Medium term electricity price forecasting using extreme learning machine

LADAN PARHIZKARI[1], ARSALAN NAJAFI[1,*], AND MAHDI GOLSHAN[1]

[1] *Faculty of Engineering, Sepidan Branch, Islamic Azad University, Sepidan, Iran*
[*] *Corresponding author: arsalan.najafi@iausepidan.ac.ir*

Manuscript received 9 October, 2019; revised 21 December, 2019, accepted 14 January, 2020. Paper no. JEMT-1910-1203.

Accurate electricity price forecasting gives a capability to make better decisions in the electricity market environment when this market is complicated due to severe fluctuations. The main purpose of a prediction model is to forecast future prices. For doing this, the predicted variable (as output) and historical data (as input) should be close to each other. Machine learning is known as one of the most successful ways of forecasting time series. Extreme learning machine (ELM) is a feed-forward neural network with one hidden layer. Hence, in this paper, an extreme learning machine has been used for predicting electricity prices in a medium-term time horizon. The real data of New York City electricity market has been utilized to simulate and predict the electricity price in four seasons of the year. Finally, the findings are compared with multi-layer perceptron (MLP) results, which prove the efficiency of the model.

keywords: *Electricity price, Forecasting, Time series, Extreme machine learning.*

http://dx.doi.org/10.22109/jemt.2020.204642.1203

1. INTRODUCTION

A. Motivation

In changing from the vertical structure of the power system to the competitive system in a restructured system, the economic transactions are traded based on power signals. These signals are affected by price offering in the electricity market. The majority of these price signals are produced in the day-ahead market. It is very important for market actors and influential entities to understand the severe fluctuations of electricity prices to increase their incomes. Therefore, market participants try to predict prices accurately [1].

B. Literature review

There are various methods to recognize and predict the behavior of a complex system such as time series. In many complex systems, especially nonlinear ones, it is impossible to use classical methods for prediction and control. These methods have features such as; intelligence, knowledge, expertise, and ability to learn and adaptation with the environment [2, 3]. Many methods have been examined in different time frames.

Different researches have been performed in a short-term period. Multiple regression in [4, 5], variable time regression in [6], Box-Jenkis model in [7] and time series CARCH in [8] have been studied. Computational intelligence methods such as a neural network in [9, 10] and fuzzy logic in [11] have been used. In [12], a forecasting strategy has been suggested for real-time electricity market using publicly available market data. This study has used high-resolution data along with hourly data as inputs of two separate forecasting models with different forecast horizons. An intra-hour has also been considered to provide accurate updates on price predictions. Reference [13] has proposed a robust short-term price forecasting in the day-ahead transactions. Accuracy and effectiveness have been improved using a hybrid method for electricity price forecasting via artificial neural network and artificial cooperative search algorithm. An adaptive hybrid model including variational mode decomposition, self-adaptive particle swarm optimization, seasonal auto-regressive integrated moving average and deep belief network has also been presented in [14] for a short time horizon. In addition, a compound method from time series and ANFIS fuzzy logic has been suggested in [15]. In [16], a compound method has predicted the market prices according to the pick prices in a day.

In [17], long-term and medium-term prediction have been done with a resolution of one hour. Reference [18] has determined the influence of out data on price prediction accuracy at a specific time. This article considers a threshold for data and removes out data for a correct prediction. In [19], one month prediction has been done for planning. It is obvious that long time predictions decrease investment costs. A long-term electricity price forecasting has been presented in [20]. The authors have used an auto-regression with exogenous variables and its non-linear counterpart; i.e., an auto-regression with exogenous variables and a neural network have predicted 728 days. A method is presented in [21] using a combination of dual decomposition and multi-objective optimization that is generally constituted from data analysis, optimization, evaluation, and prediction models.

Reference [22] has suggested a hybrid method of autoregressive and Kernel for forecasting the spikes in a real-time market. The method has been implemented in two stages. In the first stage, the prices are forecasted using an autoregressive time varying model, and in the second stage, a kernel regression is utilized to estimate the price spikes.

