International Journal on Emerging Technologies **10**(2a): 188-195(2019)

ISSN No. (Print) : 0975-8364
ISSN No. (Online) : 2249-3255

The use of Recurrent Neural Networks to Solve a Regression Type Problem

Marat Rashitovich Safiullin[1], Leonid Alekseevich Elshin[2] and Ayzat Minekhanovich Gilmanov[3]
[1]Vice Rector for Economic and Strategic Development, Doctor of Economics, Professor. Kazan Federal University Director of center, Head University/Institute of Management, Economics and Finance, senior research fellow, Doctor of science, Kazan Federal University, Russia.
State Budgetary Institution Center "Perspective Economic Researches", Academy of Sciences, Republic of Tatarstan Kazan National University of Science and Technology
Department of Economics, Kazan Federal University, Russia.
[2]State Budgetary Institution Center of Perspective Economic Researches, Academy of Sciences, Republic of Tatarstan, Kazan National University of Science and Technology, Russia.

(Corresponding author: Marat Rashitovich Safiullin)
Received 05 May 2019, Revised 16 July 2019 Accepted 25 July 2019)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: The development of the cryptocurrency market and its integration into the system of economic, operational, financial and other processes determine the need for a comprehensive study of this phenomenon of particular relevance to this is the fact that in recent months, discussions on the prospects for legalizing the cryptocurrency market and the possibilities of using its tools in the economic activity of economic agents have been intensified at the state level. Despite sometimes controversial views and approaches that have emerged at the moment among Russian experts regarding the solution of this issue, the development of the crypto market is independent of its state regulation. This causes and actualizes the conduct of scientific research in the field of studying the main parameters and prospects for the crypto market development, including through the use of mathematical analysis methods.
The paper deals with the problem of predicting the trend of financial time series using the LSTM neural network. The time series compiled from the BTC/USD currency pair is analyzed, the timeframe is a day. The authors analyzed the neural network architecture, built a neural network model taking into account the heterogeneity and random volatility of the time series, developed and implemented an algorithm for solving the problem in the Python system. For training the neural network, data for the period from 24/09/2013 to 17/03/2019 (a total of 2002 data sets) were used.
The experiment boils down to the fact that the constructed neural network model tries to determine the trend of the time series for one next timeframe. The training with a "teacher" was conducted. To determine the prediction error, the root-mean-square error (RMSE) was calculated.
The research results are represented in tabular and graphical form.

Keywords: time series; prediction; artificial neural networks; artificial neural network architecture; LSTM.

I. INTRODUCTION

Direct distribution networks, or multilayer perceptrons, have a fixed number of inputs, and each of them is perceived by the others as an independent [1-5]. However, in recurrent networks, the communication between neurons is not limited solely to the movement of information in one direction, but it is also possible to return the value to "myself" [6, 7]. Thus, a neuron can memorize information that was previously submitted to its input. That is why recurrent neural networks are the best choice for predicting time series and sequences. By the nature of the inputs and outputs, tasks are divided into five options:
- one input, one output (one-to-one);
- one input, sequence of outputs (one-to-many);
- a sequence of inputs, one output (many-to-one);
- input sequence, output sequence (many-to-many);
- synchronized sequences of inputs and outputs (synchronized many-to-many).

II. METHODS

This paper solves the "many-to-many" regression problem when training with a "teacher" using the recurrent LSTM layer.

Conventional recurrent networks very poorly cope with situations where something needs to be "remembered" for a long time: the influence of the hidden state or the input from the step number t on subsequent states of the recurrent network exponentially decays. That is why the LSTM (Long Short-Term Memory) model is used in this study, where an additional cell is added to simulate "long memory".

III. RESULTS AND DISCUSSION

To construct a neural network that predicts the rate in the BTC /USD currency pair one day in advance, data from was taken [11].
An example of the input data is presented in Table 1.

Input data:
Low - the lowest price in the time period t,
High - the highest price in the period of time t,
Open - the opening price in the time period t,
Close - the closing price in the time period t,
Volume from - trading volume in bit coins in the period of time t,
Volume to - trading volume in dollars in a period of time t.

Output :
Low - the lowest price in the period of time t + 1,
High - the highest price in the period of time t + 1,
Close - the closing price in the time period t + 1.
To improve the performance of neural networks, the authors used data normalization within [0: 1] [8]. The result is presented in Table 2.
The structure of the constructed model is as follows:
1. Input layer;
2. Hidden LSTM - layer;
3. Output layer
Also, for the three models that showed the best results (with the smallest RMSE value), neural network training diagrams, and also projected data and historical data comparison diagrams were constructed.
Python software and libraries such as keras, pand as, numpy, sklearn, matplotlib were used to build the network.
To support GPU computing, data have been converted to float 32 format.

Table 1: Sample input data that discloses information about the current rate of Bitcoin cryptocurrency. BTC/USD.

