Investigation of Artificial Intelligence Methods in the Short-Term and Middle-Term Forecasting in Financial Sphere

Yuriy Zaychenko, Helen Zaichenko and Oleksii Kuzmenko

Institute for Applied System Analysis, Igor Sikorsky Kyiv Polytechnic Institute, Peremogy avenue 37, Kyiv, 03056, Ukraine

Abstract

In this paper the problems of short- and middle-term forecasting at the financial sphere are considered. For this problem intelligent forecasting methods: LSTM and hybrid deep learning networks based on GMDH are suggested. The optimal parameters of LSTM and hybrid networks were found. Optimal structures of hybrid networks were constructed for short-term and middle-term forecasting. The experimental investigations of LSTM and hybrid DL networks were carried out and their accuracy was compared. The fields of preferable application of LSTM and hybrid DL networks in forecasting problems in finance are determined.

Keywords [1]

Short-term, middle-term financial forecasting, LSTM, hybrid DL network, optimization

1. Introduction. Analysis of previous works

Problems of forecasting share prices and market indexes at stock exchanges pay great attention of investors and various money funds. For their solution were developed and investigated powerful intelligent methods and technologies among them neural networks and fuzzy logic systems.

The efficient tools of modelling and short- and middle-term forecasting of non- stationary time series are LSTM networks. They were developed and successfully applied for forecasting at stock exchanges for long time [1-5]. As alternative approach for forecasting in finance is application of various types of neural network: MLP [6], fuzzy neural networks [7,8], neo-fuzzy networks [9] and Deep learning (DL) networks [10]. New trend in sphere DL networks is new class of NN networks – hybrid DL networks based on GMDH method [11]. The application self-organization in these networks enables to train not only neuron weights but to construct optimal structure of a network. Due to method of training in these networks weights are adjusted not simultaneously but layer after layer. That prevents the phenomenon of vanishing or explosion of gradient. It’s very important for networks with many layers.

The first works in this field used as nodes of hybrid network Wang-Mendel neurons with two inputs [11]. But drawback of such neurons is the demand to train not only neural weights but the parameters of fuzzy sets in antecendents of rules as well. That needs lot of calculational expenses and large training time as well. Therefore, later DL neo-fuzzy networks were developed in which as nodes are used neo-fuzzy neurons by Yamakawa [12,13]. The main property of such neurons is that it’s necessary to train only neuron weights not fuzzy sets. That demands less computations as compared with Wang-Mendel neurons and significantly cuts training time as a whole. The investigation of both classes of hybrid DL networks were performed and their efficiency at forecasting in financial sphere was compared in [13]. Therefore, it presents a great interest to compare the efficiency of hybrid DL networks and LSTM at the problem of forecasting at financial sphere. The goal of this paper is to investigate the accuracy of hybrid DL networks and LSTM at the problem of forecasting market indices at the stock exchange, compare their efficiency at the different intervals and to determine the

Information Technology and Implementation (IT&I-2022), November 30 – December 02, 2022, Kyiv, Ukraine
EMAIL: zaychenkoyuri@ukr.net (Yu. Zaychenko); syncmaster@bigmir.net (H. Zaichenko); oleksii.kuzmenko@ukr.net (O. Kuzmenko)
ORCID: 0000-0001-9662-3269 (Yu. Zaychenko); 0000-0002-4630-5155 (H. Zaichenko); 0000-0003-1581-6224 (O. Kuzmenko)

classes of forecasting problems for which the application of corresponding computational intelligence technologies is the most perspective.

2. The description of the evolving hybrid GMDH-neo-fuzzy network

The evolving hybrid DL-network architecture is presented in Fig.1. To the system's input layer a $(n \times 1)$-dimensional vector of input signals is fed. After that this signal is transferred to the first hidden layer. This layer contains nodes, and each of these neurons has only two inputs.

At the outputs of the first hidden layer the output signals are formed. Then these signals are fed to the selection block of the first hidden layer.