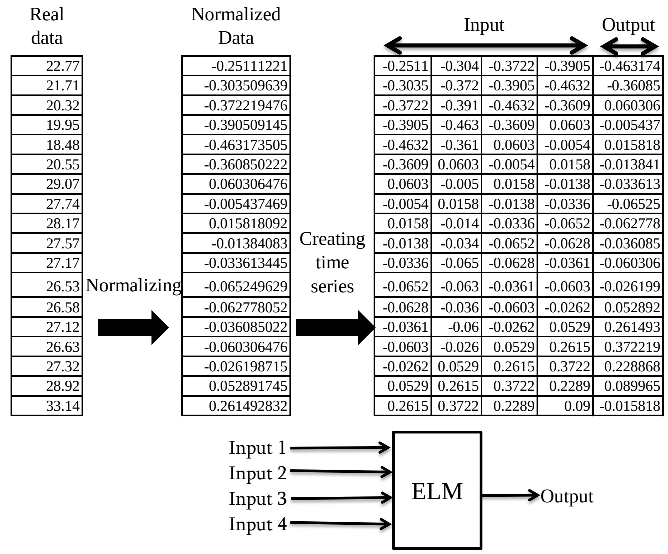
Artificial neural networks are modern systems that are used for computational methods in machine learning and in output predicting of complex systems. The main idea of these systems is inspired by biological neural systems for analyzing data, learning and making knowledge. The key element is to make new structures for analyzing the system's information. This system includes many analytical and continuous elements called neurons. The neurons are coordinated with each other to solve a problem and transfer information with some synapses [3]. The extreme machine learning (ELM) network is a kind of single hidden layer feed-forward neural network that is very fast in learning and generalizing its ability. This method has widely been used in recent years in various research areas. For instance, [23] proposes a hybrid model using extreme learning machines, aiming to forecast wind speed data, and its effectiveness is examined by using real-world data. Reference [24] has studied a carbon price prediction model utilizing a combination of the time series of a complex network and ELM. The ELM has also been used in aerospace science by detecting the fault of the aircraft engine [25]. Moreover, fault diagnosis has been studied in rotating machinery in [26].

C. Paper contribution

This paper investigates the prediction of medium-term electricity price, while many papers have focused on short-term time horizon (e.g., see, Refs. [4–7, 9, 10]), on long-term horizon (e.g., see, Refs. [19, 20]) or on spike prediction (Ref. [22]). Moreover, contrary to popular thinking in which many researchers believe that all the parameters should be trained in feed-forward networks ([9, 10, 13]), this paper randomly generates hidden nodes (input layer weights and biases) and analytically determines the output weights using ELM. In this paper, the ELM is used to predict the electricity market prices. The main contribution of this paper is running and adapting extreme learning machine in the prediction of medium-term horizon of electricity prices. The proposed approach has been implemented in New York City real data, and the results have been compared with the MLP neural network for evaluating the effectiveness of the proposed model.

2. CREATING DATA SAMPLE FROM TIME SERIES

It is necessary to make a specific frame of the forecasting model for analyzing the time series. The output of time series depends on the previous times. Therefore, the specific times of the past should be considered and recorded. For predicting time t, a data-set is created based on times $t-1$ to $t-4$. Fi. 1 shows how the data-set is made. The data are normalized first, and then four inputs are utilized to define the output. At each time t, one sample (including four inputs) is entered the system and one output is generated. The main purpose is to design a model that receives these four inputs and predicts the output. For instance,



Real data
22.77
21.71
20.32
19.95
18.48
20.55
29.07
27.74
28.17
27.57
27.17
26.53
26.58
27.12
26.63
27.32
28.92
33.14

Normalized Data
-0.25111221
-0.303509639
-0.372219476
-0.390509145
-0.463173505
-0.360850222
0.060306476
-0.005437469
0.015818092
-0.01384083
-0.033613445
-0.065249629
-0.062778052
-0.036085022
-0.060306476
-0.026198715
0.052891745
0.261492832

Input				Output
-0.2511	-0.304	-0.3722	-0.3905	-0.463174
-0.3035	-0.372	-0.3905	-0.4632	-0.36085
-0.3722	-0.391	-0.4632	-0.3609	0.060306
-0.3905	-0.463	-0.3609	0.0603	-0.005437
-0.4632	-0.361	0.0603	-0.0054	0.015818
-0.3609	0.0603	-0.0054	0.0158	-0.013841
0.0603	-0.005	0.0158	-0.0138	-0.033613
-0.0054	0.0158	-0.0138	-0.0336	-0.06525
0.0158	-0.014	-0.0336	-0.0652	-0.062778
-0.0138	-0.034	-0.0652	-0.0628	-0.036085
-0.0336	-0.065	-0.0628	-0.0361	-0.060306
-0.0652	-0.063	-0.0361	-0.0603	-0.026199
-0.0628	-0.036	-0.0603	-0.0262	0.052892
-0.0361	-0.06	-0.0262	0.0529	0.261493
-0.0603	-0.026	0.0529	0.2615	0.372219
-0.0262	0.0529	0.2615	0.3722	0.228868
0.0529	0.2615	0.3722	0.2289	0.089965
0.2615	0.3722	0.2289	0.09	-0.015818

Fig. 1. Converting time series to a specific format for implementing in the forecasting process.

in the first row, t_5 is predicted using t_1 to t_4, and in the second row, t_6 is predicted by t_2 to t_5. In a similar way, t_7 is predicted in the third row and this process continues until the last time. The block below the tables in the figure represents the dependency of each output on the four previous times.

Some of the generated samples are considered to train the parameters of the model, and the rest are taken into account for testing purposes. The generated time series are recognized and predicted using ELM. The input data are normalized in the interval [-1,1] in order to have an equal effect of different values. Mean square error (MSE), root mean square error (RMSE) and mean absolute percentage error (MAPE) are used to assess the forecasting process accuracy. Mathematical definitions of MSE and RMSE are given as follows:

$$MSE = \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N} \tag{1}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{N}} \tag{2}$$

$$MAPE = \frac{1}{N}\sum_{1}^{N}|\frac{\hat{y}_i - y_i}{y_i}| \times 100\% \tag{3}$$

where y_i is the electricity price at time i, $\hat{y}_i$ is the predicted price at time i and N is the number of samples.