Date time	Low	High	Open	Close	Volume from	Volume to
2013-09-24	132.5	136.59	133.4	134.78	8575.98	1148734.45
2013-09-25	134.7	138.0	134.78	135.0	9517.07	1293657.92
...	...	...	...	...	...	...
2019-03-16	3924.98	4069.11	3927.08	4027.01	41319.81	166216280.52
2019-03-17	3970.97	4030.38	4027.01	3994.18	8442.71	33698596.43

Table 2: Normalized data disclosing information about the current rate of Bit coin crypto currency, BTC/USD.

Date time	Low	High	Open	Close	Volume from	Volume to
2013-09-24	0.00218873	0.000232539	0.00113543	0.00120724	0.0157957	0.000150287
2013-09-25	0.00230663	0.000303973	0.00120717	0.00121868	0.0176149	0.000173492
...	...	...	...	...	...	...
2019-03-16	0.205438	0.199462	0.198363	0.20357	0.0790897	0.02658
2019-03-17	0.207903	0.1975	0.203558	0.201863	0.0155381	0.005362

Table 3: Results of the 15 best experiments.

Loss	Activation	Optimizer	Neurons	Epochs	Batch_size	Dropout	RMSE
logcosh	relu	Nadam	16	488	216	17.00	139.35
logcosh	linear	Nadam	26	310	100	19.00	139.87
mse	linear	Nadam	6	381	80	3.00	141.21
mse	linear	Nadam	14	373	80	8.00	141.34
mae	tanh	Adadelta	5	494	110	11.00	141.71
logcosh	tanh	Nadam	30	382	144	0.00	142.44
mae	linear	Nadam	12	455	123	4.00	142.48
mse	selu	Nadam	21	231	116	1.00	142.95
logcosh	selu	Nadam	8	486	115	2.00	143.25
mse	elu	Nadam	22	449	144	15.00	143.39
logcosh	elu	RMSprop	ten	429	137	17.00	143.57
msle	linear	Nadam	26	461	116	17.00	143.67
mse	tanh	RMSprop	17	480	106	9.00	143.92
msle	tanh	RMSprop	24	460	71	14.00	143.97
mae	relu	Nadam	16	369	192	16.00	144.20

The data obtained were divided into training and test samples (1600 training samples and 401 test samples).The training set was used to find the relationship between input and output variables, while test data were used to evaluate model performance. Data assignment for training and tests ones was performed using a random sample.
An important task in the construction of a neural network is the correct selection of the loss function, optimizer and parameters.
The random search method was used for the selection of parameters [9, 10, 12]. To estimate the reliability of the constructed model, the Root-Mean-Square Error (RMSE) was calculated [13].
For the selection of parameters by the random search method, the following elements were used:
1. Loss function: Logcosh, MAE, MSE, MSLE.
2. Optimizer: SGD, Adam, Adagrad, AdaDelta, RMSprop, Adamax, Nadam.
3. Activation function: ELU, SELU, Softplus, Softsign, ReLu, Tanh, Sigmoid, Hard sigmoid, Linear.
4. The number of neurons in the hidden layer is from 1 to 30.
5. The number of epochs is from 100 to 500.
6. The amount of data in one package is from 50 to 300.
7. Dropout is from 0 to 20.
In total, more than 200 experiments were conducted. The results of the 15 best experiments are presented in Table 3. Fig. 1 shows the calculation error diagram for the test and training samples. The hyperbolic cosine logarithm was chosen as the approximation function, the number of neurons is 16, and the optimizer is Nadam. It can be seen that with an increase in the number of iterations, the error value decreases (the errors for the training and test data were 5.652479786606858e-05 and 7.527783456680481e-05, respectively).

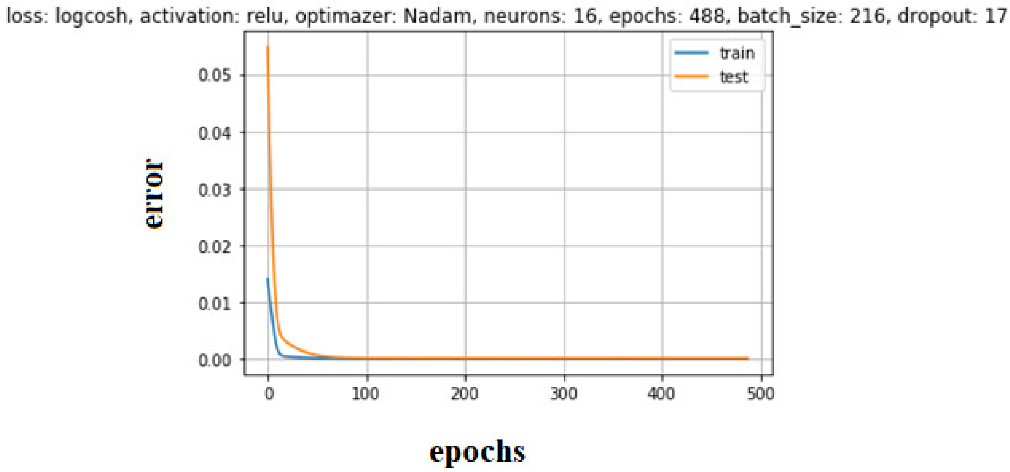

Fig. 1. Calculation error diagram for test and training samples.