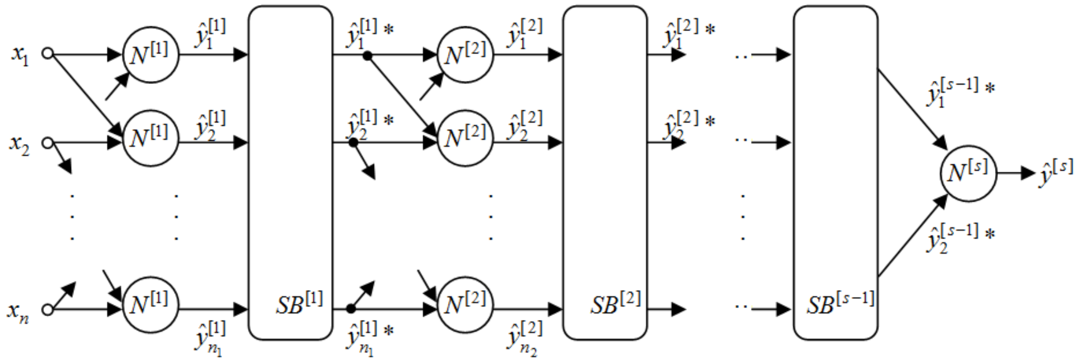

Figure 1: Evolving GMDH-network

It selects among the output signals $\hat{y}_l^{[1]} \cdot n_1 *$ ($n_1 *= F$ is so called freedom of choice) most precise signals by some chosen criterion (mostly by the mean squared error $\sigma^2_{y_l^{[1]}}$). Among these $n_1 *$ best outputs of the first hidden layer $\hat{y}_l^{[1]} *\cdot n_2$ pairwise combinations $\hat{y}_l^{[1]} *, \hat{y}_p^{[1]} *$ are formed. These signals are fed to the second hidden layer, that is formed by neurons $N^{[2]}$. After training these neurons output signals of this layer $\hat{y}_l^{[2]}$ are transferred to the selection block $SB^{[2]}$ which choses F best neurons by accuracy (e.g. by the value of $\sigma^2_{y_l^{[2]}}$) if the best signal of the second layer is better than the best signal of the first hidden layer $\hat{y}_1^{[1]} *$. Other hidden layers forms signals similarly. The system evolution process continues until the best signal of the selection block $SB^{[s+1]}$ appears to be worse than the best signal of the previous s th layer. Then we return to the previous layer and choose its best node neuron $N^{[s]}$ with output signal $\hat{y}^{[s]}$. And moving from this neuron (node) along its connections backwards and sequentially passing all previous layers we finally get the structure of the GMDH-neo-fuzzy network.

It should be noted that in such a way not only the optimal structure of the network may be constructed but also well-trained network due to the GMDH algorithm. Besides, since the training is performed sequentially layer by layer the problems of high dimensionality as well as vanishing or exploding gradient are avoided.

2.1. Neo-fuzzy neuron as a node of hybrid GMDH-system

Let's introduce into consideration the architecture of the node that is presented in Fig.2 and is suggested as a neuron of the proposed GMDH-system. As a node of this structure a neo-fuzzy neuron (NFN) by Takeshi Yamakawa and co-authors in [9] is used. The neo-fuzzy neuron is a nonlinear multi-input single-output system shown in Fig.2. The main difference of this node from the general neo-fuzzy neuron structure is that each node uses only two inputs. It realizes the following mapping:

$$\hat{y} = \sum_{i=1}^{2} f_i(x_i) \quad (1)$$

where x_i is the input i $(i = 1,2,\dots,n)$, $\hat{y}$ is a system output. Structural blocks of neo-fuzzy neuron are nonlinear synapses NS_i which perform transformation of input signal in the form

$$f_i(x_i) = \sum_{j=1}^{n} w_{ji}\mu_{ji}(x_i) \tag{2}$$

and realize fuzzy inference: if x_i is x_{ji} then the output is w_{ji},where x_{ji} is a fuzzy set which membership function is μ_{ji}, w_{ji} is a synaptic weight in consequent [11].

