DOI: https://doi.org/10.15276/hait.05.2022.16
UDC 004.8

The study of the quality of multi-step time series forecasting

Petr M. Tishyn[1)]
ORCID: https://orcid.org/0000-0003-2506-5348; petrmettal@gmail.com. Scopus Author ID: 57190400970
Victor S. Buyukli[1)]
ORCID: https://orcid.org/0000-0001-7384-2290; vityabuyukli@gmail.com
[1)] Odessa Polytechnic National University, 1, Shevchenka Ave. Odessa, 65044, Ukraine

ABSTRACT

The work is devoted to the study of the quality of multistep forecasting of time series using the electricity consumption data for forecasting. Five models of multistep forecasting have been implemented, with their subsequent training and evaluation of the results obtained. The dataset is an upgraded minute-by-minute measurement of four years of electricity consumption. The dataset has been divided into training, validation, and test samples for training and testing models. The implementation is simplified by using the Tensor Flow machine learning library, which allows us to conveniently process and present data; build and train neural networks. The Tensor Flow functionality also provides standard metrics used to assess the accuracy of time series forecasting, which made it possible to evaluate the obtained models for forecasting the time series of electricity consumption and highlight the best of those considered according to the given indicators. The models are built in such a way that they can be used in studies of the quality of time series forecasting in various areas of human life. The problem of multistep forecasting for twenty four hours ahead, considered in the paper, has not yet been solved for estimating electricity consumption. The obtained forecasting accuracy is comparable to recently published methods for estimating electricity consumption used in other conditions. At the same time, the forecasting accuracy of the constructed models has been improved in comparison with other methods.

Keywords: Time series; forecasting; TensorFlow; electricity consumption; neural networks

***For citation*: Tishyn P. M., Buyukli V. S. The study of the quality of multi-step time series forecasting. *Herald of Advanced Information Technology*. 2022; Vol.5, No.3: 210–219. DOI: https://doi.org/10.15276/hait.05.2022.16**

INTRODUCTION

The term “time series” is understood to mean the sequence of data points that appear successive over a period of time. Time series analysis combines methods for studying time series, both allowing to understand the patterns common to data sets, and trying to build a forecast [1].

Time series modeling is widely used and proposed in various areas of improving the accuracy and efficiency of forecasting [2]. Forecasting is one of the goals of time series analysis by identifying a model based on previous data and assuming that current information will also come in the future.

Time series forecasting includes building a model to predict future values based on known past values and predict future data before it is available.

The development of forecasting methods is determined by the degree of mathematical description of the processes taking place in various branches of science and technology, having in mind mathematical achievements, technical limitations, the quality and volume of data sampling and resource constraints, including time ones [3].

Improving the accuracy of time series forecasting is an important but often challenging task. An effective way to improve forecasting accuracy can be to explore and use multiple models. At present, there is a growing need not only to improve the accuracy of modeling, but also to create qualitatively new models that take into account the nonlinear behavior of the observed research processes. Methods and tools for studying time series are rapidly progressing. Among modern tools, the Tensor Flow machine learning library can be noted [4].

It is recommended to take into account the features of the considered time series when forecasting. At the same time, one should not forget about the various types of forecasting. This can be either one-dimensional or multivariate forecasting, or multistep time series forecasting.

Forecasting can be applied to such time series as electricity consumption and prices, sales in retail chains, freight and passenger traffic, as well as to forecast road traffic (namely traffic jams) and market prices.

LITERATURE REVIEW

Until recently, statistical methods remained the main methods of time series forecasting. However, the mathematical models associated with these methods are usually linear, and therefore they cannot predict complex phenomena and processes in which the data model may be non-linear. In these cases, the apparatus of neural networks comes to the rescue [5]. Presently, there are various time series forecasting models: regressive and autoregressive models, neural network models, exponential smoothing models, models based on so-called Markov chains, classification models, etc. The most widely used of them are autoregressive and neural network models [6]. Forecasting the values of a time series is the task of extrapolating data, but in truth, neural networks are engaged in the task of interpolation, which significantly increases the reliability of the forecast [7]. The area of interval analysis [8] suggests that observations and estimates in the real world are typically incomplete or uncertain and therefore do not accurately represent real data. Interval data were also considered in the field of symbolic data analysis [8]. This field is related to multivariate analysis, pattern recognition and artificial intelligence, and seeks to extend classical exploratory data analysis and statistical methods to symbolic data.

Interval (multistep) forecasting is a kind of the forecasting of binary outcomes [9]. Various approaches have been introduced for analyzing interval data. A number of authors have considered neural network models for managing ones [10, 11], [12]. In [13], states of sleepiness were predicted using a polynomial logistic regression model, in which physiological and behavioral parameters, as well as subjective assessment of sleepiness, corresponded to independent variables and dependent variable, respectively. In [14], interval forecasting of large and rapid changes in wind energy (the so-called “ramps”) was considered, since they can affect the reliability of the power grids.

Energy consumption is skyrocketing all over the world as developing countries catch up with developed countries, and thus emissions of polluting gases into the atmosphere are higher than ever before. This effect will increase as the world's population grows and technology advances in today's society. In [15] the neural network is used for interval prediction of energy demand in buildings, which gives fairly accurate and reliable forecasts. In [16], an artificial neural network predicts solar energy in order to guarantee a reliable power supply and reduce environmental pollution through prediction algorithms and model similarities.

