TIME SERIES PREDICTION WITH MACHINE LEARNING TECHNIQUES. PART II – RECURRENT NEURAL NETWORKS MODELS APPLIED TO FORECASTING ROMANIA'S ENERGY CONSUMPTION

BOGDAN DIACONU, *University "Constantin Brâncuși" of Tg-Jiu*
LUCICA ANGHELESCU, *University "Constantin Brâncuși" of Tg-Jiu*
MIHAI CRUCERU, *University "Constantin Brâncuși" of Tg-Jiu*
CRISTINEL RACOCEANU, *University "Constantin Brâncuși" of Tg-Jiu*
ADRIANA TUDORACHE, *University "Constantin Brâncuși" of Tg-Jiu*
MARIUS EREMIA VLAICU POPA, *Complexul Energetic Oltenia*

ABSTRACT: Forecasting the electricity consumption is a critical activity for planning and management of the national grid. Various approaches exist to forecast the energy consumption. In this paper, a machine learning approach was employed based on the publicly available energy consumption dataset from Transelectrica S.A. – Romania's national grid operator. The dataset presented as a time series of momentary consumption data (power) was processed and transformed into a time series of daily energy consumption (energy). Then, several machine learning models based on artificial neural networks were developed and their performance was assessed. It was found that LSTM and fully connected models have similar performance and can be used as a backbone of more complex models which consider multivariate time series.

Keywords: - **Keyords:** - *Time series; Machine learning; Recurrent Neural Networks; Forecasting*

1.INTRODUCTION

Accurate electricity consumption forecast is essential for long-term planning of electricity production facilities. Both overestimation and underestimation have serious negative economic consequences impacting significantly the economic growth at national level.

Traditional methods include time series models, regression models, Box–Jenkins models, econometric models, neural networks, ant colony optimization, genetic algorithms and statistical learning models. More recent studies report good performance of methods such as Least-Squares Support Vector Machines [1], Particle Swarm Optimization [2] genetic algorithms [3].

Electricity consumption is a multivariate type of time series with complex influences from many variables, such as GDP, macroeconomic conditions and many more others.

The time scale of prediction is a factor that determines the choice of the forecasting algorithm as well as the object of prediction. The selection of the features for the model of choice also depends on the time scale and object of prediction. Hwang et al [4] presented a review of studies with different time scales and object of prediction. Hwang et al [4] conducted a study attempting to predict the electricity consumption for commercial buildings. A number of five different models were tested and two time scales – monthly and daily were considered. In this paper, several artificial neural networks models for time series modelling will be considered and a critical comparison between different types will be carried out.

2. DATA SOURCE AND PRELIMINARY PROCESSING

The data used in this study has been collected from the Romania national grid operator, TRANSELECTRICA S.A. (transelectrica.ro). The database available at https://www.transelectrica.ro/ro/web/tel/home includes both production in national electricity production facilities, the consumption and the imports. A detailed breakdown of the production on source types is available, as shown in Figure 1.

Data	Consum [MW]	Medie Orara Consum [MW]	Productie [MW]	Carbune [MW]	Hidrocarburi [MW]	Ape [MW]	Nuclear [MW]	Eolian [MW]	Foto [MW]	Biomasa [MW]	Sold [MW]
30-10-2021 15:33:59	6423	6475	5887	1324	1462	731	1420	496	390	64	536
30-10-2021 15:24:09	6396	6475	5878	1329	1464	709	1414	487	412	62	518
30-10-2021 15:14:19	6447	6475	5902	1330	1464	720	1418	493	413	65	546
30-10-2021 15:04:29	6411	6475	5934	1327	1464	721	1420	487	449	66	477
30-10-2021 14:54:39	6363	6475	5948	1324	1463	709	1416	512	459	65	415
30-10-2021 14:44:49	6427	6475	5952	1323	1462	713	1419	502	470	63	475
30-10-2021 14:34:59	6418	6475	6009	1334	1462	746	1420	511	473	63	409
30-10-2021 14:25:09	6526	6475	6001	1329	1458	753	1421	509	469	62	525
30-10-2021 14:15:19	6465	6475	5978	1333	1465	714	1419	501	483	62	488
30-10-2021 14:05:29	6501	6475	6007	1331	1462	740	1420	495	495	64	494
30-10-2021 13:55:39	6592	6665	6079	1341	1464	788	1415	505	501	64	514
30-10-2021 13:45:49	6548	6665	6105	1353	1470	788	1416	505	509	63	443
30-10-2021 13:35:59	6699	6665	6141	1358	1469	814	1417	503	516	63	558

Figure 1. The production/consumption database maintained by TRANSELECTRICA (excerpt)

The momentary values of the production/consumption are given in MW every 9 minutes and 50 seconds (approximately). Values as old as 2007 are available.