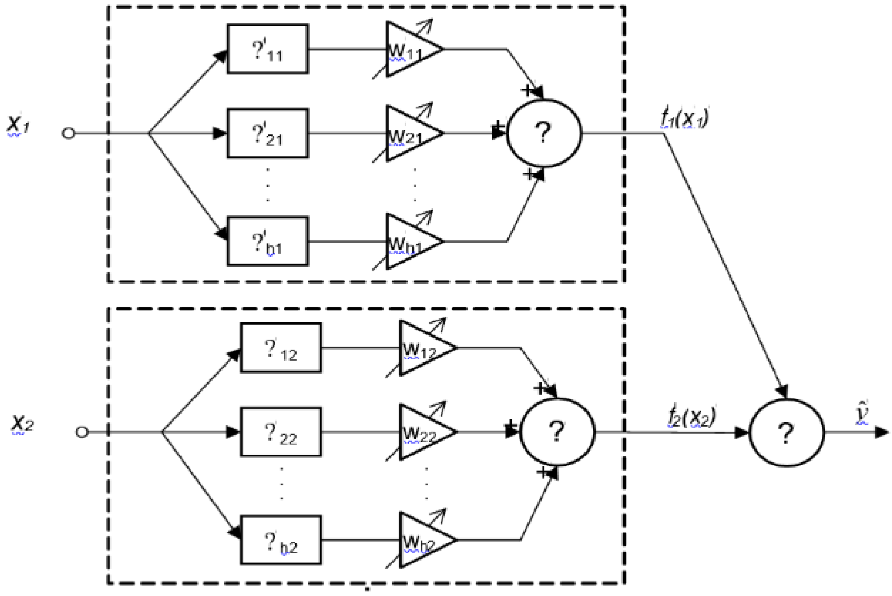

Figure 2: Architecture of neo-fuzzy neuron with two inputs

2.2. The neo-fuzzy neuron learning algorithm

The learning criterion (goal function) is the standard local quadratic error function:

$$E(k) = \frac{1}{2}\big(y(k) - \hat{y}(k)\big)^2 = \frac{1}{2}e(k)^2 = \frac{1}{2}\left(y(k) - \sum_{i=1}^{2}\sum_{j=1}^{h} w_{ji}\mu_{ji}(x_i(k))\right)^2 \tag{3}$$

It is minimized via the conventional stochastic gradient descent algorithm. In case we have priori defined data set the training process can be performed in a batch mode for one epoch using conventional least squares method [12]

$$w^{[1]}(N) = \left(\sum_{k=1}^{N}\mu^{[1]}(k)\mu^{[1]T}(k)\right)^{+}\sum_{k=1}^{N}\mu^{[1]}(k)y(k) = P^{[1]}(N)\sum_{k=1}^{N}\mu^{[1]}(k)y(k) \tag{4}$$

where $(\bullet)^{+}$ means pseudo inverse of Moore-Penrose (here $y(k)$ denotes external reference signal (real value).

If training observations are fed sequentially in on-line mode, the recurrent form of the LSM can be used in the form [11,12]

$$\begin{cases} w_l^{ij}(k) = w_l^{ij}(k-1) + \dfrac{P^{ij}(k-1)\left(y(k) - \left(w_l^{ij}(k-1)\right)^T\varphi^{ij}\big(x(k)\big)\right)\varphi^{ij}\big(x(k)\big)}{1 + \left(\varphi^{ij}\big(x(k)\big)\right)^T P^{ij}(k-1)\varphi^{ij}\big(x(k)\big)}, \\ P^{ij}(k) = P^{ij}(k-1) - \dfrac{P^{ij}(k-1)\varphi^{ij}\big(x(k)\big)\left(\varphi^{ij}\big(x(k)\big)\right)^T P^{ij}(k-1)}{1 + \left(\varphi^{ij}\big(x(k)\big)\right)^T P^{ij}(k-1)\varphi^{ij}\big(x(k)\big)}. \end{cases} \tag{5}$$

3. LSTM model and architecture

Recurrent neural networks (RNNs) are based on the idea of passing through sequence of time steps with derivatives that do not explode or vanish. The idea is that some gated self-loop can be introduced that allows to decide what information should be forgotten, saved, and kept. That decision should be based on features that neural networks learn during training. One of the most successful architectures that implement that idea is Long-Short-Term-Memory (LSTM) recurrent neural network [1-5].

LSTM replaces the regular RNN unit with LSTM block that has its internal memory and can be recurrently connected with other LSTM blocks. LSTM has two types of connections – external connections that are similar to the recurrent connection between RNN hidden units, and internal state $s_i^{(t)}$ that has a recurrent internal connection to itself. In addition, LSTM block has three main gates that control an information flow and decide whether new information should be forgotten or saved and whether we need to keep old information in memory [5]. An example of LSTM architecture is shown in Fig.3.