Research groups in their works [17, 18] used convolutional neural networks (CNN) to predict electricity demand. Convolution neural networks automates the learning of features based on input data and does not require additional effort from the researcher.

In [19], the authors proposed a model for predicting the electricity consumption of buildings in America using the architecture of a recurrent neural network (RNN), especially long short-term memory (LSTM). In [20], an LSTM-based model is proposed to study patterns of electricity consumption both for individuals and for different households. Long short-term memory overcomes the shortcomings of recurrent neural network because especially long short-term memory can extract more information from long input data [21].

The choice of models in the work is due to the most common approaches that were used in solving the problem of forecasting in this particular area.

Note that the problem of multi-step forecasting of time series of electricity consumption for 24 hours ahead has not yet been considered by the scientific community. Therefore, the development of an optimal accuracy model for multi-step forecasting of electricity consumption for 24 hours ahead could be considered as an urgent task. Forecast values for 24 hours or 1 day ahead are convenient for households, and can also be used to calculate the price of paying for electricity.

PURPOSE AND OBJECTIVES OF THE RESEARCH

The purpose of the study is to determine the structure of the optimal model for multistep forecasting of the time series of electricity consumption for 24 hours in advance.

To achieve this, the following tasks were set:
– choosing of the initial data (data set);
– preparation of the data for processing (normalization for submission to the model);
– creation (construction) of the architecture of models by adding predefined layers;
– training the models;
– evaluation of the obtained results.

DATA SET

The data set [22] is a multivariate time series reflecting electricity consumption over four years. This set is used to compare various models for forecasting the time series of electricity consumption [23, 24]. Electricity consumption measurements were recorded every minute.

In addition to date and time, the data set consists of seven variables (columns):

1) total active power (measured in kilowatts);

2) total reactive power (~ in kilowatts);

3) average voltage (~ in volts);

4) average value of current strength (~ in amperes);

5) active energy consumption of the kitchen room (~ in watt-hours of active energy);

6) active energy consumption of the laundry (~ in watt-hours of active energy),

7) active energy consumption of climate control systems (~ in watt-hours of active energy).

The statistical properties of the presented data set are summarized in Table 1.

Table 1. **Summary dataset statistics**

Column	Count	Mean	Std	Min	Max
1	34589	65.4	53.7	2.8	393.6
2	34589	7.4	4	0	46.5
3	34589	14449.6	197.3	719.1	15114
4	34589	277.1	224.8	11.4	1703
5	34589	67.1	211.6	0	2902
6	34589	77.5	250.5	0	2786
7	34589	386.9	440.5	0	1293

***Source*: compiled by the authors**

The dataset had been upgraded by:

– adding the variable called “residual active energy consumption” (calculated by subtracting the sum of three defined active energies from the total consumed energy);

– replacing the missing values by indications of the same time, but a day earlier;

– reducing the sample from every minute measurements of electricity consumption to hourly values.

The proposed changes to the original data set are necessary to “adjust” the data representation to the specifics of the models. It is necessary to have hourly values in the dataset to predict 24 hourly values.

As a result, 34589 records were obtained, dividing into training, validation and test (70, 20 and 10 percent, respectively) samples.

Data normalization occurs by subtracting the mean value and dividing by the standard deviation of each feature. However, the mean and standard deviation are only calculated using the training data so that the models do not “have access” to the validation and test sample values.

The data set is quite large. Therefore, the time series obtained for the observed value of *total active power* are presented using two time intervals of a smaller value. Fig. 1 shows a graph of the observed value of *total active power* in the interval from 1 to 1000 observation points. Fig. 2 shows the same observed value in the interval from 14750 to 15750 observation points.

From the presented figures, it can be seen that the data set has a high variability, non-linearity and multidirectional trends at certain intervals. All this limits the use of statistical methods. Fig. 3 shows the observed value of *total reactive power* in the interval from 1 to 1000 observation points.

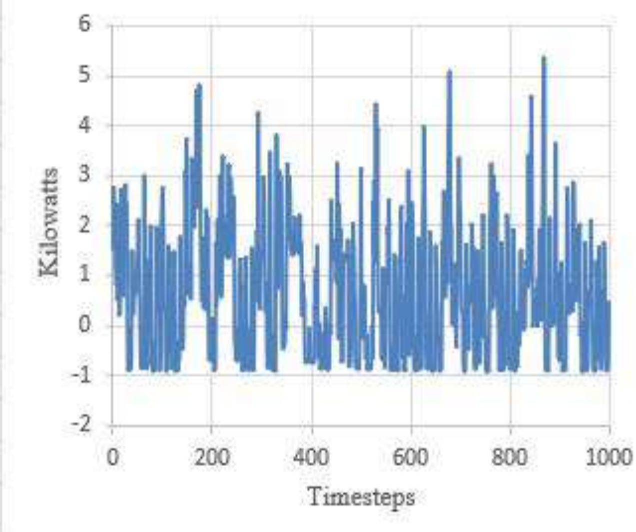

Fig. 1. **Graph of total active power in the interval 1 to 1000**
***Source:* compiled by the authors**

