# INVESTIGATION OF THE EFFECTIVENESS OF ARTIFICIAL NEURAL NETWORKS OF DIFFERENT GENERATIONS IN THE TASK OF FORECASTING IN THE FINANCIAL SPHERE

#### Ye. BODYANSKIY, Yu. ZAYCHENKO, He. ZAICHENKO, O. KUZMENKO

Abstract. This paper discusses ANNs of different generations. The efficiency of using computational intelligence in the task of short- and medium-term forecasting in the financial sphere is investigated. For the investigation, a fully connected feed-forward network (Back Propagation), a recurrent network (LSTM), a hybrid deep learning network based on self-organization (GMDH neo fuzzy), and a hybrid system of computational intelligence based on bagging and group method of data handling (HSCI bagging) were chosen. The experimental parameters chosen are the prediction interval, the number of inputs, the percentage of validation data in the training set, and the number of fuzzifiers (for GMDH neo-fuzzy). Experiments were conducted, and the best results for different prediction intervals were compared. The optimal parameters of the networks and the feasibility of their use in the task of forecasting at different intervals are determined.

Keywords: generations of ANNs, Back Propagation, LSTM, GMDH neo fuzzy, HSCI bagging.

## INTRODUCTION

Generations of artificial neural networks represent different stages in the evolution of artificial intelligence (AI) and machine learning technologies. Each generation introduces new approaches, architectures, and improvements that make neural networks more powerful and efficient.

The first generation of artificial neural networks encompasses early developments in artificial intelligence and machine learning. The basic model of this generation is the perceptron, developed in 1957 by Frank Rosenblatt [1]. It was the simplest type of artificial neural network and consisted of three main components. The S-System (Sensory System) was represented by a set of points in a TV raster, or a set of photocells. The A-System (Association System) performed the switching functions between input and output. The R-System (Response System) consisted of a typically of a relatively small number of units. Such a model allowed solving only linearly separable problems, i.e. problems where a straight line can be drawn to separate data classes. However, it was not possible to solve more complex problems, such as the XOR-problem, where classes cannot be separated by a straight line. Although the perceptron was an important step forward, its capabilities were limited. After researchers realized that it could not solve nonlinear problems, the development of neural networks slowed down for a while, but the first generation of neural networks laid the foundation for future research. It demonstrated the ability of machines to learn from experience, albeit with limited capabilities. This led to the further development of more complex models in the following generations.

© Publisher IASA at the Igor Sikorsky Kyiv Polytechnic Institute, 2025 124 ISSN 1681–6048 System Research & Information Technologies, 2025, № 1

The second generation of artificial neural networks has advanced significantly compared to the first, introducing multi-layer architectures and advanced learning methods such as the back-propagation algorithm. This generation opened up new possibilities for solving more complex problems that could not be solved by simple first-generation models. The main feature of this generation was the introduction of Multilayer Perceptrons (MLPs). Such networks consisted of several layers of neurons: an input layer, one or more hidden layers, and an output layer. The second generation will include the Back Propagation neural network, which was proposed by Rumelhart, Hinton, and Williams in 1986 [2]. The authors of this paper first showed how to train such a network with an arbitrary number of layers and proposed a recurrent gradient-type algorithm for training it for a network with an arbitrary structure. This network belongs to the class of feedforward neural networks. The Back Propagation Network has been widely used in numerous tasks of function approximation, forecasting, and pattern recognition. Its versatility is defined by the Universal Approximation Theorem [3]. With its multilayer architecture and backpropagation algorithm, the second generation of neural networks was able to solve problems that cannot be separated linearly, such as the XOR-problem. This significantly expanded the scope of neural networks.