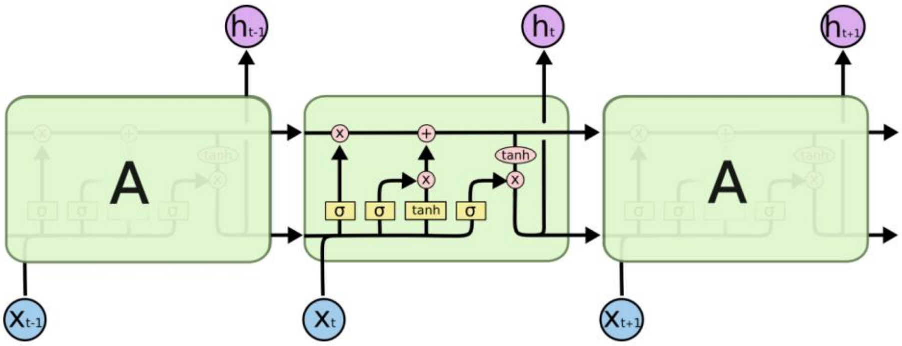

Figure 3: Simple diagram of the LSTM block

A forward pass through a single LSTM block consists of several main steps that are supposed to interact with the internal network state. The first step is a forget gate unit that decides which information should be erased in the internal state. The internal state saves all information from all previous steps. The process of "forgetting" information from previous steps can be expressed with the usage of sigmoid function and weights matrices:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)} \right) \tag{6}$$

where $x^{(t)}$ is a current input vector at time step t; $h^{(t)}$ is a hidden unit vector at the current time step that contains information from LSTM block outputs in the previous time steps; b_i^f is forget gate bias vector; U^f is a matrix of input weights for forget gate; W^f is a matrix of recurrent weights for forget gate.

The next step for LSTM block consists of several intermediate steps. First, the input gate decides which information in the internal state should be updated with new data. Then, the network creates a list of new elements that reflect new information that should be added to the internal state. Finally, the network combines all information from previous steps and updates the internal state $s_i^{(t)}$. All these operations are described with the following equation [4, 5]:

$$s_i^{(i)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j}\, x_j^{(t)} + \sum_j W_{i,j}\, h_j^{(t-1)} \right) \tag{7}$$

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g\, x_j^{(t)} + \sum_j W_{i,j}^g\, h_j^{(t-1)} \right) \tag{8}$$

where b is a bias vector into LSTM block; U – input weights in the LSTM block; W is recurrent weights into LSTM block; $g_i^{(t)}$ is an external input gate function.

The last step of LSTM block decides which information should be returned as output. Output value calculated using output gate mechanism:

$$h_i^{(t)} = tanh\left(s_i^{(t)} q_i^{(t)}\right) \tag{9}$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_i U_{i,j}^o\, x_j^{(t)} + \sum_j W_{i,j}^o\, h_j^{(t-1)}\right) \tag{10}$$

where b^o, U^o, W^o respectively bias vector, input, and recurrent weights matrices of output gate.

For training LSTM stochastic gradient method and its modern modifications are used. LSTM architecture has been successful on real-world tasks in different domains and shows that it works much better with long-term dependencies than poor RNNs [4, 5].

4. Experimental investigation and results analysis

4.1. Data set

As an input data was taken corporate Emerging Markets Bond total Return Index (EMBRI) at NASDAQ stock exchange at the period since January till August 2022. The sample consisted of instances which were divided on training and test subsamples.

Flow chart of EMBRI is presented in the Fig. 4.

Figure 4: Emerging Markets Corporate Bond Total Return Index (EMBRI)

4.2. Experimental investigations of hybrid DL networks

The first series of experiments were performed with hybrid Deep Learning network with neo-fuzzy neurons as nodes. At experiments the following parameters were varied: ratio training/test sample, number of inputs 3-5, number of fuzzy sets per variable 3-5 and membership functions: Bell, Gaussian and Triangular. The goal of experiments was to find optimal parameters values. Forecast period was taken 5 days, as the accuracy metrics were taken MSE and MAPE. In the first experiment Bell MF was explored. After experiment optimal parameters were found for the hybrid DL network: number of inputs – 3, number of fuzzy sets – 3, ratio – 0.8. With these parameters values the best accuracy at the test sample was attained: MSE = 0,424, MAPE = 0,155.

The next experiments were performed for hybrid network with Gaussian MF. After experiments were found optimal parameters of hybrid DL network: number of inputs – 3, number of rules – 4, ratio training/test – 0.6. With these parameters the forecasting results are presented in the Table 1.

In the Table 2 the process of structure generation of the best network is presented with optimal parameters. As it follows from Table 2 optimal structure consists of 3 layers: 3 nodes at the first layer, 3 nodes at the second layer and 1 node at the third layer. Flow chart of forecasting with this network structure is prese at the Fig. 5.