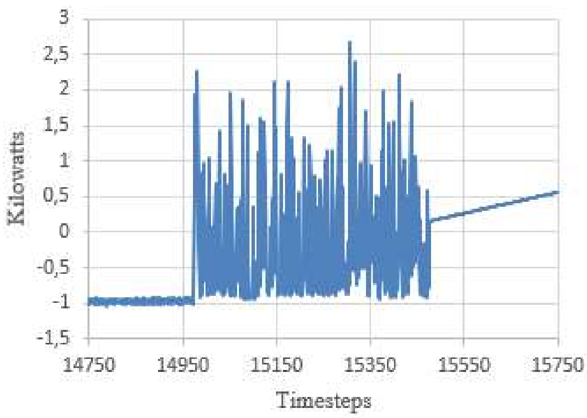

Fig. 2. **Graph of total active power in the interval 14750 to 15750**
***Source:* compiled by the authors**

Fig.4, Fig.5 and Fig.6 present the time series of observed values of *active energy consumption of the kitchen room, active energy consumption of the laundry* and *active energy consumption of climate control systems* over the interval from 1 to 1000 observation points.

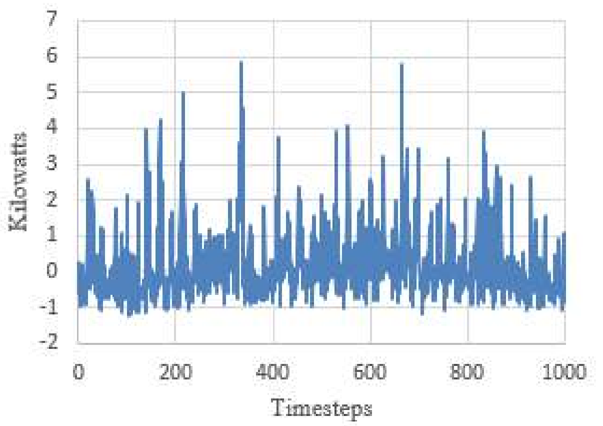

Fig. 3. **Graph of total reactive power in the interval 1 to 1000**
Source: **compiled by the authors**

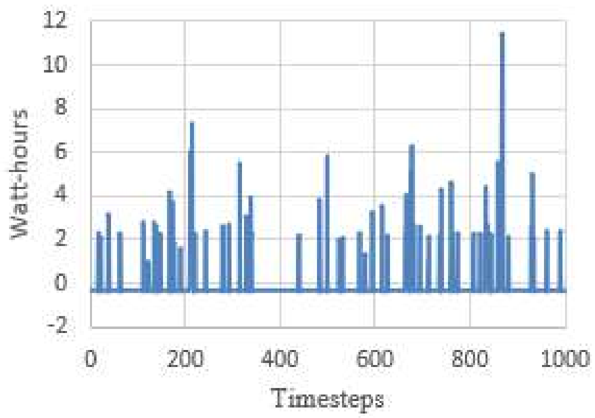

Fig. 4. **Graph active energy consumption of the kitchen room in the interval 1 to 1000**
Source: **compiled by the authors**

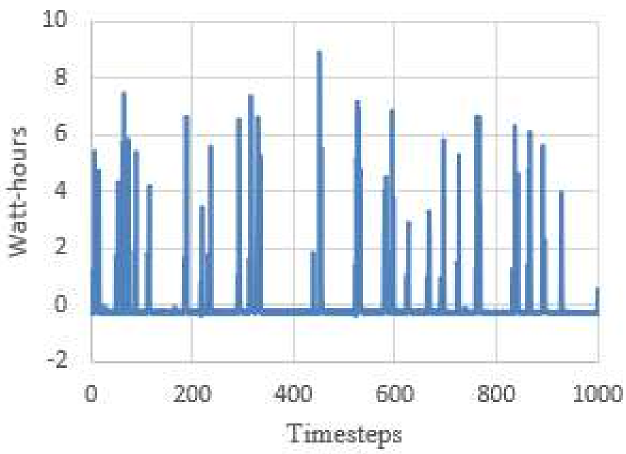

Fig. 5. **Graph of active energy consumption of the laundry in the interval 1 to 1000**
Source: **compiled by the authors**

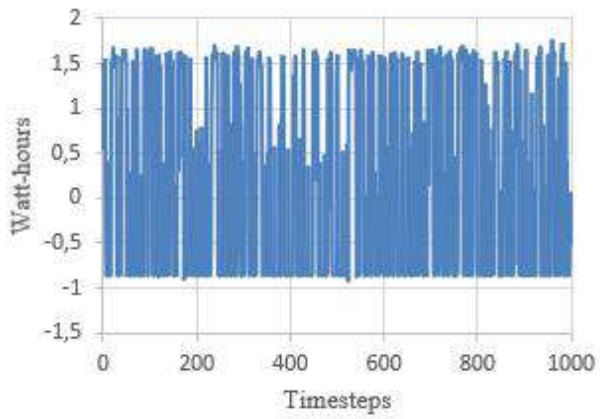

Fig. 6. **Graph of active energy consumption of climate control systems in the interval 1 to 1000**
Source: **compiled by the authors**

The pmdarima library was used to compare the quality of the presented models with other approaches. The pmdarima library works by generalizing all ARMA, ARIMA, SARIMAX models into one class. The library provides the auto_arima() function, which builds the best predictive model given the criteria. The time series obtained for the observed value of *total active power* was divided into two sets. As a result, 34589 records are divided into training and test (70% and 30%, respectively) samples. A model for forecasting the values of the time series was built on the training sample. The best model was the ARIMA(1,1,3) model. This model allows us to evaluate the accuracy of time series forecasting for the observed value of *total active power* on the test sample.