The third generation of artificial neural networks was marked by the emergence of Deep Learning [4], an approach that has led to significant breakthroughs in many areas of artificial intelligence. Deep neural networks consist of a large number of layers (deep architecture), which allows modeling more complex and abstract data representations. Deep Learning requires large amounts of data for training and powerful computing resources, in particular graphics processing units (GPUs). This has become possible due to the development of data storage and processing technologies, as well as improvements in hardware. It is worth noting that Jurgen Schmidhuber refers to the Group Method of Data Handling (GMDH) [5; 6] as the earliest deep learning method, noting that it was used to train a neural network consisting of eight layers back in 1971 [7]. This method was proposed in the late 60s and early 70s by acad. A.G. Ivakhnenko and his colleagues. This method is based on the selective selection of models on the basis of which more complex models are built. The modeling accuracy at each subsequent step increases due to the model's complexity. Solving complex problems has led to the emergence of new types of neural networks. A special type of deep neural network, convolutional neural networks (CNNs) [8], has been developed for working with images. They use special layers (convolutional layers) that can detect various image features, such as contours, textures, and objects, at different levels of abstraction. Recurrent neural networks (RNNs) [9-11] are designed to work with sequences of data, such as text or audio. They store information about previous processing steps, which allows them to consider context in natural language processing and speech recognition tasks. In addition to error backpropagation, the third generation uses advanced optimization, regularization, and normalization techniques to train deep networks more efficiently, reducing the likelihood of overtraining. The third generation of neural networks has become a real breakthrough in the field of artificial intelligence. This generation, artificial neural networks have become a key tool in the development of intelligent systems that are now used in everyday life.

The fourth generation of artificial neural networks is characterized by the emergence of new architectures and methods that have further expanded the capabilities of artificial intelligence. The main feature of this generation is the ability to perform several different tasks using a single architecture. Quite often, new approaches to training are used, such as pre-training methods on large data sets followed by fine-tuning on specific tasks, which allows creating models that can easily adapt to different contexts and tasks. This hybrid approach to building neural networks allows them to be used in such industries as medical diagnostics. virtual assistants. business process automation, autonomous vehicles. personalized advertising, and much more. Their versatility and adaptability are also achieved through self-organization. The fourth generation is represented by self-organizing deep learning networks [12-15]. The key feature of this type of network is that it builds its structure in the process of learning. Transformers have become an important innovation of the fourth generation. Transformers use a selfattention mechanism that allows the model to efficiently process sequential data, such as text, and consider the dependencies between sequence elements regardless of their location. Large Language Models (LLMs) are one of the most striking achievements of the fourth generation, such as GPT (Generative Pre-trained Transformer), BERT (Bidirectional Encoder Representations from Transformers), and others. These models are capable of generating, analyzing, and understanding text at a level that was previously considered unattainable. They can perform translation, text generation, question answering, and other tasks. In addition to language models, the fourth generation includes powerful generative models such as DALL-E, Stable Diffusion, and others. These models can create images based on textual descriptions, opening up new possibilities in creativity, design, and many other industries. The fourth generation of artificial neural networks has significantly expanded the boundaries of what is possible in the field of artificial intelligence. The use of transformers and large language models has allowed for new heights in understanding and generating natural language, which has become the foundation for many innovations in various industries. This generation has also emphasized the importance of multitasking and versatility, as models have become capable of performing a variety of tasks using a single architecture. Such advances continue to change the technological landscape and drive the further development of artificial intelligence, making it even more powerful and useful in various aspects of life.

What will the next generation of artificial neural networks look like? What opportunities will we have? What tasks will we be able to solve? To answer these questions, it is worth taking a closer look at the evolution of the structure of artificial neural network training algorithms and practically exploring their capabilities.

### **EVOLUTION OF ANNS**

The perceptron [1] is the basis for more complex neural networks, where hidden layers are added to solve nonlinear problems and work with more complex data. The Back Propagation neural network has a multilayer fully connected architecture. During training, neuron weights are adjusted to reduce the error between the predicted result and the actual value [2]. This process ensures efficient training of the neural network based on a large amount of data [3].

The third-generation LSTM neural network is more complex (Fig. 1).



Fig. 1. LSTM recurrent network architecture

A forward pass through a single LSTM block consists of several main steps that are supposed to interact with the internal network state [4; 10; 11]. 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:

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

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:

$$s_i^{(i)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma \left( b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)} \right),$$
(2)

$$g_i^{(t)} = \sigma \left( b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)} \right),$$
(3)

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:

$$h_i^{(t)} = \tanh(s_i^{(t)}q_i^{(t)}),$$
 (4)

$$q_i^{(t)} = \sigma \left( b_i^o + \sum_i U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)} \right),$$
(5)

where  $b^o$ ,  $U^o$ ,  $W^o$  is 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.