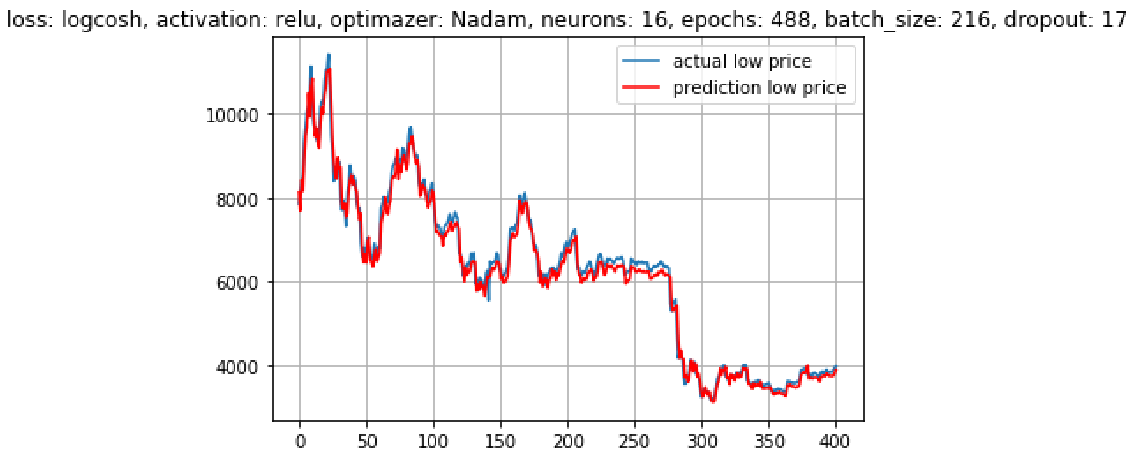

Fig. 2. Charts of the lowest actual and predicted Bitcoin value against the dollar in the period from February 11, 2018, to March 17, 2019.

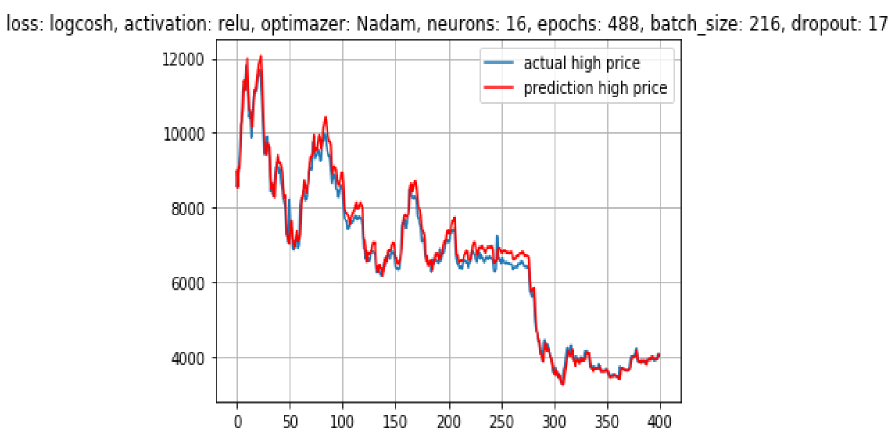

Fig. 3. Charts of the highest actual and predicted Bitcoin value against the dollar in the period from February 11, 2018, to March 17, 2019.

Fig. 2 shows the lowest price charts in the period from February 11, 2018, to March 17, 2019, and the predicted value of the trained model. The hyperbolic cosine logarithm was chosen as the target function, the number of neurons in the hidden LSTM-layer is 16, and the optimizer is Nadam.This chart shows a high correlation between actual and predicted values.

Fig. 3 shows the highest price charts in the period from February 11, 2018, to March 17, 2019, and the predicted value of the trained model. The hyperbolic cosine logarithm was chosen as the target function, the number of neurons in the hidden LSTM-layer is 16, and the optimizer is Nadam. This chart also shows the good prediction results of the resulting model.