PREDICTION MODELS

In the work, the models are built for interval forecasting of time series of electricity consumption: a linear model (Linear), Dense-model, convolutional model (Conv), long short-term memory model (LSTM) and autoregressive model (AR LSTM). These models include convolutional and recurrent neural networks.

Architecture and description of models

Each model predicts the future values of the total active power for 24 hours.

1) A linear model predicts time steps based on a single input time step with a linear projection. The layers used here are: Lambda, Dense (fully connected) and Reshape.

The Lambda layer is used so that arbitrary (user customized) expressions can be applied as a separate layer when building the sequential models. In this and other models, the layer is used to represent model data in the desired form; it acts as an input one.

Fully connected or Dense layer. The neurons of this layer are connected to every neuron of the previous layer. That is, the Dense layer receives information from all nodes of the previous one.

The Reshape layer reshapes the input data into a given shape (24 metrics).

2) The Dense model can be considered an improvement on the previous model by adding another Dense layer with a relu activation function between the Lambda and Dense layers.

The added Dense layer consists of units of neurons (512) connected by synapses to the elements of the input tensor by its last index.

The Dense layer implements the operation:

output = activation(dot(input, kernel) + bias), (1)

where *activation* – is the element-wise activation function passed as the activation argument;

dot – matrix product;

input – input vector;

kernel –the weight matrix created by the layer

bias – the bias vector created by the layer (applicable only if the boolean value of the use_bias parameter is True).

The three most common activation functions are sigmoid (or sig), hyperbolic tangent (tanh) and a variant of ramp function called the Rectifying Linear Unit (ReLU) [25].

ReLU ("relu" further) was chosen because such a function is a good approximator. Also, relu is less computationally demanding than hyperbolic tangent or sigmoid, as it performs simpler mathematical operations.

This function return 0 if it takes a negative argument, in the case of a positive argument, the function returns the number itself:

$$f(x) = max(0, x). \quad (2)$$

It should be noted that relu is also used to create deep neural networks (including CNNs).

3) Convolutional model (Conv) makes predictions based on fixed width history. The layers applied here are: Lambda, Conv1d (activation function: relu), Dense, Reshape.

The Conv1d convolution layer is the main building block of a CNN (Fig.7), which creates a kernel (1x3) that is convolved with the input of the layer in a single spatial (or temporal) dimension to obtain a tensor of the original data (summing the results of the element-wise product for each fragment).

In Conv1d, the kernel slides along one dimension. The weights of the convolution kernel (small matrix) are unknown and supposed to be setting during the training.

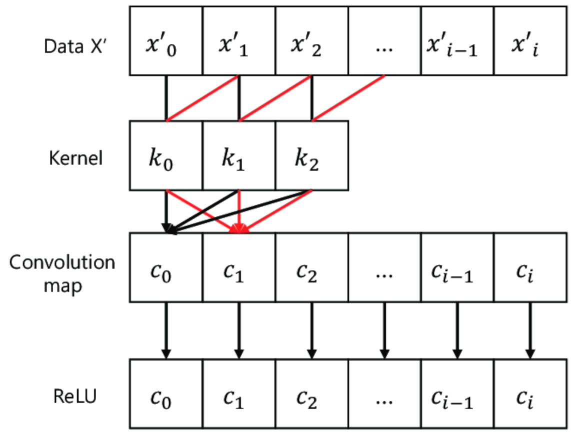

Fig. 7. **Operation of a one-dimensional convolution layera**
Source: **[26]**

The Conv1d layer is used as it can identify simple patterns in the data well and then use those simple patterns to create more complex ones. This helps to extract features of interest more efficiently from shorter (fixed length or kernel size) chunks in the overall dataset.

4) The next model is a kind of architecture of recurrent neural networks – LSTM model.

The LSTM model accumulates 24 hours of internal state before making one prediction for the next 24 hours. Layers: LSTM (long short-term memory layer), Dense, Reshape. The LSTM layer architecture is shown in Fig. 8.

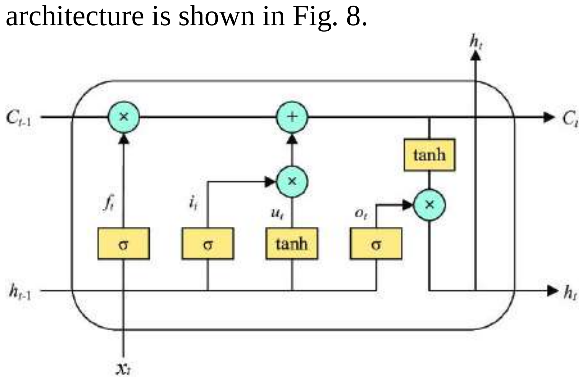

Fig. 8. **Long short-term memory model layer architecture**
Source: **[27]**

The LSTM layer "learns" long-term relationships between time steps in a time series and sequence data.

