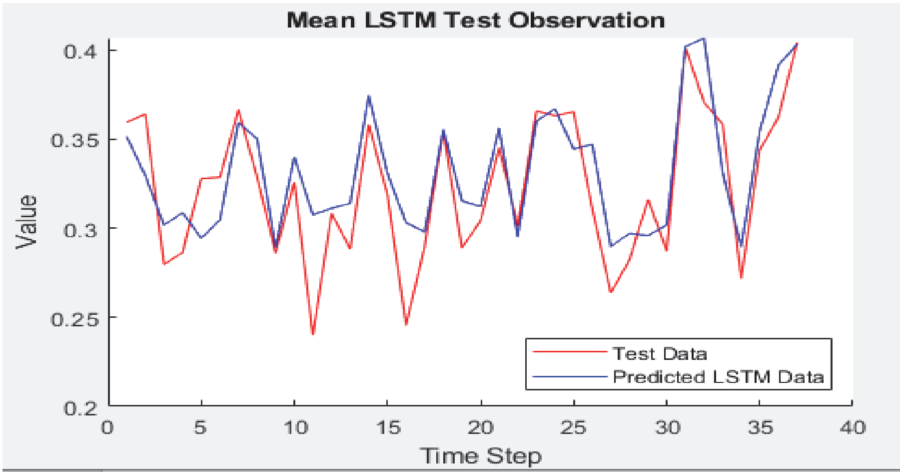


Figure 4. The course of the average fit of the deep network model to the DAM system in successive randomly selected days. Time step - successive randomly selected testing days, Value - the average value of the price for a given day for real data and data received from the network. Source: Own elaboration in MATLAB and Simulink environment [8].

The learning process of the Perceptron ANN is shown in Figure 5. As you can see, the network learned the system much faster than in the case of the LSTM network, because practically already in the third epoch it was learned, which is also caused by a different method of learning the network. The error reduction in the network training process measured as MSE (mean squared error) was two orders of magnitude.

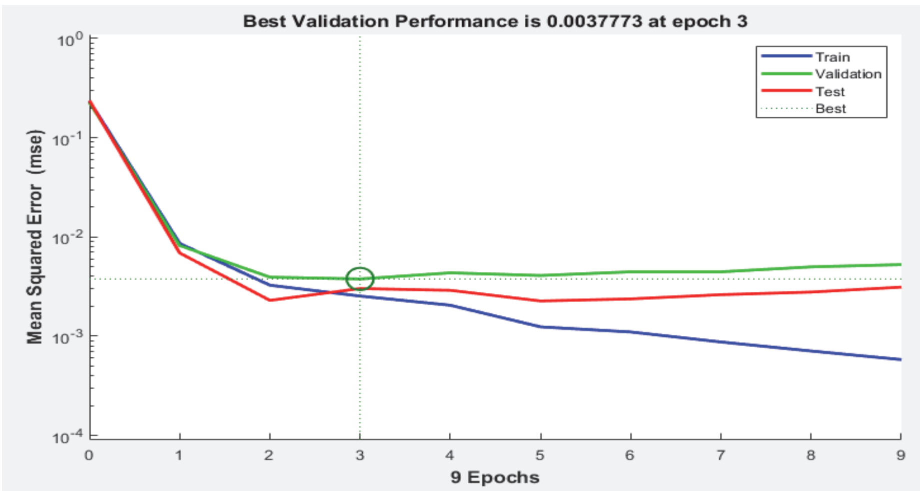


Figure 5. The course of the Perceptron network learning process. Symbols x(epochs) axis - successive iterations of network training, y-axis - MSE error value for successive learning phases. Source: Own elaboration in MATLAB and Simulink environment [8].

As in the case of the LSTM network, the quality of model fit at the training stage was checked using randomly selected test data. Figure 6 shows the quality of model fit to real data based on test data.

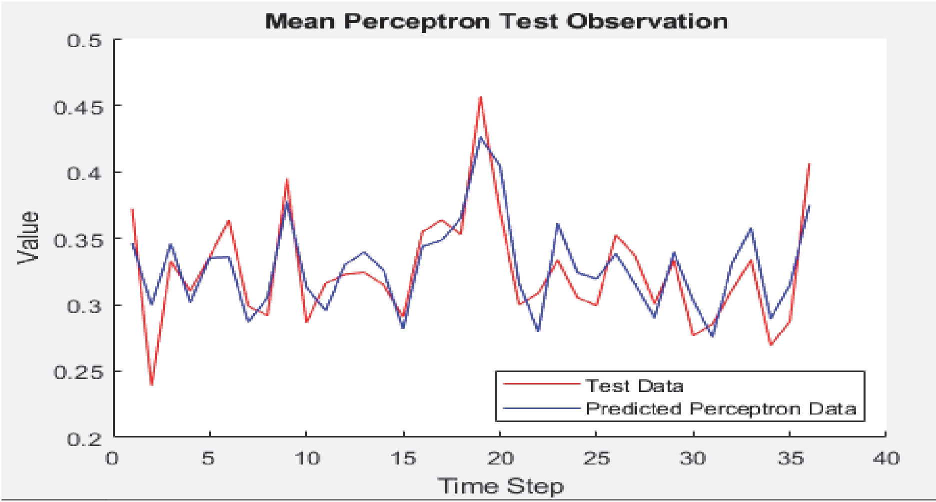


Figure 6. The course of the average fit of the Perceptron network model to the DAM system in successive randomly selected days. Time step - successive randomly selected testing days, Value - the average value of the price for a given day for real data and data received from the network. Source: Own elaboration in MATLAB and Simulink environment [8].