3. MULTI LAYER PERCEPTRON

For introducing the MLP, the mathematical model of a biological neuron is depicted in Fig. 2. This model is constituted of inputs, weights, biases, and a function that yields the output. This figure shows a three-layer network and can be extended to a multi-layer structure to make the MLP network as described in Fig. 3. The MLP is a neural network that includes at least three layers. The basic form of the MLP consists of an output and a hidden layer [27] and [28].

By involving kth pattern to the network in an incremental learning manner, with considering n_0 sample and j neuron in

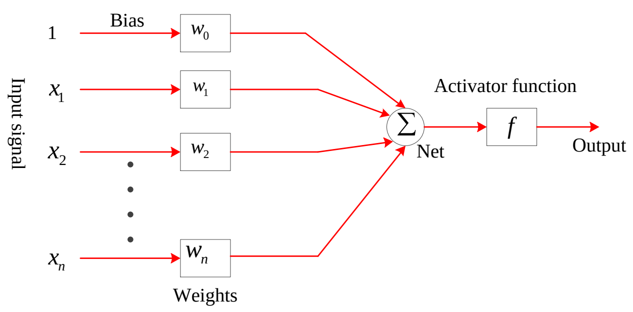


Fig. 2. Mathematical model of a biological neuron and its components.

the first hidden layer, the network *net* in layer one is calculated as follows [27]:

$$net_j^1(k) = \sum_{i=0}^{n_0} w_{ij}^1(k) x_i(k) \tag{4}$$

where $w_{ij}1(k)$ is the weight of input i and neuron j of the first layer, and $x_i(k)$ is the input i of pattern k. The output of layer one is recalculated by implementing an activator function as follows:

$$O_j^1(k) = f_j^1(net_j^1)(k) \tag{5}$$

where $O_j^1(k)$ is the output of neuron j in the first layer; $f(.)$ is the transfer function, which changes the environment of inputs into either linear or non-linear environment. This transfer results in a better decoupling of input data and making more conceptual data. There are various activator functions including threshold, hard limit function, piece-wise linear function, sigmoid function, linear function, tangent sigmoid, etc.

By comparing the actual target (T) and the forecasted output by MLP (O^2) in stage k, the error function $e_j(k)$ is defined as follows:

$$e_j(k) = (T_j(k) - O_j^2(k)) \tag{6}$$

In this equation, there are m error functions (equal to number of outputs). Taking into account the mean value, the error function is represented as follows:

$$E(k) = \frac{1}{m}\sum_{j=1}^{m}(e_j(k))^2 \tag{7}$$

where $E(k)$ is the mean value of the error. The algorithm of training middle layers using gradient decent with partial derivatives is given as follows:

$$w^2(k+1) = w^2(k) - \eta_{w^2} \cdot \frac{\partial E}{\partial w^2} \tag{8}$$

$$\frac{\partial E}{\partial w^2} = \frac{\partial E}{\partial e} \times \frac{\partial e}{\partial o^2} \times \frac{\partial o^2}{\partial net^2} \times \frac{\partial net^2}{\partial w^2} = e(k)(-1)(f^2(net^2(k)))(O^1(k)) \tag{9}$$

Therefore, the weights w^1 and w^2 are also updated as follows:

$$w^2(k+1) = w^2(k) + \eta_{w^2} \cdot e(k) \cdot f'^{(2)}(net^2(k)) \cdot O^1(k) \tag{10}$$

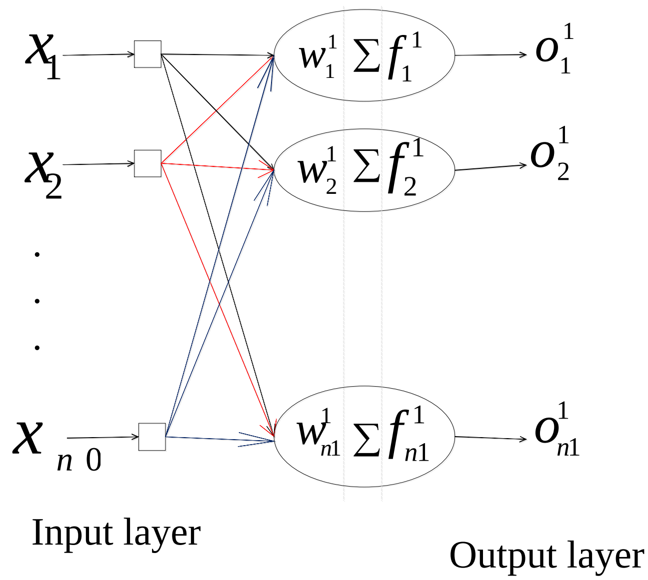


Fig. 3. MLP network [27].

$$w^1(k+1) = w^1(k) + \eta_{w^1} \cdot e(k) \cdot f'^{(1)}(net^1(k)) \cdot x_i(k) \tag{11}$$

where η_{w^1} and η_{w^1} are the learning rates.