The layer can be described as follows:

$$\begin{aligned} f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \\ i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\ o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\ h_t &= o_t \circ \sigma_h(c_t), \end{aligned} \quad (3)$$

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

where x_t is the input vector; h_t is the output vector ($h_0 = 0$); c_t is the state vector ($c_0 = 0$); W, U, b are the parameter matrices; f_t is the forgetting gate vector, the weight of remembering old informatione; i_t is the input gate vector, the weight of obtaining new information; o_t is the output gate vector, a candidate for the output; σ_g is the activation function based on the sigmoid; σ_c, σ_h are the activation functions based on hyperbolic tangent; $\circ$ is the Hadamard product.

5) An autoregressive model (AR LSTM) has the same basic form as an LSTM: an LSTM layer followed by a Dense layer. The difference is that the LSTM belongs to the LSTMCell class, and so it is "wrapped" in a top-level RNN-class layer that manages state and sequence results.

An autoregressive model is a time series model in which the values of a time series at a given moment linearly depend on the previous values of the same series. That is, a time series model is used in which its current value linearly depends on the previous (retrospective) values of the same series. A linear relationship means that the current value is equal to the weighted sum of several previous values in the series.

The LSTMCell is a cell class for the LSTM layer. This class processes one step in the entire time sequence input, while LSTM processed the entire sequence at a time.

The RNN is a base class for repeating layers, which takes as a parameter a cell instance or a list of cell instances; as well as the boolean value return_state = True, which indicates that the input sequence is processed in reverse order and the reverse sequence is returned.

As a cell, the layer accepts the cell class for the LSTM layer – LSTMCell, which indicates the dimension of the output space, units = 24. In this case, the activation function to use is specified by default: hyperbolic tangent. The activation function for the repeated step is sigmoid. In this case, the fraction of units that must be discarded for the linear transformation of the recurrent state is zero, that is, it is not used.

COMPILING AND TRAINING MODELS

Of importance for the implementation of classes and functions is the use of TensorFlow. TensorFlow is an open-source machine learning software library that allows a user to conveniently process and feed data; build and train neural networks [28]. Criteria for choosing this library: the Tensor Flow library today is one of the standard tools for working with artificial neural networks.

In the models, the data is divided into three sets: training, validation, and delayed (test) samplings.

The training data is used to train the model, the validation data is used to find the best model architecture, and the delayed one is reserved for the final evaluation of the model. That is, the first two are involved in network training: the training set is used to optimize the model weights, and the validation set provides metrics after each training epoch (iteration) that help evaluate the quality of model training. The test set is needed to compare classification accuracy among different models. Thus, test data is still useful in determining how well a model will generalize what it has learned to new data [11].

Before compiling the model, it is necessary to determine the optimizer, that is, on a specific algorithm that updates the weights in the process of training the model.

The Stochastic Gradient Descent (SGD) algorithm involves updating the neural network weights using a single training sample at each step. SGD does not perform "excessive" calculations, since, unlike the classical gradient descent, the algorithm's error function is calculated not over the entire training set, but only over one example.

However, due to the fact that at each step of the SGD the calculation of the gradient (the largest increase in some value) is based on different examples of the original data set, updating the weight coefficients is accompanied by frequent fluctuations in the objective function. One of the possible solutions to this problem is the dynamic modification of the learning rate, implemented in a group of adaptive optimization algorithms, including RMSProp and Adam [29].

The root-mean-square propagation (RMSProp) learning rate is configurable for each parameter. RMSProp has shown excellent adaptation of the learning rate across applications.

Adam or the adaptive moment estimation method is an "upgrade" of the RMSProp optimizer. The weight update rule for Adam is determined based on the use of estimates of two different moments, in the first of which the values of partial derivatives calculated earlier are used (as in the method of moments), and in the second their squares (as in RMSProp).

It can be concluded that the Adam method is sufficiently resistant to the choice of hyperparameter values. In this regard, it would be most appropriate to choose Adam as an optimizer, given its advantages over SGD and RMSProp and the use of their best characteristics.

As an objective function, an estimate of the root-mean-square error was chosen, which averages the sum of the squared differences between the predicted and true values:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\gamma_i - Y_i)^2 , \qquad (4)$$

where n – is the number of forecasts;

γ_i – is the i -th forecast;

Y_i – is the i -th observed value.

The objective function is used to calculate the errors between the actual and received values. If the forecast is very different from the true value, then squaring makes the difference even more significant and visible.

Models were trained with the following parameters set: series size is 32, the number of epochs is 50. For compilation, Adam was chosen as an optimizer (the parameters are presented in Table 2) the objective function is MeanSquaredError (MSE).

Table 2. **Parameters of Adam optimizer**

Parameter	Value
Learning rate	0.001
Biased moment estimation update factors	0.9; 0.999
Gradient interval	0.5
Learning rate reduction factor	0

***Source*: compiled by the authors**

RESULTS EVALUATION

The TensorFlow functionality also provides different model evaluation metrics. Mean absolute error (MAE) and root-mean-square deviation (RMSE) were chosen as the metrics for evaluating the obtained models.