Table 1

The best forecast with Gaussian MF (inputs: 3; rules: 4; ratio: 0.6)

Date	Real	Forecast	MSE	MAPE
17.08.2022	399,15	398,44	0,504	0,178
18.08.2022	399,12	398,17	0,902	0,238
19.08.2022	397,97	398,50	0,281	0,133
22.08.2022	396,82	397,40	0,336	0,146
23.08.2022	396,62	396,93	0,096	0,078
		Min:	0,096	0,078
		Avg:	0,424	0,155
		Max:	0,902	0,238

Table 2

The process of structure generation for the best result with Gaussian MF (inputs: 3; rules: 4; ratio: 0.6)

NFN	SB1	SB2	SB3
(0, 1)	6,458466		
(0, 2)	2,746775		
(1, 2)	4,979467		
((0, 1), (0, 2))		0,020353	
((0, 1), (1, 2))		0,003018	
((0, 2), (1, 2))		0,000982	
(((0, 1), (0, 2)), ((0, 1), (1, 2)))			0,028025
(((0, 1), (0, 2)), ((0, 2), (1, 2)))			0,045476
(((0, 1), (1, 2)), ((0, 2), (1, 2)))			0,049383

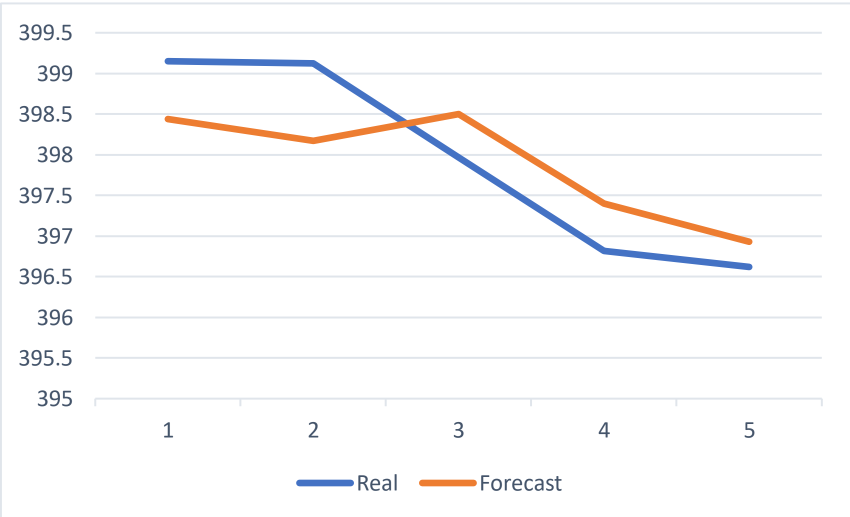

Figure 5: The best forecast with Gaussian MF (inputs: 3; rules: 4; ratio: 0.6)

The next experiment was carried out with hybrid DL network with triangular MF. After experiment were found optimal parameters and structure of the hybrid network: 5 inputs, 3 MF and 0.8 ratio. After experiments the accuracy of hybrid networks with different MF was compared. The results are presented in the Fig. 6. At the next series experiments LSTM networks were investigated. The goal of experiments was to find the optimal parameters. The following parameters varied:

number of inputs 3-5, ratio training/test 0.6, 07, 0.8. The optimal parameters values were found: n=3, ratio 0.7. After that the LSTM with optimal parameters was applied for training and forecasting. The results are presented in the Table 3. Flow charts of training and validation are presented in the Fig. 7.

In the next experiments the efficiency of the best models of hybrid DL network and LSTM was investigated and compared with different ratios training/test sample. The corresponding results for different in puts are presented in the Tables 4-6. Analyzing these results, we may conclude GMDH-neo-fuzzy network has better forecasting accuracy at the interval period 5 than LSTM for various ratios. In the next experiments the forecasting efficiency of hybrid DL networks and LSTM at different forecasting intervals were investigated. In the Table 7 forecasting results of GMDH-neo-fuzzy network and LSTM at interval 7 days and in the Fig. 8 forecasting accuracy are presented.