The main steps of the MLP are summarized as follows:

Step 1: determining input data.
Step 2: normalizing input data.
Step 3: creating a time series.
Step 4: determining input and target for train and test data.
Step 5: calculating the output of the network 1.
Step 6: calculating the output of network 2.
Step 7: calculating error.
Step 8: updating weights (w^1 and w^2) based on the learning rate and the error in each epoch.
Step 9: repeating step 5 to 8 until reaching the number of epochs.

4. EXTREME LEARNING MACHINE

Feed-forward neural networks have been used in many fields due to their properties. A single hidden layer feed-forward neural network (SLFN) with at most N hidden neurons and almost any nonlinear activation function can learn N distinct observations at function approximation problems, where N is the number of samples per data-set. All the parameters of the SLNF need to be tuned in almost all traditional learning algorithms. As the number of hidden layers increases, the number of training parameters and their training time increase. Initial parameters may also not be well trained. However, in many gradient-based learning algorithms, the problem is being many iterative learning steps, converging to a local minimum and slow learning process. Therefore, there is a need to provide a fast learning algorithm in SLNFs. Contrary to popular thinking in which many researchers believe that all the parameters should be trained in the feed-forward networks, one may generate input weights and first hidden layer biases randomly. Therefore, an extreme learning machine has been proposed to solve the mentioned issues [29]. Extreme learning machine algorithms can be easily implemented. These algorithms also tend to reach

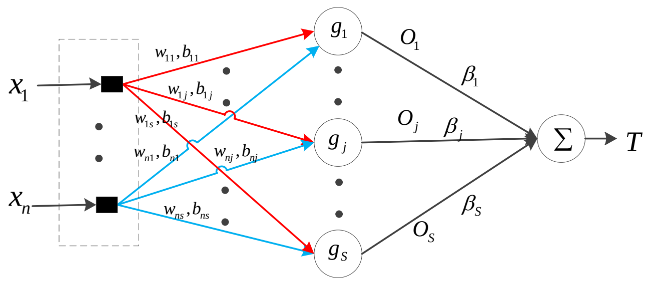


Fig. 4. ELM network [29].

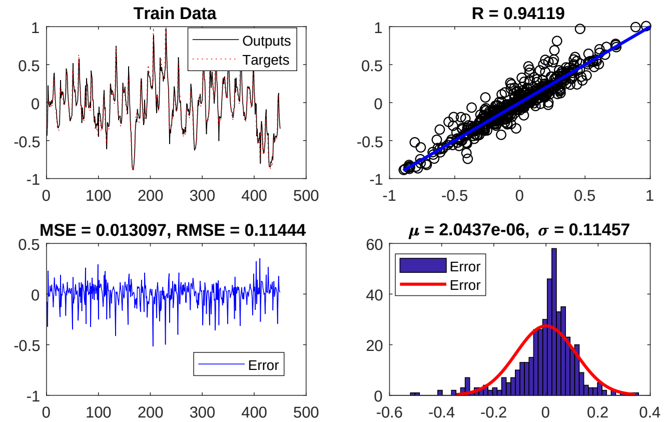


Fig. 5. Results of training of ELM in February.

the smallest training error, obtain the smallest norm of weights and run extremely fast, compared to the other popular SLFN learning algorithms. Extreme learning machine has resolved the problems of descending gradient training based feed-forward neural networks. Extreme learning machine shows that the hidden layer can be produced randomly. In order to better understanding, consider a single layer MLP network in which the first layer learns. In fact, the ELM is like the MLP network but its layer is produced randomly, and only the second layer (output layer) is trained. Assume that the input layer with n dimension is related to the hidden layer of ELM with dimension S as shown in Fig. 4, so the output of the network takes the following form:

$$f_N(x) = \sum_{i=1} N\beta_i h_i(x) = h(x)\beta \quad \textbf{(12)}$$

where $\beta = [\beta_1, \beta_2, ..., \beta_N]^T$ is the weight of the output layers that connects the hidden layer to the output layer. $h(x) = [g_1(x), g_2(x), ..., g_N(x)]$ is the output of the hidden layer nodes. For n training samples, the following equation is defined for the ELM:

$$H\beta = T \quad \textbf{(13)}$$

where $T = [t_1, ..., t_S]^T$ is the relative labels of target and $H = [h^T(x_1), ..., h^T(x_S)]^T$ is the matrix of the hidden layer. We can compute beta weight as follows:

$$\beta = H^{\dagger}T \quad \textbf{(14)}$$

where $H^{\dagger}$ is the Moore-Penrose matrix. Fig. 4 shows ELM structure with related weights. It should be noted that in addition to the difference in learning, the other difference between ELM and MLP methods is the utilization of the least squares error instead of gradient descent to decrease the error in network training. The training of algorithm in ELM is only done in neuron weights of the output or the second layer. The β parameter in ELM is equivalent to w^2 in MLP. In contrast to the MLP, the first layer weights w^1 is obtained randomly. The weights of the output/hidden layer in ELM are calculated using the least mean square (LMS).