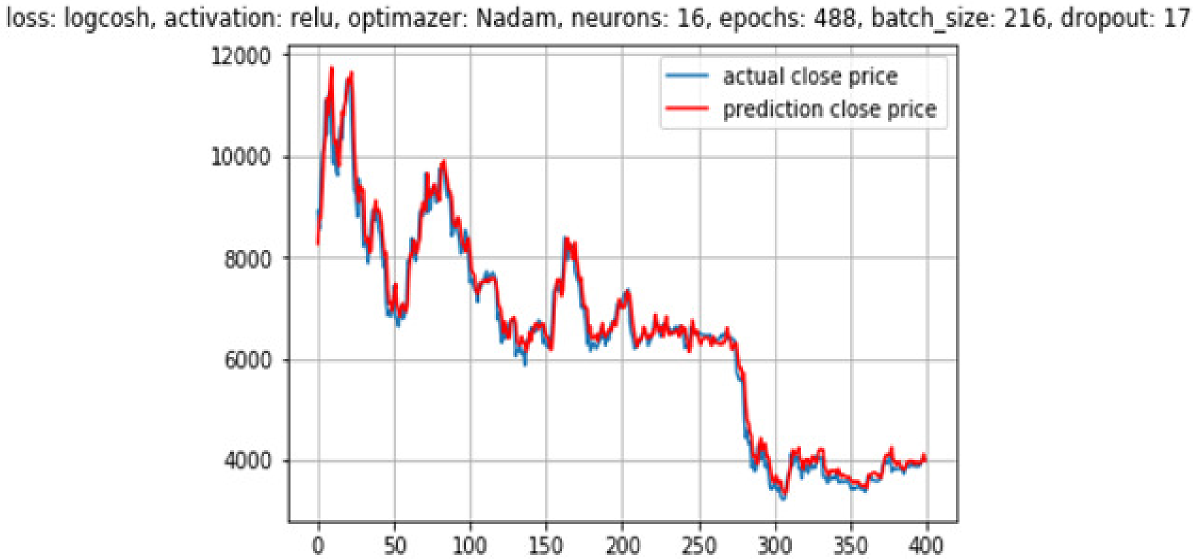

Fig. 4. Closing price charts for the actual and predicted Bitcoin value against the dollar in the period from February 11, 2018, to March 17, 2019.

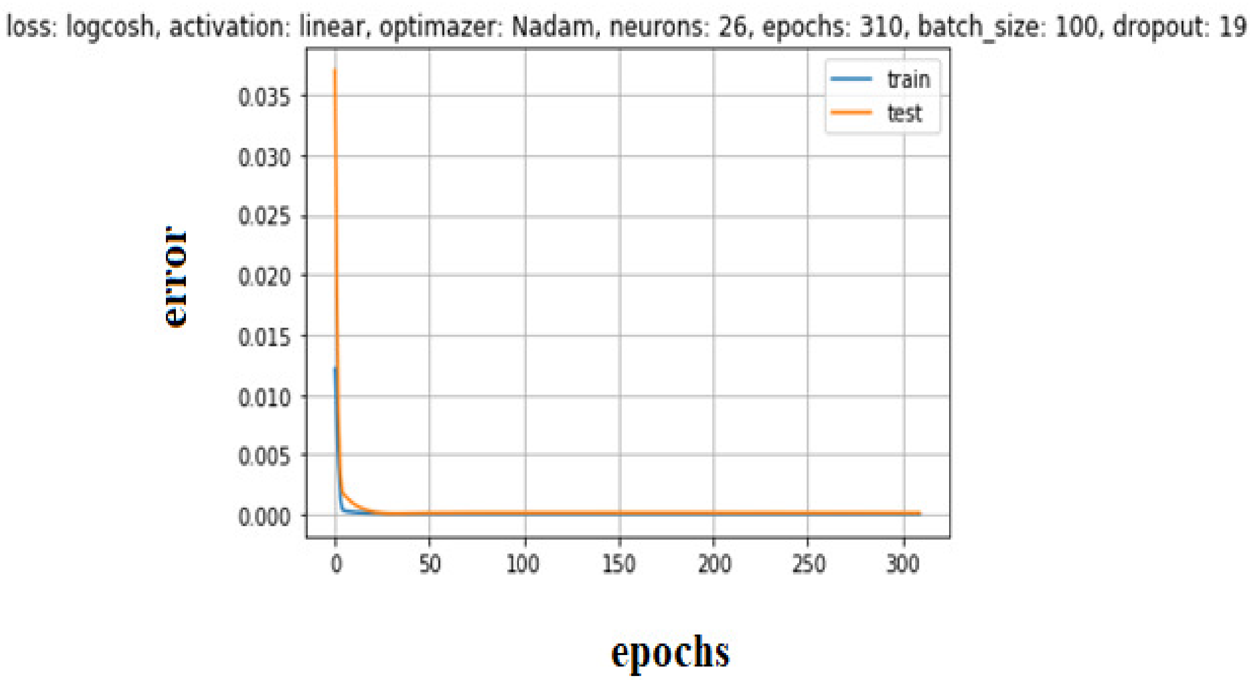

Fig. 5. Calculation error diagram for test and training samples.

Fig. 4 shows the closing price charts in the period from February 11, 2018, to March 17, 2019, and the predicted values in the trained model. The hyperbolic cosine logarithm was chosen as the target function, the number of neurons in the hidden LSTM-layer is 16, and the optimizer is Nadam. This chart also shows the good prediction results of the resulting model.

Fig. 5 shows the computational error diagrams for the test and training samples. The hyperbolic cosine logarithm was chosen as the target function, the number of neurons in the hidden LSTM-layer is 26, and the optimizer is Nadam. It can be seen that with an increase in the number of iterations, the error value decreases (the error for the training and test data is 5.938153853106739e-05 and 7.912181546795182e-05, respectively).

Fig. 6 shows the lowest price charts in the period from February 11, 2018, to March 17, 2019, and the predicted values for the trained model. The hyperbolic cosine logarithm was chosen as the target function, the number of neurons in the hidden LSTM-layer is 26, and the optimizer is Nadam. This chart shows a high correlation between actual and predicted values.