The approach used in this paper was to determine the daily energy consumption by integrating the momentary values over time:

$$E_{day} = \int_{0}^{24} P_{cons} \cdot d\tau$$

The TRANSELECTRICA database allows custom selection of the time interval, as shown in Figure 2.

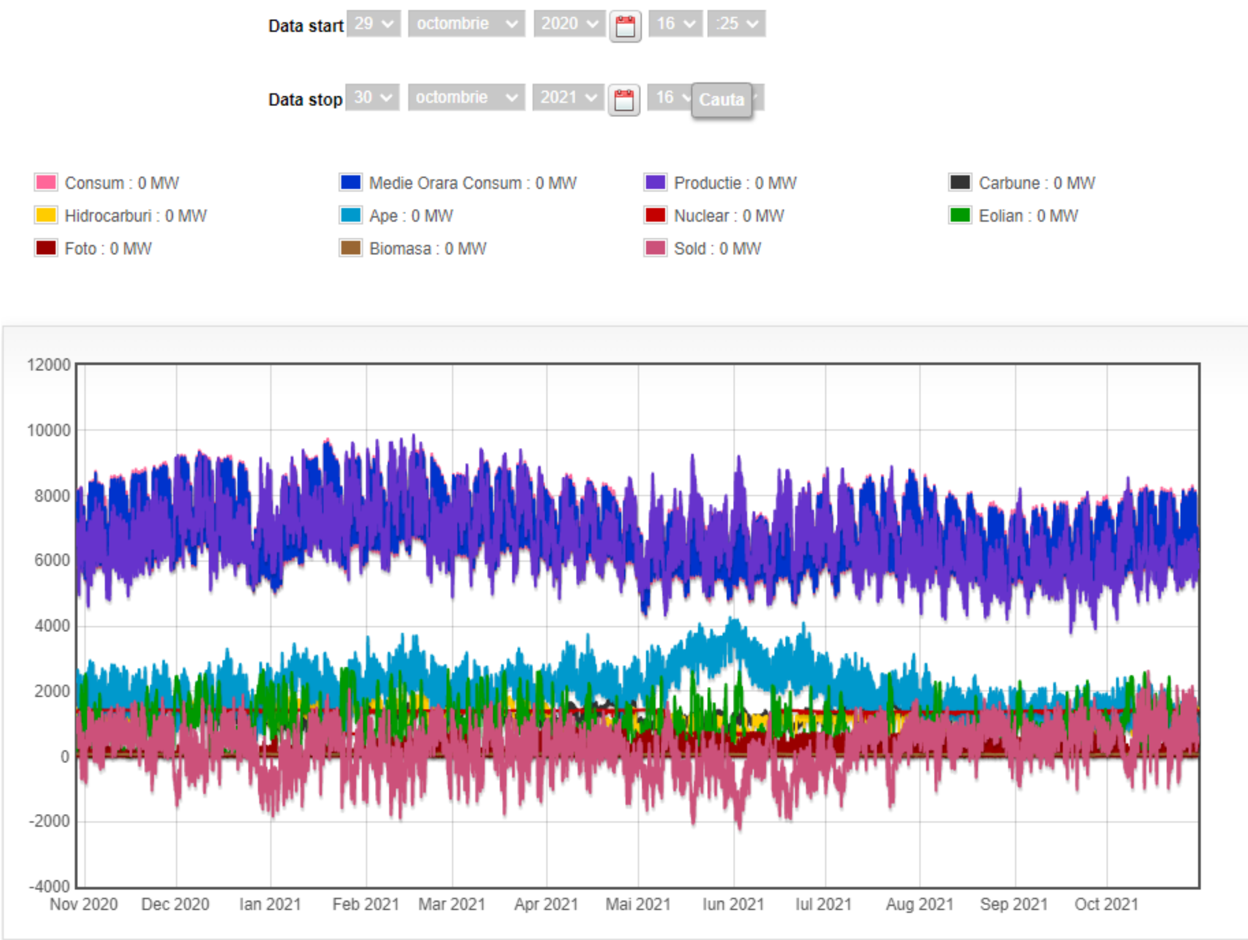

Figure 2. Selection of the interval and display of the data (screen capture from transelectrica.ro)

In this paper, the analysis interval starts on October 22nd, 2013 at 00:28:58 and ends on October 23rd, 2021 at 00:20:19. The widget generated a database selection containing 429239 entries. The Excel file was saved locally and converted to csv format.