The steps of implementing ELM are similar to the MLP which was mentioned in Section 3. There are just two differences such that, the hidden layer output matrix H and β are calculated instead of w^2 and w^1 is generated randomly.

5. SIMULATION RESULTS

The real data of electricity market of New York City in 2017 have been simulated to evaluate the efficiency of the proposed model [30]. The data of one month of each season has been selected to show the ability of the model in dealing with different patterns. The selected months are February, May, August, and November. The model is made using four previous data to predict the current time as mentioned in Section 2. A total of 70% of the data are used for training purposes and the rest for testing. The number of hidden neurons is considered to be 15, and tangent sigmoid function is used for the transfer function. The results related to training and test of February prices are depicted in Figs.5 and 6. The figures include the output and target, regression, error value and normal distribution function of error value. The value of regression of training and test in February are, respectively, 0.942 and 0.966, demonstrating the high accuracy of the ELM. Moreover, the value of the error is very small and the corresponding MSE verifies this issue. In addition, it can be seen that the target and output are very close due to the well tracing of the target by output. Similarly, the values of regressions in other months are greater than 0.93 in both training and testing, which shows the high capability of the proposed model (see Figs. 7 - 12). Comparing these figures shows close values for training regression. The regression values are, respectively, 0.941, 0.938, 0.983 and 0.961, meaning the best performance in August and the worst in May. Indeed, the difference in regression values refers to the complexity of the prices pattern. As a result, the minimum value of the mean error occurs in August (0.0697) and the maximum error in May (0.123). Regarding the test results, the regression values for February, May, August and November are, respectively, 0.965, 0.954, 0.979 and 0.962, showing the acceptable performance of the ELM in all months. High accuracy of the ELM leads to low values of the mean errors, which are, respectively, 0.087, 0.123, 0.0793 and 0.0957. Additionally, the targets and outputs are very close leading to small values of an errors in the figures.

It is obvious that, the efficiency of each model can be analyzed by comparing to other well-known models. For this purpose, the forecasting process has been conducted by a multi-layer perceptron neural network with the data given in Table 1. The input data are similarly normalized in the interval [-1,1], and the tangent sigmoid has been taken into account as the activator function. There are 168 test data (one week) and the rest are the training data.

The results of forecasting prices in the aforementioned months are obtained and the test results are shown in Figs. 13 - 16 and Table 2. One week (168 hours and seven days) is depicted for each month as the test data. For instance, there are six similar days in February and one different day. All days have been

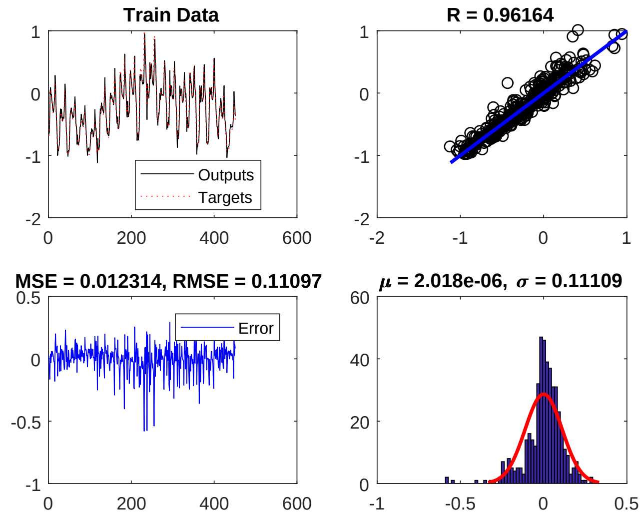


Fig. 11. Results of training of ELM in November.

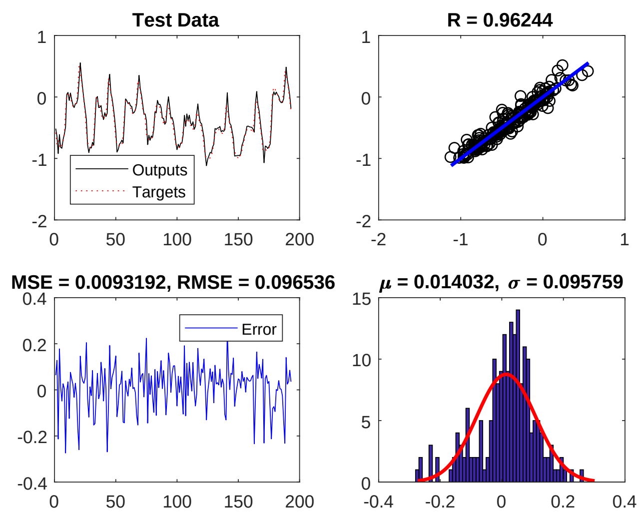


Fig. 12. Results of testing of ELM in November.

similar to semi-spike points. There is less difference between the peak and off-peak hours in November as the cold month of the year. In most days, there is only one peak per day, which makes the prediction simple although this is a different pattern comparing to other months. However, the ELM can predict a different pattern properly. The superiority of the ELM over MLP can be seen in the figure, again.