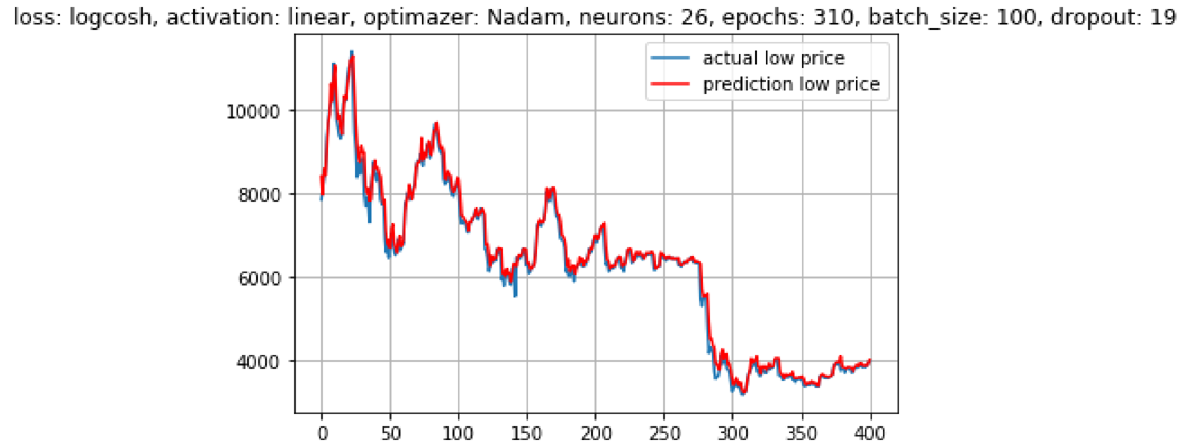

Fig. 6. Price charts of the lowest actual and predicted Bitcoin values against the dollar in the period from February 11, 2018, to March 17, 2019.

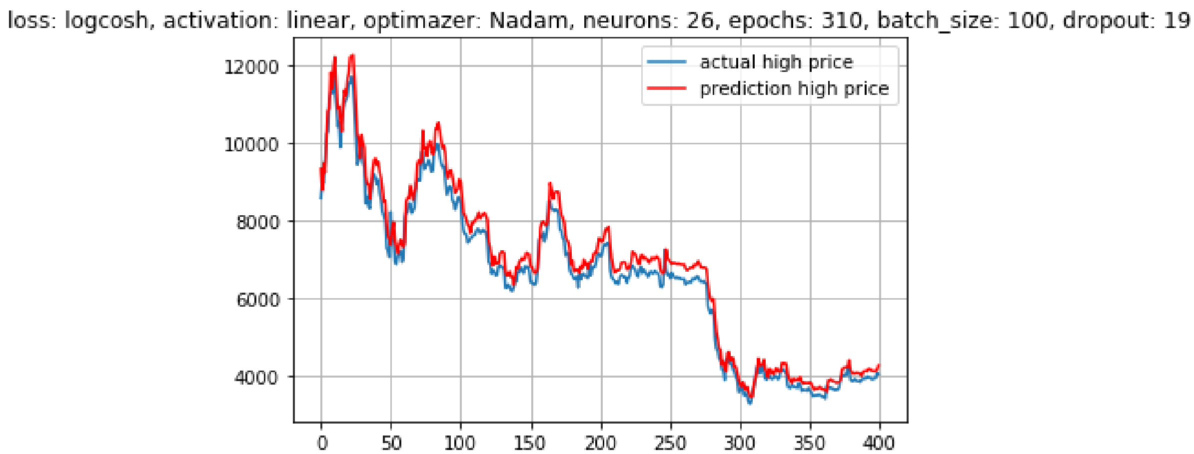

Fig. 7. Charts of the highest actual and predicted Bitcoin value against the dollar in the period from February 11, 2018, to March 17, 2019.

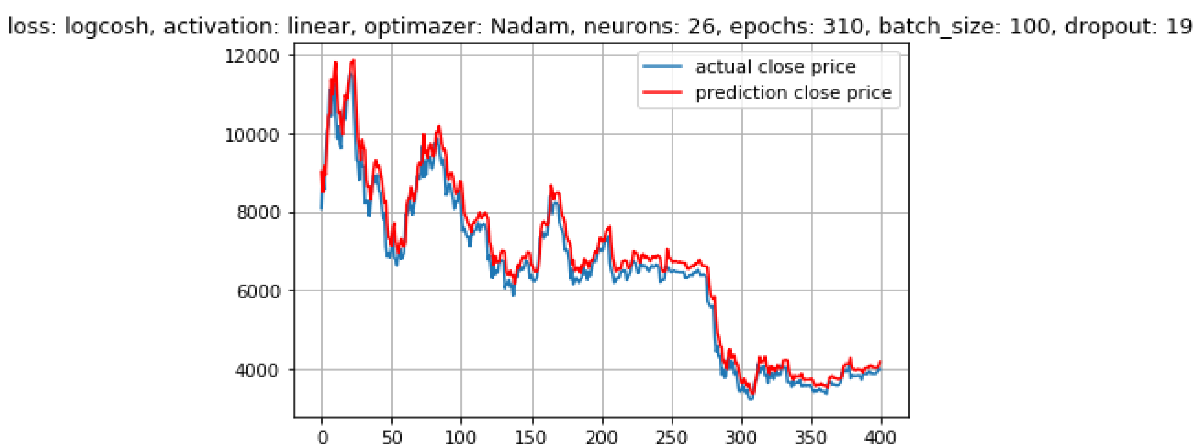

Fig. 8. Closing price charts of the actual and predicted Bitcoin value against the dollar in the period from February 11, 2018, to March 17, 2019.