MAE measures the average sum of the absolute difference between the actual value and the predicted value:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|or_t - pr_t\right| , \qquad (5)$$

where or_t is the actual normalized value; pr_t is the predicted normalized value.

RMSE root-square of MSE:

$$RMSE = \sqrt{MSE} . \qquad (6)$$

First of all, the problem of determining the best structure in all models is solved. For this, models were trained for multistep prediction of the values of the total active power with the input of one variable (total active power). The results of the metrics are shown in Fig. 9 and Fig. 10.

It can be seen from the obtained diagrams that the LSTM and AR LSTM models, both for MAE and RMSE, showed a good result in predicting the time series of electricity consumption on this data set. The LSTM model has the best performance among the other considered models on the test sample.

Additionally, the models were trained with input:

– 3 variables (total active power, total reactive power and average current value);

– 7 variables (average voltage is not fed to the model);

– 8 variables.

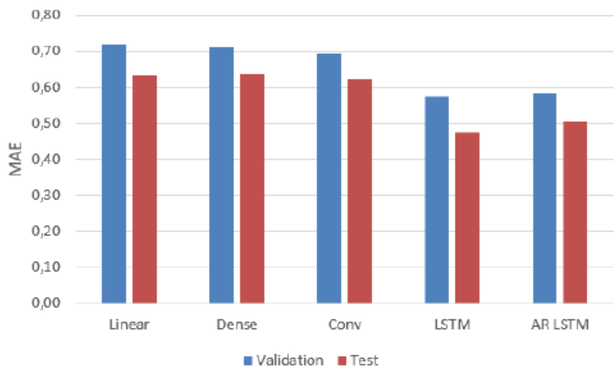

Fig. 9. **Indicators of the mean absolute error of models**
***Source:* compiled by the authors**

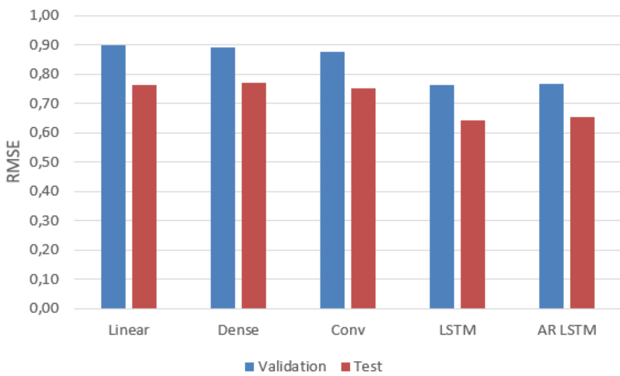

Fig. 10. **Indicators of the root-mean-square deviation of models**
***Source:* compiled by the authors**

The results are presented in Table 3 as test sample metrics.

Table 3. **Indicators of model metrics on the test sample**

Number of variables to be fed	Metric	Model				
		Linear	Dense	Conv	LSTM	AR LSTM
1	MAE	0.6308	0.6366	0.6214	0.4739	0.5042
	RMSE	0.7638	0.7707	0.7505	0.6427	0.6524
3	MAE	0.6327	0.6319	0.6086	0.5288	0.5244
	RMSE	0.8094	0.8065	0.7878	0.7253	0.7254
7	MAE	0.5647	0.5455	0.5271	0.4824	0.4717
	RMSE	0.7883	0.7758	0.7608	0.7182	0.7308
8	MAE	0.5689	0.5557	0.5315	0.4850	0.4899
	RMSE	0.7904	0.7824	0.7656	0.7152	0.7259

***Source*: compiled by the authors**

From the analysis of Table 3, we can conclude that when using the RMSE metric, the best results were shown by the LSTM model from one input variable (0.6427). The features of this model make it possible to better take into account the initial data compared to other models under consideration. With an increase in the number of submitted variables, the results did not change for the better.

When using the MAE metric, the smallest indicator (0.4717) was obtained for the AR LSTM model when seven variables were fed to the input of the model.

We also note that although similar tasks of multistep forecasting of the time series of electricity consumption for 24 hours in advance have not yet been solved by the scientific community, these errors are comparable to the errors obtained in [23, 24].

To compare the accuracy of the obtained results with the ARIMA (1, 1, 3) model described in the DATA SET section, it can be noted that the RMSE error for this model was 0.9026 and the MAE error was 0.7108.

CONCLUSIONS

Th paper presents five interval forecasting models for the time series of electricity consumption, implemented using TensorFlow. The values of the total active power were predicted. The LSTM model with one input variable (total active power) on the test set has RMSE values less than other models (0.6427).

The AR LSTM model also performed well in comparison. The lowest MAE (0.4717) was obtained for the AR LSTM model when seven variables were fed into the model input. At the same time, the forecasting accuracy of the constructed models has been improved in comparison with other methods.

Thus, the optimal forecasting models for this problem among those considered are LSTM and AR LSTM. The created models can be applied to time series studies in other areas..

REFERENCES

1. De Gooijer, J. G. & Hyndman, R. J. “25 years of time series forecasting”. *International journal of forecasting*. 2006; 22(3): 443–473. DOI: https://doi.org/10.1016/j.ijforecast.2006.01.001.