The training results for the trained networks are presented in Table 1.

Table 1. Learning results of the Deep Network and the Perceptron Network. Source: Own elaboration in MATLAB and Simulink environment [8].

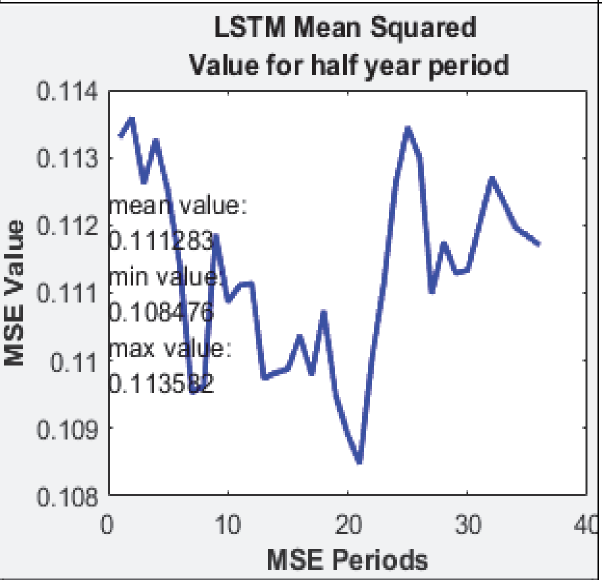
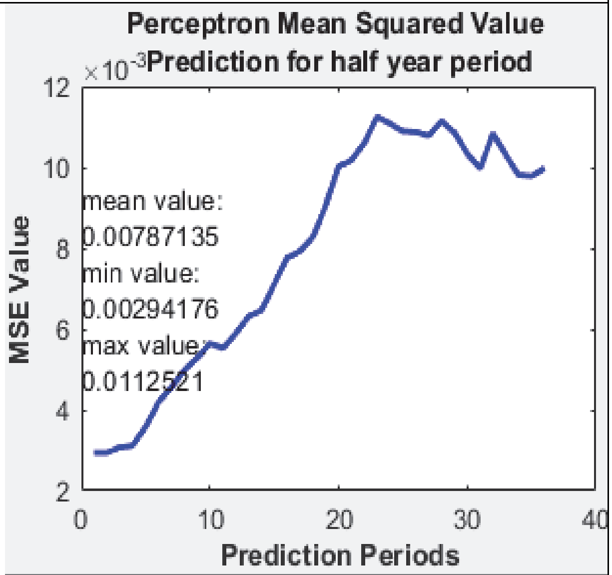
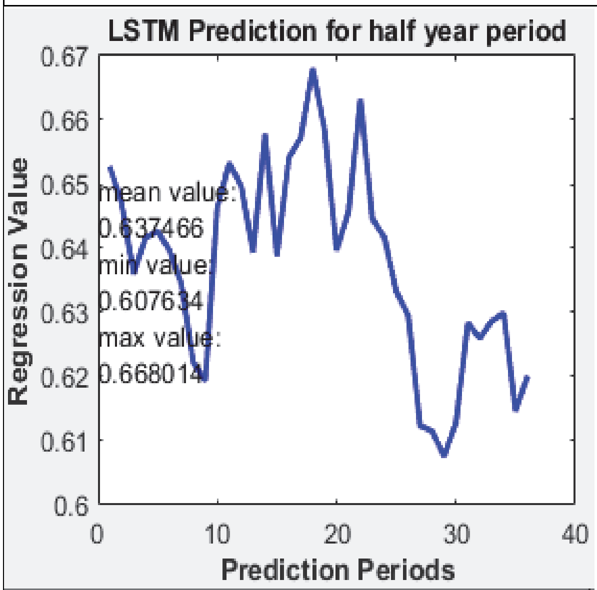
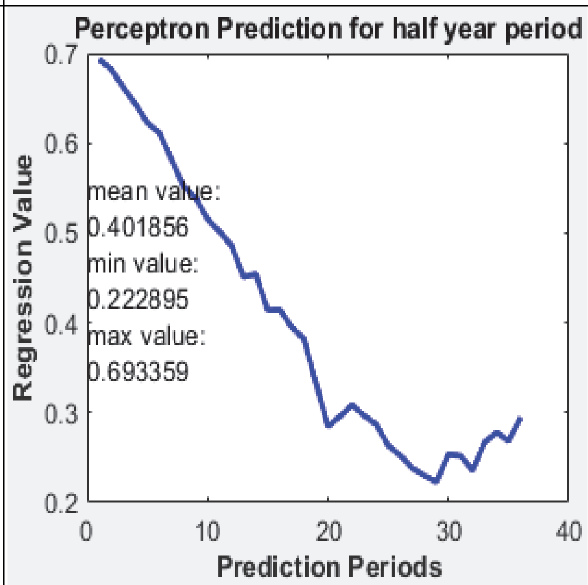
x	Perceptron	Deep Network
Mean Square Error	0.00294176347306251	0.0023072683
Regression	0.693358525142921	0.669756710529327