Fig. 7 shows the highest price charts in the period from February 11, 2018, to March 17, 2019, and the predicted value of the trained model. The hyperbolic cosine logarithm was chosen as the target function, the number of neurons in the hidden LSTM-layer is 26, and the optimizer is Nadam.

Fig. 8 shows the closing price charts in the period from February 11, 2018, to March 17, 2019, and the predicted values of the trained model. The hyperbolic cosine logarithm was chosen as the target function, the number of neurons in the hidden LSTM-layer is 26, and the optimizer is Nadam.

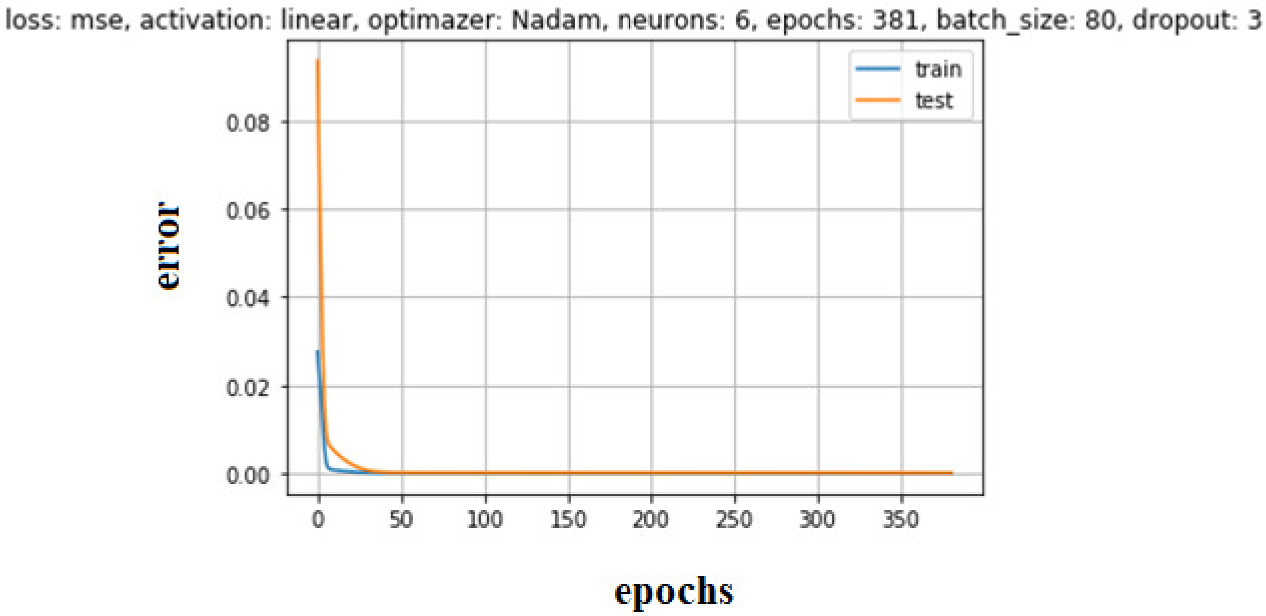

Fig. 9. Calculation error diagram for test and training samples**.**

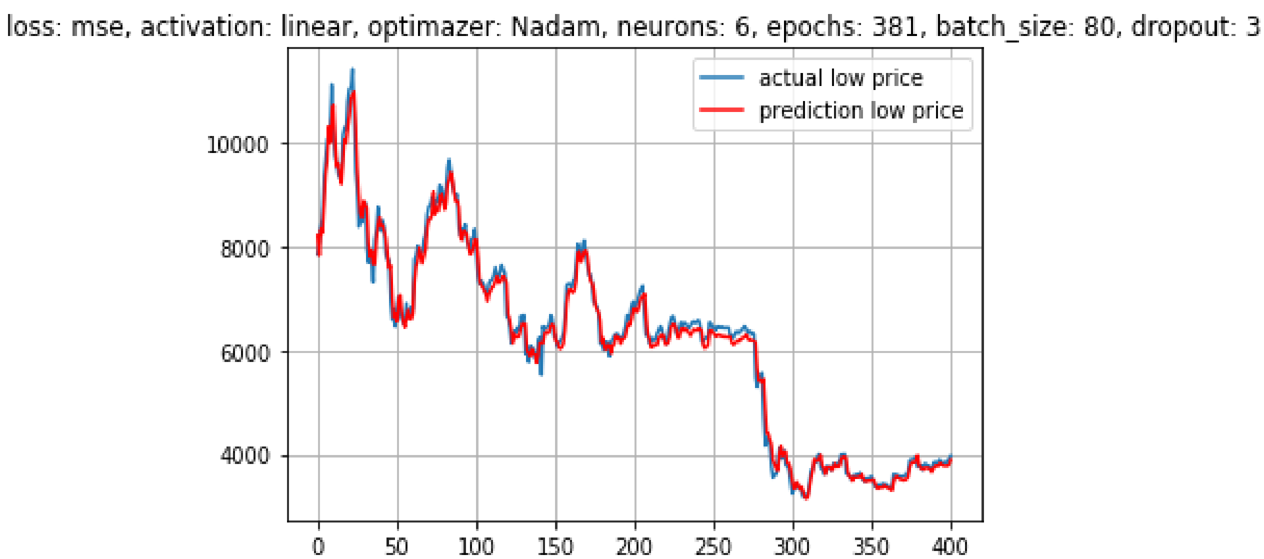

Fig. 10. Charts of the lowest actual and predicted Bitcoin value against the dollar in the period from February 11, 2018, to March 17, 2019.

Fig. 9 shows the calculation error diagram for the test and training samples. The root mean square error function was chosen as the target function, the number of neurons in the hidden LSTM layer is 6, and the optimizer is Nadam. It can be seen that with an increase in the number of iterations, the error value decreases (the error for the training and test data is 0.00011214402533390141 and 0.00014266766753280537, respectively).