2. Shumway, R. H. & Stoffer, D. S. Time Series and Its Applications. 3rd ed. Springer. 2017.

3. Shchelkalin, V. N. “Hybrid mathematical models and methods for time series forecasting taking into account external factors” (in Russian). *Eastern European Journal of Advanced Technology.* 2014; Vol. 6 No. 4 (72): 38–58. DOI: https://doi.org/10.15587/1729-4061.2014.31729.

4. “TensorFlow official website”. – Available from: https://www.tensorflow.org. – [Accessed: Sep. 2021].

5. Devyatkov, V. V. Mateychuk, R. A. Mishchenko, I. I. & Kuznetsov, N. A. “Modern information technology”. *7th International Scientific Conference of Students and Young Scientists* (in Ukrainian). Odessa National Polytechnic University. 2017. p.11–12

6. Nguyen, T. K. T., Antoshchuk, S. G., Nikolenko, A. A., Tran, K. T. & Babilunha, O. Yu. “Non-stationary time series prediction using one-dimensional convolutional neural network models”. *Herald of Advanced Information Technology*. 2020; 3 (1): 362–372. DOI: https://doi.org/10.15276/hait.01.2020.3.

7. Efremova, E. A. & Dunaev, E.V. “Application of neural networks for forecasting financial time series” (in Russian). *Reports of TURSUR*. 2004. p. 192–196.

8. Nguyen, H. T., Kreinovich, V., Wu, B. & Xiang, G. "Computing statistics under interval and Fuzzy Uncertainty. *Applications to Computer Science and Engineering*. *Springer.* Berlin: Heidelberg. 2012. *Bock, H. H. & Diday, E. "Analysis of symbolic data". Springer.* Berlin: Heidelberg. 2000.

9. Lahiri, K., Yang, L., Elliott, G. & Timmermann, A. "Forecasting binary outcomes. Hand book of economic forecasting". 2013; 2: 1025–1106. DOI: https://doi.org/10.1016/B978-0-444-62731-5.00019-1.

10. Patin˜o-Escarcina, R. E., Bedregal, B. R. C. & Lyra, A. "Interval computing in neural networks: one layer interval neural networks". *In: G. Das, V.P. Gulati (Eds.), Proceedings of the Seventh International Conference on Information Technology*. CIT, Hyderabad: India. 2004. p. 68–75.

11. Roque, A. M., Mater, C., Arroyo, J. & Sarabia Ar. "iMLP: applying multi-layer perceptrons to interval-valued data". *Neural Process. Lett.* 2007; 25: 157–169. DOI: https://doi.org/10.1007/s11063-007-9035-z.

12. Rossi, F. & Conan-Guez, F. B. "Multilayer perceptron on interval data". *In: K. Jajuga, A. Sokolowski, H.H. Bock (Eds.). Classification, Clustering, and Data Analysis (IFCS 2002).* Cracow: Poland. 2002. p. 427–434.

13. Murata, A., Fujii, Y. & Naitoh, K. "Multinomial logistic regression model for predicting driver's drowsiness using behavioral measures". *6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences*. 2015; 3: 2426–2433. DOI: https://doi.org/10.1016/j.promfg.2015.07.502.

14. Cui, M., Ke, D. & Sun, Y. "Wind power ramp event forecasting using a stochastic scenario generation method". *IEEE Transactions on sustainable energy*. 2015; 6: 422–433. DOI: https://doi.org/10.1109/TSTE.2014.2386870.

15. Ryu, S., Noh, J. & Kim, H. "Deep neural network based demand side short term load forecasting". *Energies*. 2017; 10 (1): 3. DOI: https://doi.org/10.3390/en10010003.

16. Torres, J. F., Troncoso, A., Koprinska, I., Wang, Z. & Martínez-Álvarez, F. "Deep learning for big data time series forecasting applied to solar power". *In International Joint Conference SOCO'18-CISIS'18-ICEUTE'18. SOCO'18-CISIS'18-ICEUTE'18. Advances in Intelligent Systems and Computing,*. Springer. Cham. 2018; 771: 123–133. DOI: https://doi.org/10.1007/978-3-319-94120-2_12.

17. García-Ascanio, C. & Maté, C. "Electric power demand forecasting using interval time series: A comparison between VAR and iMLP". *Energy Policy*. 2010; 38: 715–725. DOI: https://doi.org/10.1016/j.enpol.2009.10.007.

18. Kang, T., Lim, D., Tayara, H., & Chong, K. "Forecasting of power demands using deep learning". *Applied Sciences*, 2020; 10 (20): 7241. DOI: https://doi.org/10.3390/app10207241.

19. Wang, X., Fang, F., Zhang, X., Liu, Y., Wei L. & Shi Y. "LSTM-based short-term load forecasting for building electricity consumption". *In 2019 IEEE 28th International Symposium on Industrial Electronics (ISIE). IEEE.* 2019. p. 1418–1423. DOI: https://doi.org/10.1109/ISIE.2019.8781349.

20. Alonso, A. M., Nogales, F. J. & Ruiz, C. "A single scalable LSTM model for short-term forecasting of disaggregated electricity loads". 2019. DOI: https://doi.org/10.48550/arXiv.1910.06640.