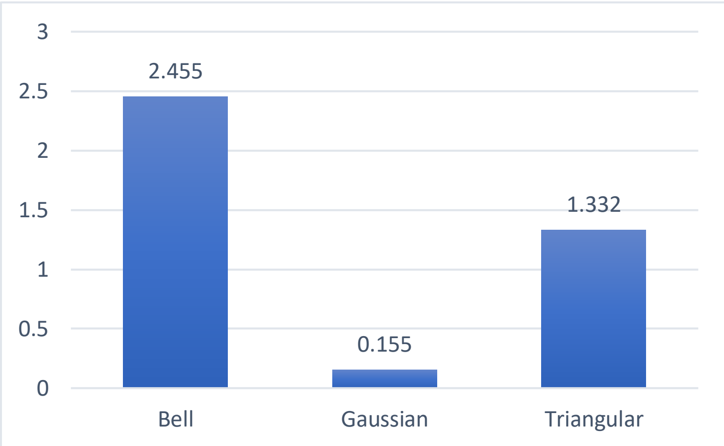

Figure 6: Comparison of the best MAPE (avg.) values for different MF

Table 3
The best forecast of LSTM (inputs=3; ratio=0.7)

Date	Real	Forecast	MSE	MAPE
17.08.2022	399,15	397,08	4,285	0,519
18.08.2022	399,12	397,58	2,372	0,386
19.08.2022	397,97	397,52	0,203	0,113
22.08.2022	396,82	396,71	0,012	0,028
23.08.2022	396,62	396,48	0,019	0,035
		Min:	0,012	0,028
		Avg:	1,378	0,216
		Max:	4,285	0,519

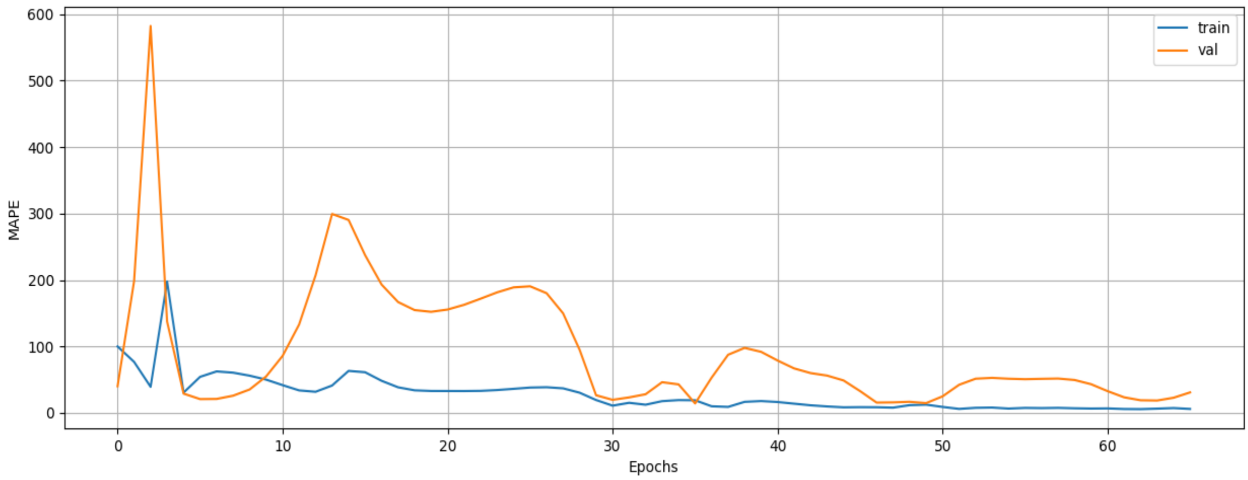

Figure 7: Training an LSTM network (MAPE)

Table 4
MSE for different training/test ratio and 3 inputs

Ratio training/test	GMDH-neo-fuzzy	LSTM
60/40	0,424	4,074
70/30	1,242	1,378
80/20	1,248	2,023

Table 5
MSE for different training/test ratio and 4 inputs

Ratio training/test	GMDH-neo-fuzzy	LSTM
60/40	2,894	4,715
70/30	1,783	2,107
80/20	3,491	5,412

Table 6
MSE for different training/test ratio and 5 inputs

Ratio training/test	GMDH-neo-fuzzy	LSTM
60/40	5,556	1,961
70/30	7,414	4,628
80/20	5,902	2,152

Table 7
The best forecast for period 7 – MAPE

Date	Real	GMDH-neo-fuzzy	LSTM
15.08.2022	400,32	0,412	1,806
16.08.2022	400,70	0,317	1,899
17.08.2022	399,15	0,704	1,518
18.08.2022	399,12	0,807	1,511
19.08.2022	397,97	0,859	1,224
22.08.2022	396,82	0,935	0,932
23.08.2022	396,62	0,630	0,877