A hybrid deep learning network based on self-organization — GMDH-neofuzzy – is a fourth-generation network (Fig. 2) [12; 13].



Fig. 2. The process of synthesis of the GMDH-neo-fuzzy network structure

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  $n_1 = c_n^2$  nodes, and each of these neurons has only two inputs.

At the outputs  $N^{[1]}$  of the first hidden layer the output signals are formed. Then these signals are fed to the selection block of the first hidden layer.

It selects among the output signals  $\hat{y}_{l}^{[1]} n_{1} * (n_{1} * = F)$  is so called freedom of choice) most precise signals by some chosen criterion (mostly by the mean squared error  $\sigma_{y_{l}^{[1]}}^{2}$ ). Among these  $n_{1}^{*}$  best outputs of the first hidden layer  $\hat{y}_{l}^{[1]} * n_{2}$  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  $\sigma_{y_{l}^{2}}^{2}$ ) if the best signal of the second layer is better than the best signal of the first hidden layer  $\hat{y}_{l}^{[1]}$ . Other hidden layers forms signals similarly. The system evolution process continues until 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.

Investigation of the effectiveness of artificial neural networks of different generations

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 [5; 6]. Besides, since the training is performed sequentially layer by layer the problems of high dimensionality as well as vanishing or exploding gradient are avoided.

Let's introduce into consideration the architecture of the node that is suggested as a neuron of the GMDH-system. As a node of this structure a neo-fuzzy neuron (NFN) by Takeshi Yamakawa and co-authors in is used [14]. The neofuzzy neuron is a nonlinear multi-input single-output system shown in Fig. 3. 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:

$$\hat{y} = \sum_{i=1}^{2} f_i(x_i),$$
(6)

where  $x_i$  is the input i (i = 1, 2, ..., 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

$$f_i(x_i) = \sum_{j=1}^h w_{ji} \mu_{ji}(x_i)$$
(7)

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  $\mu_{ii}$ ,  $w_{ji}$  is a synaptic weight in consequent.

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

$$E(k) = \frac{1}{2}(y(k) - \hat{y}(k))^2 = \frac{1}{2}e(k)^2 = \frac{1}{2}\left(y(k) - \sum_{i=1}^2 \sum_{j=1}^h w_{ji}\mu_{ji}(x_i(k))\right)^2.$$
 (8)

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

$$w^{[1]}(N) == \left(\sum_{k=1}^{N} \mu^{[1]}(k) \mu^{[1]T}(k)\right)^{+} \sum_{k=1}^{N} \mu^{[1]}(k) y(k) = P^{[1]}(N) \sum_{k=1}^{N} \mu^{[1]}(k) y(k), \quad (9)$$

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

$$w_{l}^{ij}(k) = w_{l}^{ij}(k-1) + \frac{P^{ij}(k-1)(y(k) - (w_{l}^{ij}(k-1))^{T} \varphi^{ij}(x(k)))\varphi^{ij}(x(k))}{1 + (\varphi^{ij}(x(k)))^{T} P^{ij}(k-1)\varphi^{ij}(x(k))},$$
  

$$P^{ij}(k) = P^{ij}(k-1) - -\frac{P^{ij}(k-1)\varphi^{ij}(x(k))(\varphi^{ij}(x(k)))^{T} P^{ij}(k-1)}{1 + (\varphi^{ij}(x(k)))^{T} P^{ij}(k-1)\varphi^{ij}(x(k))}.$$
 (10)

Consider the HSCI-bagging network (Fig. 4). It is a hybrid system of computational intelligence (HSCI) built on the basis of the ensemble approach and batching, which builds its architecture in the process of learning based on the ideas of GMDH [15].