The quantitative comparison of ELM and MLP has been provided in Table 2. The RMSE and MAPE of the ELM in all months are lower than those of the MLP. For instance, the RMSE of training for MLP and ELM are, respectively, 0.1244 and 0.1141, and these values are 0.1041 and 0.0887 for the test data. However, the main superiority of ELM is its computational time showing its high speed.

Here, for showing the dependency of the ELM training on the previous samples, which were mentioned in Section 2, a sensitivity analysis is conducted. The results of RMSE with respect to the four states of $t-1$ to $t-6$ are given in Table 3. The best results have been obtained when the four previous samples

Table 1. Characteristics of MLP

number of inputs	number of outputs	neuron of hidden layers	η_w
2	4	1	0.05
epoch	normalization interval	Train percent	training algorithm
100	[-1,1]	0.75	Decent gradient

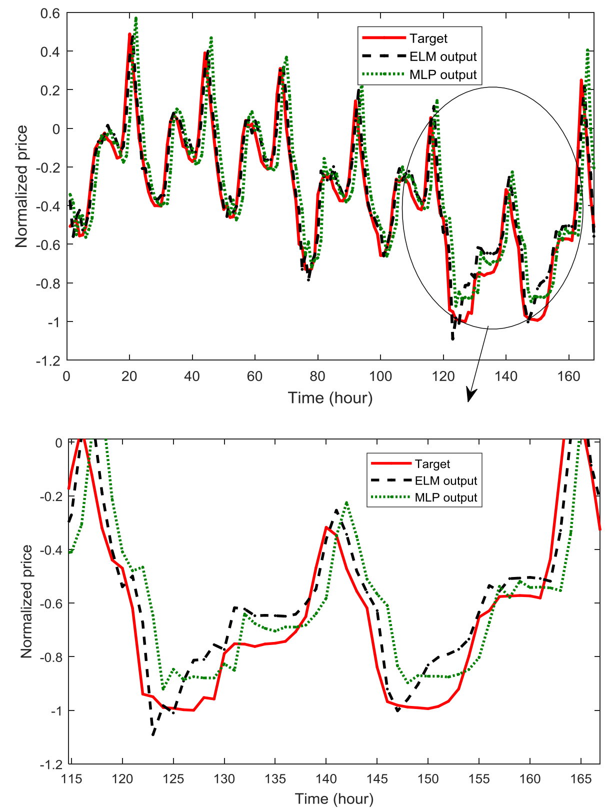


Fig. 13. Comparing the target and output of ELM and MLP in February.

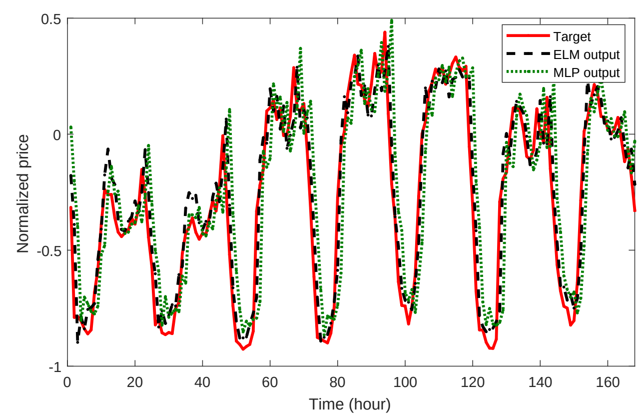


Fig. 14. Comparing the target and output of ELM and MLP in May.

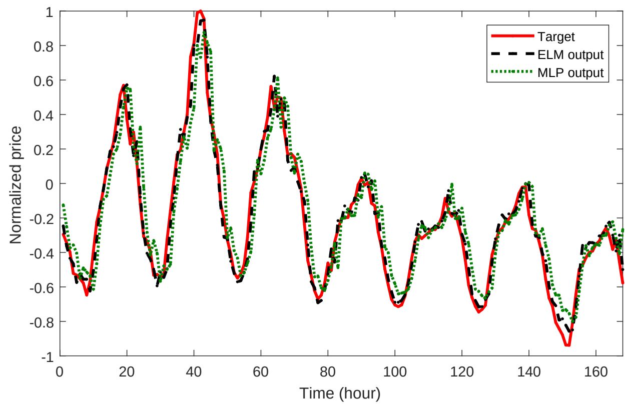


Fig. 15. Comparing the target and output of ELM and MLP in August.