3. PREPROCESSING

Both preprocessing and actual processing of the data (raw data generated by the transelectrica.ro widget was in the form presented in Figure 3) was carried out using the Google library TensorFlow. TensorFlow is an open-source, freely available library for artificial intelligence and machine learning written in Python. The most recent stable release is 2.6.0 (released on August 11th, 2021). The data analysis and manipulation tool Pandas was used to convert the first column of the database (see Figure 1) into a datetime object in order to allow operations like subtraction and conversion of time intervals expressed in dd:mm:yyyy hh:MM:SS into seconds.

	A	B	C	D	E	F	G	H	I	J	K	L
1	Data	Consum[MW]	Medie Consum[MW]	Productie[MW]	Carbune[MW]	Hidrocarburi[MW]	Ape[MW]	Nuclear[MW]	Eolian[MW]	Foto[MW]	Biomasa[MW]	Sold[MW]
2	23-10-2021 00:20:19	6058	6338	5817	1386	1369	952	1414	619	0	77	241
3	23-10-2021 00:10:29	6077	6338	5851	1379	1366	949	1416	663	0	79	226
4	23-10-2021 00:00:39	6164	6338	5737	1347	1273	921	1419	700	0	78	427
5	22-10-2021 23:50:49	6155	6338*	5643	1352	1224	855	1415	717	0	80	512
6	22-10-2021 23:40:00	6236	6338*	5714	1354	1229	871	1416	761	0	82	523
7	22-10-2021 23:30:10	6354	6338*	5749	1367	1232	870	1419	783	0	79	605
8	22-10-2021 23:20:20	6350	6338*	5775	1375	1238	875	1417	792	0	78	575
9	22-10-2021 23:10:30	6444	6338*	5789	1364	1249	888	1413	796	0	79	654

Figure 3. The raw database as generated by the transelectrica.ro widget

Then, integration of the momentary power values was carried out over the discrete values of the time corresponding to one calendar day. In average, the number of momentary values of the consumption in one day was 148. The numerical integration was carried out employing the trapezoidal rule.

	A	B	C	D	E	F
1	Index	No	Date	Day index	Daily energy consumption	Weekday
2	0	1	10/22/2013	22	6429.953351	1
3	1	2	10/23/2013	23	6448.335498	2
4	2	3	10/24/2013	24	6468.704745	3
5	3	4	10/25/2013	25	6435.579606	4
6	4	5	10/26/2013	26	5735.72728	5
7	5	6	10/27/2013	27	5347.983391	6
8	6	7	10/28/2013	28	6313.414161	0
9	7	8	10/29/2013	29	6410.366453	1
10	8	9	10/30/2013	30	6415.854207	2
11	9	10	10/31/2013	31	6465.536157	3
12	10	11	11/1/2013	1	6448.919497	4

Figure 4. The processed data: daily energy consumption

Finally, the daily energy consumption for each day of the analysis interval was determined and a new database was generated as shown in Figure 4.

In graphical form, the data is represented in Figure 5.

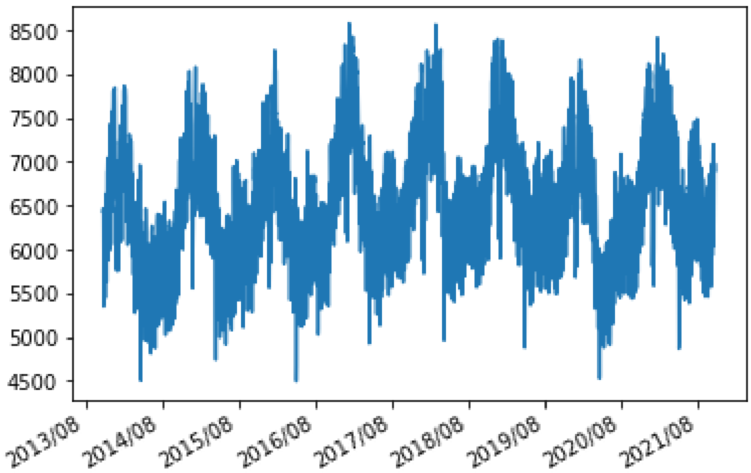

Figure 5. Daily energy consumption from 10/22/2013 till 10/23/2021

4. RNN MODELS APPLICATION TO FORECASTING DAILY ENERGY CONSUMPTION

In this paper, the daily energy consumption will be considered as a univariate time series and several artificial neural networks models will be tested.