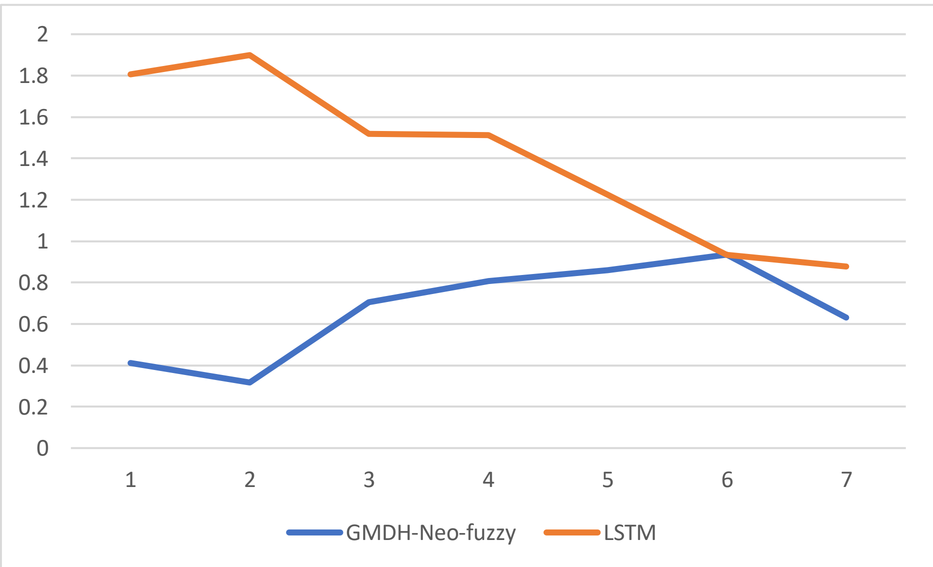

Figure 8: The best forecast for period 7 – MAPE

As it follows from the presented results the hybrid neo-fuzzy network has the better accuracy than LSTM at the interval 7 days. At the succeeding experiments the forecasting accuracy of both networks was explored at middle-term fore casting with interval 20 days. The accuracy by MAPE is presented in the Fig. 9.

4.3. Comparative experiments of hybrid DL networks and LSTM

In the final experiments the forecasting accuracy of hybrid GMDH network (GMDH-nf) and LSTM were compared at different forecasting intervals (short-term and middle-term). The corresponding results by criterion MSE and MAPE are presented in the Table 8 and Table 9 correspondingly where the point means the size of test sample.

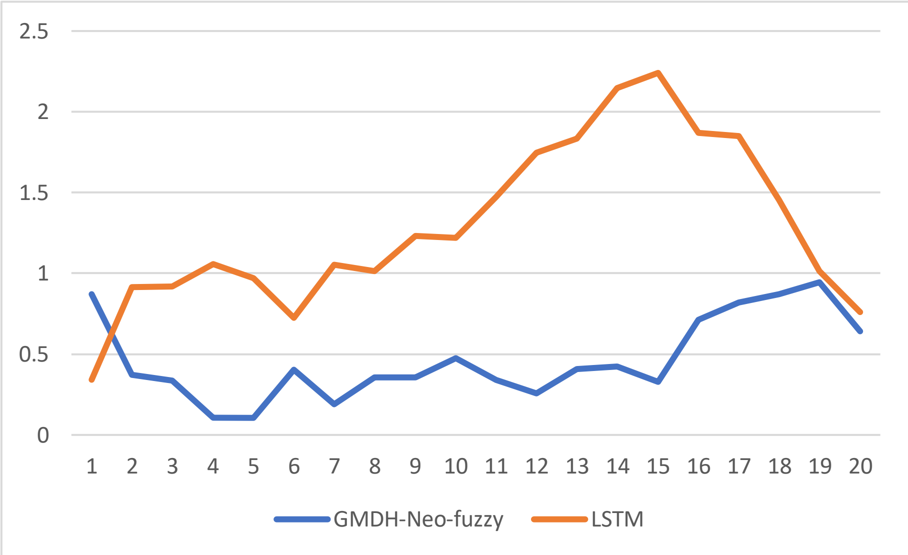

Figure 9: The best forecast for period 20 – MAPE

Table 8
Average MSE (avg.) for different intervals (periods)