6. Prediction results

The next stage of the research was to examine the predictive capabilities of the learned models (LSTM and Perceptron). The predictive ability assessment algorithm boils down to extending the prediction period in successive steps by a given value (in this case, 10 days). The assessment based on learned models in which the data was simulated.

The assessment consists in comparing the data being the output of the learned models with real data. MSE and R^2 were adopted as measures of predictive ability. Table No. 2 presents summary graphs of assessment measures for successive prediction periods.

Table 2. The results of the assessment of the prediction capabilities of the Deep Network and the Perceptron network. Source: Own elaboration in MATLAB and Simulink environment [8].

<table>
<tr><th>LSTM</th><th>Perceptron</th></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
</table>

Based on the evaluation of the graphs, it can be concluded that the Perceptron network shows a constant decrease in the quality of prediction with the length of the period for which it was not trained. The deep network based on LSTM does not show such a relationship, which can be seen on the example of the average regression index for the analyzed period, which is 0.223 for the Perceptron network and 0.607 for the LSTM network.

7. Conclusion

Generally, both the Perceptron network and the LSTM-based deep network learned the DAM system at a similar level, the MSE error and the regression index for both networks were similar. The significant difference, however, is at the level of prediction.

The Perceptron Network for the first thirty prediction periods shows the deterioration of the MSE metric and the regression index in an approximately linear fashion, to stabilize at unacceptable values for prediction purposes.

The LSTM network does not show such dependence, the quality of prediction for subsequent periods varies in a certain range, however, it does not show any downward trends, as was the case with the Perceptron network Since both the training data for both networks and the data on the basis of which the networks made predictions were the same, and the degree of learning of models based on both networks (Perceptron and LSTM) was similar, it was hypothesized that the causes of the observed discrepancies have their origins in network architecture.

The factors causing this state of affairs may be the subject of further research.

References

1. Bedi J., Toshniwal D., Deep learning framework to forecast electricity demand, Applied Energy, Vol. 238, pp 1312-1326, 2019.

2. McCulloch W. S., W. Pitts W.,(1943), A logical calculus of the ideas immanent in nervous activity, Bull. Math. Bioph. No 5, pp. 115-133, 1943.

3. Fukushima K, Neocognitron. A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biological Cybernetics. No 36, pp. 193-202, 1980.

4. Hochreiter S. and Schmidhuber J., Long Short-term Memory. Neural computation. No. 9, 1997.

5. Jiang L., Hu G., Day-Ahead Price Forecasting for Electricity Market using Long-Short Term Memory Recurrent Neural Network, 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Singapore, pp. 949-954, 2018.

6. Kingma, D. P. and Ba J., Adam: A Method for Stochastic Optimization. ArXiv. /abs/1412.6980, 2014.

7. Marszałek, A., Burczynski T., Forecasting Day-ahead spot electricity prices using deep neural networks with attention mechanism. J. Smart. Environ. Green. Comput. 1, pp. 21-31, 2021, http://dx.doi.org/10.20517/jsegc.2021.02 (accessed: 20.09.2023].

8. Matlab, Help for MATLAB and Simulink. https://www.mathworks.com /help/deeplearning/ug/list-of-deep-learning-layers.html. 2023, [accessed: 10-15 Aug. 2023].

9. Ruciński D., Modeling of the Day-Ahead Market on the Polish Power Exchange on the example of selected artificial neural networks, [chapter in monograph no 1:] Theory and Application in Artrificial Intelligence Methods, Wydawnictwo Naukowe UPH, pp. 85-117, 2021.

10. Ruciński D., The impact of the size of the training set on the predictive abilities of neural models on the example of the Day-Ahead Market System of TGE S.A. Studia Informatica. Systems and Information Technology, Vol. 1(26), pp. 5-24, 2021.

11. Tchórzewski J., Metody sztucznej inteligencji i informatyki kwantowej w ujęciu teorii sterowania i systemów (in Polish) [eng. Methods of artificial intelligence and quantum computing in terms of control theory and systems]. Wydawnictwo Naukowe. UPH. Siedlce, pages 343, 2021.

12. Zhang R., Li G., and Ma Z, A Deep Learning Based Hybrid Framework for Day-Ahead Electricity Price Forecasting," in IEEE Access, Vol. 8, pp. 143423-143436, 2020.