The performance metrics used to assess the performance of different models were the following:

- Mean Absolute Error:

$$mae = \frac{1}{N}\sum_{i=1}^{N}\left|y_{pred,i} - y_{true,i}\right|$$

- Mean Squared Error:

$$mse = \frac{1}{N}\sum_{i=1}^{N}\left(y_{pred,i} - y_{true,i}\right)^2$$

- Mean Absolute Scaled Error (mean absolute error of the forecast values, divided by the mean absolute error of the in-sample one-step naive forecast):

$$mase = \frac{\frac{1}{N}\sum_{i=1}^{N}\left|y_{pred,i} - y_{true,i}\right|}{\frac{1}{N-1}\sum_{i=2}^{N}\left(y_{true,i} - y_{true,i-1}\right)}$$

- Mean Absolute Percentage Error:

$$mape = 100 \cdot \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_{pred,i} - y_{true,i}}{y_{true,i}}\right|$$

- Root Mean Squared Error:

$$rmse = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_{pred,i} - y_{true,i}\right)^2}$$

The forecasting strategy will employ a horizon value always equal to 1 day; the window size will be varied from 7 to 30 days to assess its effect on the model performance. The dataset was split into training and testing subsets, with the first 80% (in chronological order) values allocated to the training set and the remaining 20% values to the test set.

In order to generate the data, set a windowing algorithm was employed. The algorithm generates sets of pairs features – labels. The features consist of the energy consumption values for the first *w* days and the corresponding label is the next *h* days, where *w* is the window size and *h* is the horizon size. The operation of the algorithm is represented in a graphic form in Figure 6.

6429.953	6429.953	6429.953	6429.953	6429.953	6429.953	6429.953	6429.953
6448.335	6448.335	6448.335	6448.335	6448.335	6448.335	6448.335	6448.335
6468.705	6468.705	6468.705	6468.705	6468.705	6468.705	6468.705	6468.705
6435.58	6435.58	6435.58	6435.58	6435.58	6435.58	6435.58	6435.58
5735.727	5735.727	5735.727	5735.727	5735.727	5735.727	5735.727	5735.727
5347.983	5347.983	5347.983	5347.983	5347.983	5347.983	5347.983	5347.983
6313.414	6313.414	6313.414	6313.414	6313.414	6313.414	6313.414	6313.414
6410.366	6410.366	6410.366	6410.366	6410.366	6410.366	6410.366	6410.366
6415.854	6415.854	6415.854	6415.854	6415.854	6415.854	6415.854	6415.854
6465.536	6465.536	6465.536	6465.536	6465.536	6465.536	6465.536	6465.536
6448.919	6448.919	6448.919	6448.919	6448.919	6448.919	6448.919	6448.919
5939.703	5939.703	5939.703	5939.703	5939.703	5939.703	5939.703	5939.703
5447.224	5447.224	5447.224	5447.224	5447.224	5447.224	5447.224	5447.224
6316.745	6316.745	6316.745	6316.745	6316.745	6316.745	6316.745	6316.745
6473.598	6473.598	6473.598	6473.598	6473.598	6473.598	6473.598	6473.598
6626.981	6626.981	6626.981	6626.981	6626.981	6626.981	6626.981	6626.981
6634.461	6634.461	6634.461	6634.461	6634.461	6634.461	6634.461	6634.461
6465.953	6465.953	6465.953	6465.953	6465.953	6465.953	6465.953	6465.953
6151.223	6151.223	6151.223	6151.223	6151.223	6151.223	6151.223	6151.223

Figure 6. The operation of the windowing algorithm: *w* = 9, *h* = 1

1. Dense model (1). The first model tested was a standard ANN model with an input layer containing 128 units and the output layer one unit (the number of units in the output layer is always equal to the horizon value).

The Dense model diagram is presented in Figure 7.