Fig. 10 shows the lowest price charts in the period from February 11, 2018, to March 17, 2019, and the predicted values of the trained model. The root mean square error was chosen as the target function, the number of neurons in the hidden LSTM layer is 6, and the optimizer is Nadam. This chart shows a high correlation between actual and predicted values.

Fig. 11 shows the highest price charts in the period from February 11, 2018, to March 17, 2019, and the predicted values of the trained model. The root mean square error function was chosen as the target function, the number of neurons in the hidden LSTM layer is 6, and the optimizer is Nadam.

Fig. 12 shows the closing price charts in the period from February 11, 2018, to March 17, 2019, and the predicted values of the trained model. The root mean square error function was chosen as the target function, the number of neurons in the hidden LSTM layer is 6, and the optimizer is Nadam.

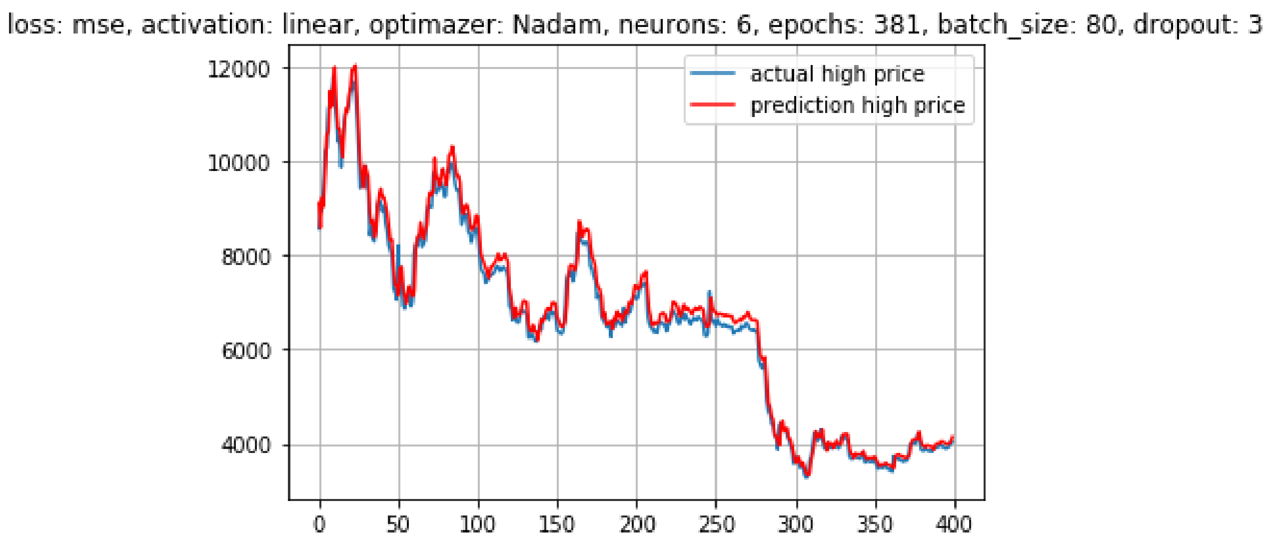

Fig. 11. Charts of the highest actual and predicted Bitcoin values against the dollar in the period from February 11, 2018, to March 17, 2019.