		Point 5	Point 7	Point 20
Period 3	GMDH-nf	0,319	0,678	1,251
	LSTM	0,702	0,408	2,374
Period 5	GMDH-nf	1,178	2,819	0,953
	LSTM	2,674	3,054	2,374
Period 7	GMDH-nf	0,238	2,714	0,863
	LSTM	5,888	6,325	2,983
Period 20	GMDH-nf	2,241	1,238	0,595
	LSTM	6,182	4,178	3,134

Table 9
Average MAPE (avg.) for different intervals (periods)

		Point 5	Point 7	Point 20
Period 3	GMDH-nf	0,094	0,192	0,262
	LSTM	0,201	0,127	0,349
Period 5	GMDH-nf	0,111	0,337	0,224
	LSTM	0,316	0,358	0,352
Period 7	GMDH-nf	0,231	0,475	0,203
	LSTM	0,489	0,510	0,364
Period 20	GMDH-nf	0,392	0,368	0,172
	LSTM	0,453	0,408	0,416

Analysis of these results shows that in a whole hybrid DL network has the better accuracy than LSTM at different short and middle forecasting intervals (3, 5, 7, 20 days).

5. Conclusion

1. In this paper the problem of forecasting at financial market with different forecasting intervals was considered (short-term and middle-term forecasting). For its solution it was suggested to apply hybrid deep learning (DL) networks based on GMDH and LSTM networks.
2. The experimental investigations were performed at the problem of forecasting Emerging Markets Bond Total Return Index (EMBRI) at NASDAQ stock exchange at the period since January till August 2022.
3. Optimization of parameters of LSTM and hybrid networks was performed during the experiments. The optimal structure of hybrid DL network was constructed using GMDH method.
4. The experimental investigations of optimized LSTM and hybrid networks were carried out at different forecasting intervals and their accuracy was compared.

In result it was established that application of hybrid DL networks has much better accuracy than LSTM at the problems of short-term and middle-term forecasting at stock exchanges.

6. References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory", Neural Computation, vol. 9, 1997, pp. 1735–1780. doi: 10.1162/neco.1997.9.8.1735.

[2] B. Hammer, "On the approximation capability of recurrent neural networks", Neurocomputing, vol. 31, 1998, pp. 107–123. doi: 10.1016/S0925-2312(99)00174-5.

[3] C. Olah, "Understanding lstm networks", 2020. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[4] A. Graves, "Generating sequences with recurrent neural networks", CoRR, vol. abs/1308.0850, 2013. doi: 10.48550/arXiv.1308.0850.

[5] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, Verlag, Berlin, Heidelberg: Springer, 2012. doi: 10.1007/978-3-642-24797-2.

[6] S.S. Haykin, Neural networks: a comprehensive foundation, 2nd ed. Upper Saddle River, N.J: Prentice Hall, 1999.

[7] Ossovsky S. Neural networks for information processing, transl. from Polish.–M. – Finance and Statistics, 2002 – 344p. (rus).

[8] Wang F. Neural Networks Genetic Algorithms and Fuzzy Logic for Forecasting // Proc. Intern. Conf. Advanced Trading Technologies. – New York, 1992, pp. 504–532.

[9] Yamakawa T., Uchino E., Miki T., Kusanagi H. A neo-fuzzy neuron and its applications to system identification and prediction of the system behavior // Proc. 2nd Intern. Conf. Fuzzy Logic and Neural Networks «LIZUKA-92». – Lizuka, 1992, pp. 477–483.

[10] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, MIT PRESS, 2016. URL: http://www.deeplearningbook.org.

[11] Yuriy Zaychenko, Yevgeniy Bodyanskiy, Oleksii Tyshchenko, Olena Boiko, Galib Hamidov. Hybrid GMDH-neuro-fuzzy system and its training scheme. Int. Journal Information theories and Applications, 2018. vol.24, Number 2, pp. 156–172.

[12] Yu. Zaychenko, Galib Hamidov. The Hybrid Deep Learning GMDH-neo-fuzzy Neural Network and Its Applications. Proceedings of 13-th IEEE Intern Conference Application of Information and Communication Technologies-AICT2019. 23–25 October 2019, Baku, pp. 72–77.

[13] Evgeniy Bodyanskiy, Yuriy Zaychenko, Olena Boiko, Galib Hamidov, Anna Zelikman. Structure Optimization and Investigations of Hybrid GMDH-Neo-fuzzy Neural Networks in Forecasting Problems. System Analysis & Intelligent Computing. Ed. Michael Zgurovsky, Natalia Pankratova. Book Studfies in Computational Intelligence, SCI, vol.1022. Springer, 2022, pp. 209–228.