21. Xu, K., Hou, R., Ding, X., Tao, Y. & Xu, Z. "Short-term time series data prediction of power consumption based on deep neural network". *IOP Conference Series: Materials Science and Engineering.* 2019; 646 (1): 1–10. DOI: https://doi.org/10.1088/1757-899X/646/1/012027.

22. "Individual household electric power consumption dataset". – Available from: https://cutt.ly/KFG6P52. – [Accessed: Sep. 2021].

23. Qin, J. "Experimental and analysis on household electronic power consumption". *In 2021 The 2nd International Conference on Power Engineering (ICPE 2021).* Nanning, Guangxi, China, 2021; 8 (5): 705-709. DOI: https://doi.org/10.1016/j.egyr.2022.02.270.

24. Parate, Aaditi & Bhoite, Sachin. "Individual Household Electric Power Consumption Forecasting using Machine Learning Algorithms". *International Journal of Computer Applications Technology and Research*. 2019; 8 (9): 371–374. DOI: https://doi.org/10.7753/IJCATR0809.1007.

25. Giulli, A. & Pal, S. "Keras library – a deep learning tool. Implementing neural networks using Theano and TensorFlow libraries". Moscow: *DMK Press.* 2018. 294 p.

26. Nam, S., Park, H., Seo, C. & Choi, D. "Forged signature distinction using convolutional neural network for feature extraction". *Appl. Sci.* 2018; 8: 153. DOI: https://doi.org/10.3390/app8020153.

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

27. Jin, Y., Xie, J., Guo, W., Luo, C., Wu, D. & Wang, R. "LSTM-CRF neural network with gated self attention for chinese NER". *IEEE Access*. 2019; 7: 136694–136703. DOI: https://doi.org/10.1109/ACCESS.2019.2942433.

28. Russell, S. & Norvig, P. "Artificial intelligence: A modern approach". *Prentice Hall*. 2003. 492 p.

29. Kingma, D. P. & Ba, J. L. "Adam: a method for stochastic optimization". *International Conference on Learning Representations*. 2015. p. 1–13. DOI: https://doi.org/10.48550/arXiv.1412.6980.

Conflicts of Interest: the authors declare no conflict of interest

Received 22.08.2022
Received after revision 05.10.2022
Accepted 17.10.2022

DOI: https://doi.org/10.15276/hait.05.2022.16
УДК 004.8

Дослідження якості багатокрокового прогнозування часових рядів

Тішин Петро Метталинович[1)]
ORCID: http://orcid.org/0000-0003-2506-5348; petrmettal@gmail.com. Scopus Author ID: 57190400970
Буюклі Віктор Сергійович[1)]
ORCID: http://orcid.org/0000-0001-7384-2290; vityabuyukli@gmail.com
[1)] Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

Робота присвячена дослідженню якості багатокрокового прогнозування часових рядів. Для прогнозування застосовуються дані споживання електроенергії. Виконано реалізацію п'яти моделей багатокрокового прогнозування з подальшим їх навчанням та оцінкою отриманих результатів. Набір даних є модернізованими щохвилинними вимірюваннями показників споживання електроенергії за чотири роки. Дані розділені на навчальну, валідаційну та тестову вибірки для навчання та тестування моделей. Реалізація спрощена завдяки використанню бібліотеки машинного навчання TensorFlow, що дозволяє зручно обробляти та подавати дані; будувати та навчати нейронні мережі. Функціонал TensorFlow надає і стандартні метрики, які застосовуються для оцінки точності прогнозування часових рядів, що дозволило оцінити отримані моделі прогнозування часового ряду споживання електроенергії та виділити найкращу із розглянутих за показниками. Моделі побудовані таким чином, що можуть бути застосовані у дослідженнях якості прогнозування часових рядів різних галузей життєдіяльності людини. Задача багатокрокового прогнозування на 24 години вперед, що розглядається в роботі, ще не вирішувалося для оцінки споживання електроенергії. Отримана точність прогнозування зіставна з опублікованими останнім часом методами оцінки споживання електроенергії, що застосовуються в інших умовах. При цьому покращено точність прогнозування побудованих моделей в порівнянні з іншими методами.

Ключові слова: часові ряди; прогнозування; TensorFlow; споживання електроенергії; нейронні мережі

ABOUT THE AUTHORS

Petr M. Tishin – PhD (Physico-Mathematical), Associate Professor of Computer Intellectual Systems and Networks Department. Odessa Polytechnic National University, 1, Shevchenko Ave. Odessa, 65044, Ukraine
ORCID: http://orcid.org/0000-0003-2506-5348; petrmettal@gmail.com. Scopus Author ID: 57190400970
Research field: Artificial intelligence methods and systems

Тішин Петро Метталинович – кандидат фізико-математичних наук, доцент кафедри гомп'ютерних інтелектуальних систем та мереж. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна

Victor S. Buyukli – Student of the Computer Intellectual Systems and Networks Department. Odessa Polytechnic National University, 1, Shevchenko Ave. Odessa, 65044, Ukraine
ORCID: http://orcid.org/0000-0001-7384-2290; vityabuyukli@gmail.com
Research field: Artificial intelligence methods and systems

Буюклі Віктор Сергійович – студент кафедри Комп'ютерних інтелектуальних систем та мереж. Національний університет «Одеська політехніка», пр. Шевченка, 1. Одеса, 65044, Україна