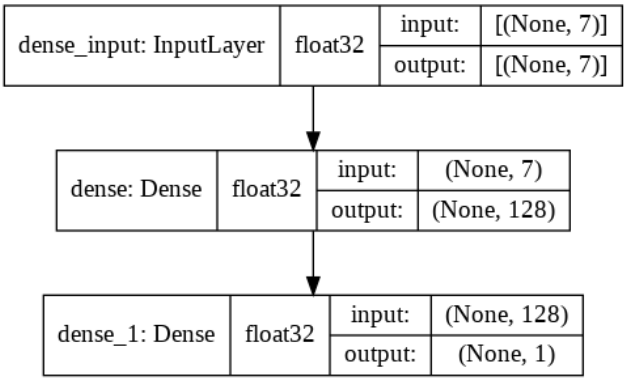

Figure 7. Dense model architecture (w = 7, h = 1)

As it can be observed in Figure 7, the input layer shape is 7, which corresponds to a window size of 7 days. This means that the model will consider a sequence of 7 days as features and the 8th day will be the label. The model was compiled using as loss function and optimizer the mean absolute error and Adam, respectively. The activation function was Rectified Linear Unit.

The training process was run for 100 epochs with a batch size of 128. As validation data, the test data was used. Once the training process was completed, predictions were carried out using the test data. The comparison between the real values and predictions is presented in Figure 8.

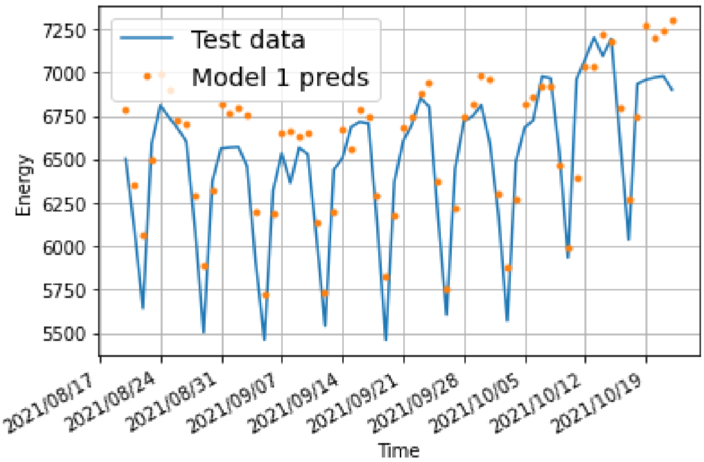

Figure 8. Model 1 (Dense) predictions vs. real values

2. Dense model (2). The second model was also a dense model with the same architecture as Model 1. The difference from the first model was the size of the window, which in this case was 30 days. The horizon size was kept to 1 day.

The comparison of the real test data to the Model 2 predictions is presented in Figure 9. It can be noticed that a significant improvement is achieved in the case of Model 2 by simply extending the window size from 7 to 30 days.

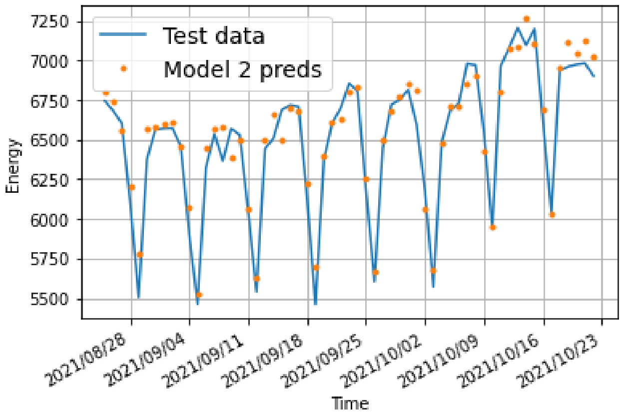

Figure 9. Model 2 (Dense) predictions vs. real values

3. Convolutional 1 D model (1). The third model consisted of a convolutional (1D) layer. The number of units was maintained (128) and the kernel size [1] was chosen 7.

After training the model and conducting predictions the results were plotted in Figure 10.