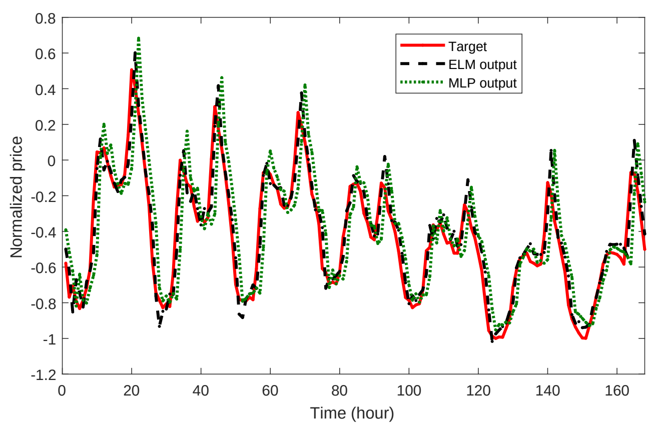


Fig. 16. Comparing the target and output of ELM and MLP in November.

$(t-4)$ are taken into account, while the results of $t-2$ to $t-6$ are very close showing a saturation in error reduction. On the contrary, the worst results are obtained when only one of the previous samples is considered. To select the best number of samples, the try and error method should be done. It is because the nature of the input data are different in each time series. Here, the best is with $t-4$ while in another problem the best result may be with lower or more than four previous samples.

6. CONCLUSION

In this paper, an extreme learning machine was used for medium-term electricity price forecasting. The electricity market price data of the New York City market was used as the data-set. These data were utilized in both training and testing. The data, at first, were converted into a specific time series in which four previous hours were taken into account to predict the current time. In order to show the capability of the proposed model, a comparison was made between the proposed model and the MLP neural network. The lower RMSE in both training and testing demonstrates the superiority of the ELM. Moreover, dealing with various patterns in different months verifies the efficiency of the ELM in facing with every time electricity prices. In addition, the short computational time shows the high speed of the ELM.

Table 2. Comparing the results of ELM and MLP

		MLP	ELM
February	RMSE train	0.1244	0.1141
	RMSE test	0.1041	0.0887
	MAPE train	94.2	89.73
	MAPE test	86.03	84.35
	Time (s)	57.61	0.189
May	RMSE train	0.1580	0.1501
	RMSE test	0.1329	0.1312
	MAPE train	299.8	276.4
	MAPE test	178.2	144
	Time (s)	59.81	0.196
August	RMSE train	0.0815	0.0652
	RMSE test	0.0897	0.0737
	MAPE train	70.01	56.39
	MAPE test	45.18	40.78
	Time (s)	59.54	0.181
November	RMSE train	0.1209	0.1123
	RMSE test	0.1041	0.0955
	MAPE train	84.7	76.97
	MAPE test	106	72.28
	Time (s)	58.2	0.207

Table 3. Comparing the results of ELM and MLP

	No. samples						
	Action	1	2	3	4	5	6
February	train	0.138	0.1165	0.1158	0.1141	0.1149	0.1175
	test	0.1528	0.1101	0.0994	0.0887	0.0942	0.0977
May	train	0.1699	0.1568	0.1538	0.1501	0.1505	0.1501
	test	0.1458	0.1331	0.1339	0.1312	0.1294	0.1344
August	Train	0.0.1027	0.0731	0.0718	0.0652	0.0688	0.0689
	Test	0.1040	0.0819	0.08	0.0737	0.0781	0.0778
November	Train	0.1420	0.1178	0.1176	0.1123	0.1153	0.1133
	Test	0.1318	0.1047	0.1092	0.0955	0.1002	0.0986

REFERENCES

1. C. Monteiro, I.J. Ramirez-Rosado, L. Alfredo Fernandez-Jimenez, and M. Ribeiro, “New probabilistic price forecasting models: Application to the Iberian electricity market,” International Journal of Electrical Power & Energy Systems, vol. 103, pp. 483–496, 2018.
2. T.M. Mitchell, “Machine learning,” Burr Ridge, IL: McGraw Hill, vol. 45, p. 37, 1997.
3. S. Haykin, “Neural networks, a comprehensive foundation,” Prentice Hall; 2 edition, vol. 91, p. 103176, 1994.
4. A.J. Conejo, J. Contreras, R. Espínola, and M.A. Plazas, “Forecasting electricity prices for a day-ahead pool-based electric energy market,” International Journal of Forecasting, vol. 21, pp. 435-462, 2005.
5. S. Bordignon, D.W. Bunn, F. Lisi, and F. Nan, “Combining day-ahead forecasts for British electricity prices,” Energy Economics, vol. 35, pp.