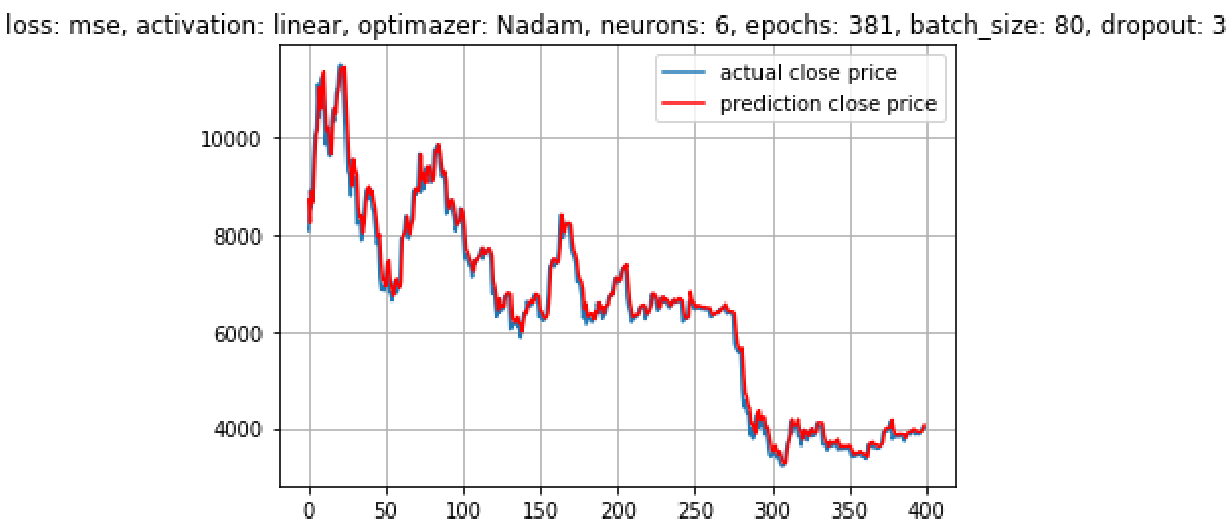

Fig. 12. Closing price charts of the actual and predicted Bitcoin value against the dollar in the period from February 11, 2018, to March 17, 2019.

IV. SUMMARY

We can draw the following conclusions from the experiments:
- the constructed recurrent neural network is well suited for predicting the price in the BTC/USD currency pair (this follows from the "RMSE" column of Table 1 and the charts of the predicted and actual value for the BTC/USD currency pair);
- error levels in the training and validation samples show that the model predicts actual data well.

V. CONCLUSIONS

The proposed method coped well with the task.
The author has considered the task of predicting the exchange rate for the BTC/USD currency pair with a daily time frame. A recurrent neural model with one hidden LSTM-layer was constructed, and parameters were selected using a random search method.
The research results are presented in tabular and graphical form.

ACKNOWLEDGEMENTS

The work is performed according to the Russian Government Program of Competitive Growth of Kazan Federal University.
The publication was prepared within the framework of the scientific project No. 18-010-00536A supported by the Russian Foundation for Basic Research.

REFERENCES

[1]. Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory, Munchen, Germany.
[2]. Chen, G. (2018). A Gentle Tutorial of the Recurrent Neural Network with Error Back Propagation. Department of Computer Science and Engineering, *SUNY at Buffalo*.

[3]. Arkhangelskaya, E.V. Kadurin, A., Nikolenko, S. I. Deep Learning. (2018). Immersion in the world of neural networks, *Publishing House Peter*.
[4]. Strizhov, V.V. (2013). Error function in regression estimations. *Factory laboratory*.
[5]. Ramchoun, H., Idrissi, M. A., Ghanou, Y., & Ettaouil, M. Multilayer Perceptron: Architecture Optimization and Training. *International Journal of Interactive Multimedia and Artificial Intelligence*, Vol. *4*, No. 1. Modelling and Scientific Computing Laboratory, Faculty of Science and Technology, University Sidi Mohammed Ben Abdellah, Fez, Morocco.
[6]. Tormozov, V.S. (2016). Investigation of Neural Memory Networks, Pacific State University, Khabarovsk.
[7]. Hinton, G.E., Rumelhart, D.E., & Williams, R.J. (1985). Learning Internal Representations by Backpropagating error, Parallel Distributes Processing: Explorations in the Microstructure of Cognition
[8]. Golenkov, V.V., & Gerget, O.M. (2013). Analysis of methods for normalizing different data types for expert systems of medical diagnostics. *Tomsk Polytechnic University, Department of Applied Mathematics*.
[9]. Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization, Departement d'Informatique et de rechercheop´erationnelle´. Universite de Montr´eal ´ Montreal, QC, H3C 3J7, Canada.
[10]. Kusherbayeva, V. T., & Sushkov, Yu, A. (2007). Statistical study of the random search algorithm, St. Petersburg State University.
[11]. Criptocompare [Digital source]. - URL: https://www.cryptocompare.com/
[12]. Ngirwa, C. C., & Ally, M. (2018). An ICT Based Solution for Pesticides Authenticity Verification: A Case of Tanzania. *Journal of Information Systems Engineering & Management, 3*(4), 27.
[13]. Vahidi, H., & Mohi Aldin Qhomsheie, M. (2015). Comparison accurate calcula numerical solution of buckling thin cylindrical shape memory alloy shell under uniform load with D.Q.M. *UCT Journal of Research in Science, Engineering and Technology, 3*(1), pp 33-38.