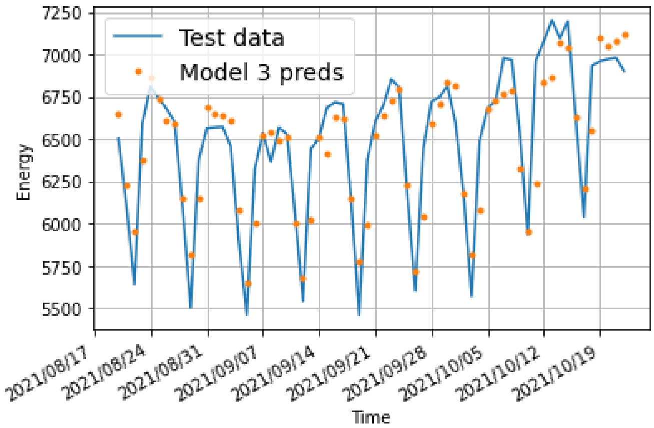

Figure 10. Model 3 (Conv1D) predictions vs. real values

4. Convolutional 1 D model (2). The same model architecture as model 3 was used with a window size of 30 days and the same horizon of 1 day. The comparison real values – Model 4 predicted values is presented in Figure 11.

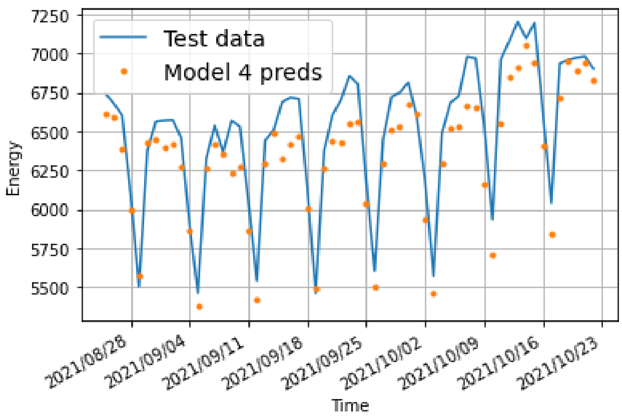

Figure 11. Model 4 real values versus predictions

5. LSTM model (1). The first type of recurrent neural network considered will be a LSTM network with one layer only and 128 units.

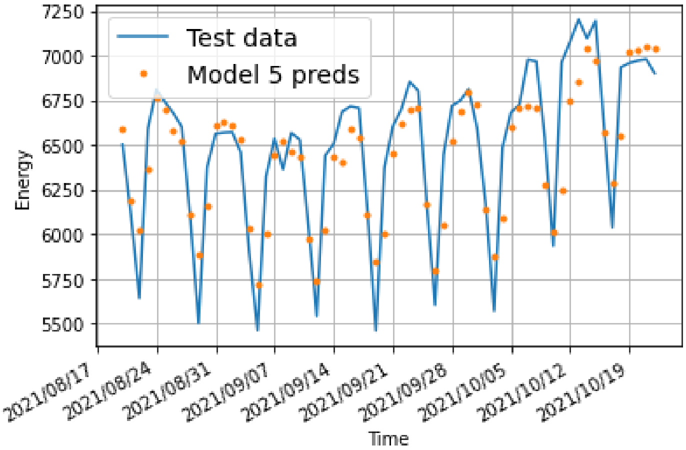

Figure 12. Model 5 real values versus predictions

6. LSTM model (2). A new LSTM model with two recurrent layers and a dense hidden layer will be the last variant. For this model, a number of 200 training epochs will be used. The model architecture is presented in Figure 13. The lambda layer shown in Figure 13 has the purpose of expanding the dimensionality of the input tensor (add one more dimension, as requested by the LSTM layer [6]). The effect of the lambda layer can be observed in Figure 13: it receives a tensor with the shape (,30) and produces at its output a tensor with the shape (,1,30).

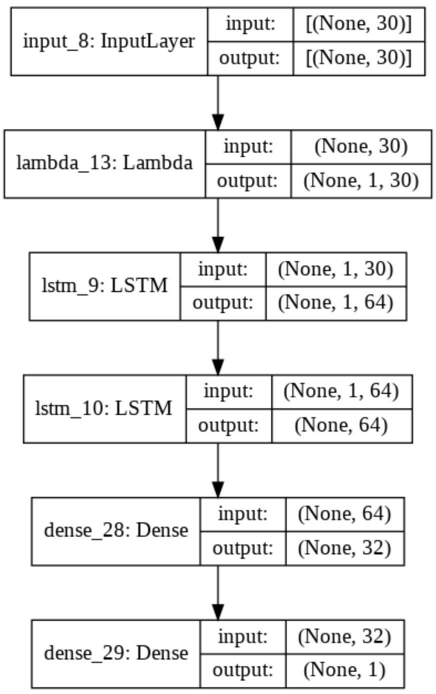

Figure 13. Model 6 architecture

The comparison between predictions and real values is presented in Figure 14.