88-103, 2013.
6. T. Jonsson, P. Pinson, H.A. Nielsen, H. Madsen, and T.S. Nielsen, “Forecasting electricity spot prices accounting for wind power predictions,” IEEE Transactions on Sustainable Energy, vol. 4, pp. 210-218, 2013.
7. Y. Dong, J. Wang, H. Jiang, and J. Wu, “Short-term electricity price forecast based on the improved hybrid model,” Energy Conversion and Management, vol. 52, pp. 2987-2995, 2011.
8. Z. Tan, J. Zhang, J. Wang, and J. Xu, “Day-ahead electricity price forecasting using wavelet transform combined with ARIMA and GARCH models,” Applied Energy, vol. 87, pp. 3606-3610, 2010.
9. H.Y. Yamin, S.M. Shahidehpour, and Z. Li, “Adaptive short-term electricity price forecasting using artificial neural networks in the restructured power markets,” International Journal of Electrical Power & Energy Systems, vol. 26, pp. 571-581, 2004.
10. W.M. Lin, H.J. Gow, and M.T. Tsai, “Electricity price forecasting using enhanced probability neural network,” Energy Conversion and Management, vol. 51, pp. 2707-2714, 2010.
11. N. Amjady, “Day-ahead price forecasting of electricity markets by a new fuzzy neural network,” IEEE Transactions on Power Systems, vol. 21, pp. 887-896, 2006.
12. H. Chitsaz, P. Zamani-Dehkordi, H. Zareipour, and P.P. Parikh, “Electricity price forecasting for operational scheduling of behind-the-meter storage systems,” IEEE Transactions on Smart Grid, vol. 9, pp. 6612-6622, 2018.
13. A. Pourdaryaei, H. Mokhlis, H.A. Illias, S.H.A. Kaboli, S. Ahmad, and S.P. Ang, “Hybrid ANN and artificial cooperative search algorithm to forecast short-term electricity price in de-regulated electricity market,” IEEE Access, vol. 7, pp. 125369-125386, 2019.
14. J. Zhang, Z. Tan, and Y.Wei, “An adaptive hybrid model for short term electricity price forecasting,” Applied Energy, vol. 258, p. 114087, 2020.
15. J.L. Zhang, Y.J. Zhang, D.Z. Li, Z.F. Tan, and J.F. Ji, “Forecasting day-ahead electricity prices using a new integrated model,” International Journal of Electrical Power & Energy Systems, vol. 105, pp. 541-548, 2019.
16. A. Mirakyan, M. Meyer-Renschhausen, and A. Koch, “Composite forecasting approach, application for next-day electricity price forecasting,” Energy Economics, vol. 66, pp. 228-237, 2017.
17. F. Ziel and R. Steinert, “Probabilistic mid- and long-term electricity price forecasting,” Renewable and Sustainable Energy Reviews, vol. 94, pp. 251-266, 2018.
18. D.O. Afanasyev and E.A. Fedorova, “On the impact of outlier filtering on the electricity price forecasting accuracy,” Applied Energy, vol. 236, pp. 196-210, 2019.
19. T. Windler, J. Busse, and J. Rieck, “One month-ahead electricity price forecasting in the context of production planning,” Journal of Cleaner Production, vol. 238, p. 117910, 2019.
20. K. Hubicka, G. Marcjasz, and R. Weron, “A note on averaging day-ahead electricity price forecasts across calibration windows,” IEEE Transactions on Sustainable Energy, vol. 10, pp. 321-323, 2019.
21. W. Yang, J. Wang, T. Niu, and P. Du, “A hybrid forecasting system based on a dual decomposition strategy and multi-objective optimization for electricity price forecasting,” Applied Energy, vol. 235, pp. 1205-1225, 2019.
22. D.H. Vu, K.M. Muttaqi, A.P. Agalgaonkar, and A. Bouzerdoum, “Short-term forecasting of electricity spot prices containing random spikes using a time-varying autoregressive model combined with Kernel regression,” IEEE Transactions on Industrial Informatics, vol. 15,
23. H.F. Yang and Y.P.P. Chen, “Representation learning with extreme learning machines and empirical mode decomposition for wind speed forecasting methods,” Artificial Intelligence, vol. 91, p. 103176, 2019.
24. H. Xu, M. Wang, S. Jiang, and W. Yang, “Carbon price forecasting with complex network and extreme learning machine,” Physica A: Statistical Mechanics and its Applications, vol. 91, p. 122830, 2019.
25. Y.P. Zhao, G. Huang, Q.K. Hu, J.F. Tan, J.J. Wang, and Z. Yang, “Soft extreme learning machine for fault detection of aircraft engine,” Aerospace Science and Technology, vol. 91, pp. 70-81,
26. X.B. Wang, X. Zhang, Z. Li, and J. Wu, “Ensemble extreme learning machines for compound-fault diagnosis of rotating machinery,” Knowledge-Based Systems, vol. 9, p. 105012, 2019.
27. F. Zhang, K. Sun, and X. Wu, “A novel variable selection algorithm for multi-layer perceptron with elastic net,” Neurocomputing, vol. 361, pp. 110-118, 2019.
28. K. Bhattacharjee and M. Pant, “Hybrid particle swarm optimization-genetic algorithm trained multi-layer perceptron for classification of human glioma from molecular brain neoplasia data,” Cognitive Systems Research, vol. 58, pp. 173-194, 2019.
29. G.B. Huang, Q.Y. Zhu, and C.K. Siew, “Extreme learning machine: Theory and applications,” Neurocomputing, vol. 70, pp. 489-501, 2006.
30. NY ISO, [Online], Available at: "http://www.nyiso.com.