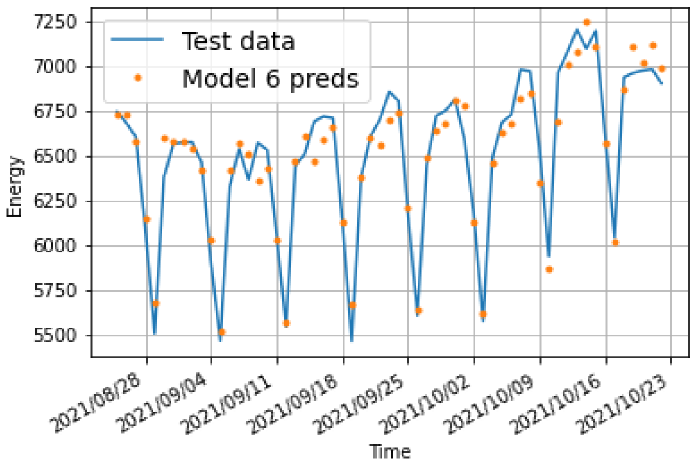

Figure 14. Model 6 predicted values versus real values

The model parameters and performance metrics are presented in Table 1. The LSTM model with two LSTM layers and one hidden fully-connected layer has the best performance (the lowest value of the mean absolute error).

Table 1. Comparison of models' performance

	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
Model parameters	Dense (128) w = 7 h = 1	Dense (128) w = 30 h = 1	Conv1D (128) kernel size=7 w = 7 h = 1	Conv1D (128) kernel size=7 w = 30 h = 1	LSTM(128) w = 7 h = 1	LSTM(64) LSTM(64) Dense(32) w = 30 h = 1
mae	225	162	215	218	226	156
mape	3.49	2.49	3.28	3.30	3.47	2.41
mase	0.623	0.448	0.593	0.603	0.625	0.432
mse	91232	48301	89727	75670	93889	51403
rmse	302	219	300	275	306	226

CONCLUSIONS

A critical comparison was carried out between 6 artificial intelligence models used to predict the electricity consumption in Romania. A dataset consisting of daily energy consumption over a period of approximately 7 years was used to train the models. Testing of the models was carried out over a period of several months in 2021. The objective of the study was to identify a model that can be further extended in order to achieve high forecasting accuracy. Several important observations can be derived from this preliminary study regarding the choice of the model parameters:

- Increasing the window size from 7 to 30 results in better prediction accuracy
- Increasing the number of epochs beyond 100 is expected to produce better forecasting accuracy
- Fully connected model (2) seems to produce comparable accuracy with LSTM model 6.

It is worthy to note that all models developed in this study are capable of distinguishing weekends from weekday, forecasting lower values for weekends.

REFERENCES

[1]Fazil Kaytez, M. Cengiz Taplamacioglu, Ertugrul Cam, Firat Hardalac, Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines, International Journal of Electrical Power & Energy Systems, vol. 67, 2015

[2]Unler A., Improvement of energy demand forecasts using swarm intelligence: the case of Turkey with projections to 2025. Energ Policy 2008;36:1937–44.

[3]Ceylan H, Ozturk HK., Estimating energy demand of Turkey based on economic indicators using genetic algorithm approach. Energy Convers Manage 2004;45:2525–37.

[4]Junhwa Hwang, Dongjun Suh, Marc-Oliver Otto, Forecasting Electricity Consumption in Commercial Buildings Using a Machine Learning Approach, Energies 2020, 13

[5]Geron, A., Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow. Concepts, Tools and Techniques to Build Intelligent Systems, 2nd Edition, O'Reilly 2019

[6]https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM?version=nightly, accessed on November 3rd, 2021

[7]Racoceanu., C. Study on the perspectives of coal burning in classic theroelectric power plants in Romania, Annals of the "Constantin Brâncuși" University of Târgu-Jiu. Engineering Series, 4 (2019), 96-99 [ISSN: 1842-4856].

[8]Racoceanu., C. Study concerning the application of the nonpolluting ash exhaust technology to the Benson steam generators, Annals of the "Constantin Brâncuși" University of Târgu-Jiu. Engineering Series, 4 (2019), 100-104 [ISSN: 1842-4856].

[9]Racoceanu Cristinel, Study on influence of energy efficiency of a steam boiler benson on environmental pollution, Fiability&Durability, no.2/2016, ISSN 1844-640X.