$\overrightarrow{x_1(k)}$	e [1]	$\hat{y}_{1}^{[1]}(k)$	el [1]	$y^{*[1]}(k)$	e [2]	$\hat{y}_1^{[2]}(k)$	el [2]	$y^{*[2]}(k)$	e [S]	$\overrightarrow{\hat{y}_1^{[s]}(k)}$	el [S]	
$\overrightarrow{x_2(k)}$	sembl	$\hat{y}_2^{[1]}(k)$	a mod	$\overrightarrow{\hat{y}_{1\mu}^{[1]}(k)}$	sembl	$\overrightarrow{\hat{y}_2^{[2]}(k)}$	a mod	$\overrightarrow{\hat{y}_{1\mu}^{[2]}(k)} \dots \rightarrow$	semblo	$\hat{y}_2^{[s]}(k)$	a mod	$\overrightarrow{y^{+[s]}(k)}$
$\overrightarrow{x_n(k)}$	En	$\hat{y}_{a}^{[1]}(k)$	met	$\frac{\vdots}{\hat{y}_{k+1,\mu}^{[1]}(k)}$	En	$\hat{y}_q^{[2]}(k)$	met	$ \stackrel{:}{\stackrel{\longrightarrow}{\hat{y}_{k+1,\mu}^{[2]}}} \xrightarrow{\dots}  $	En	$\overrightarrow{\hat{y}_q^{[s]}(k)}$	met	

Fig. 4. Hybrid system of computational intelligence based on bagging and GMDH

The architecture of the system contains 2S sequentially connected stacks, while odd stacks are formed by ensembles of parallel-connected subsystems that solve the same problem (recognition, prediction, etc.) and even ones are essentially learning metamodels that generalize the output signals of ensembles and form optimal results in the sense of the accepted criterion. The output signal of the first metamodel is the generalized optimal signal  $y^{*1}(k)$  and (n-1) output signals  $\hat{y}_{i\mu}^{[1]}(k)$ , i=1,2...,n-1(k) "best members of the ensemble". At their core,

metamodels function as selection units in traditional GMDH systems, but not only select the best results from the previous stack, but also form the optimal solution based on these results.

Further, the output signals of the first metamodel are fed to the inputs of the second ensemble, which is completely similar to the first. The outputs of the second ensemble  $\hat{y}_1^{[2]}(k)$ ,  $\hat{y}_2^{[2]}(k)$ ,...,  $\hat{y}_q^{[2]}(k)$  come to the second metamodel, which calculates the optimal signal  $y^{*[2]}(k)$  and  $(n-1)\hat{y}_{i\mu}^{[2]}(k)$  "closest" to it. The last *S*-th ensemble is similar to the first two, and the output of the last *S*-th metamodel is  $y^{*[s]}(k)$ , which exactly corresponds to a priori established requirements for the quality of solving the problem under consideration.

Each of the ensembles contains q different computational intelligence systems that solve the same problem. There may still be simple neural networks such as a single-layer perceptron, radial- basis neural network (RBFN), counterpropagating neural network, etc., which do not use error backpropagation procedure for training, neuro-fuzzy systems such as ANFIS, Wang–Mendel or Takagi–Sugeno–Kang type, wavelet-neuro systems, neo-fuzzy neurons and others, the output signal of which linearly depends on the adapted parameters, which allows to use optimal speed learning algorithms.

The input information, on the basis of which the system is configured, is a training selection of input signals:

$$x(1), x(2), ..., x(k), ..., x(N),$$
  
$$x(k) = (x_1(k), ..., x_i(k), ..., x_n(k))^{\mathrm{T}} \in \mathbb{R}^n,$$
(11)

and its corresponding scalar referce signals y(1),...,y(k),...,y(N). On the basis of these observations, the elements of the first ensemble are tuned independently of each other, at the outputs of which q scalar signals  $\hat{y}_p^{[2]}(k)$ , p = 1,2,...,q, are formed, which are conveniently represented in the form of a vector  $\hat{y}^{[1]}(k) = (\hat{y}_1^{[1]}(k),...,\hat{y}_p^{[1]}(k),...,\hat{y}_q^{[1]}(k))^T$ . These signals are sent to the inputs of the first metamodel, at the outputs of which n sequences  $\hat{y}^{*[1]}(k)$ ,  $\hat{y}_{1\mu}^{[1]}(k),...,\hat{y}_{n-1,\mu}^{[1]}(k)$  the main of which is  $y^{*[1]}(k)$  while others are auxiliary. The main signal of the metamodel  $y^{*[1]}(k)$  is the union of the outputs of all members of the ensemble in the form of:

$$y^{*[1]}(k) = \sum_{p=1}^{q} w_p^{*[1]} \hat{y}_p^{[1]}(k) = \hat{y}^{[1]T}(k) w^{*[1]}, \qquad (12)$$

where  $w^{*[1]} = (w^{*[1]}, ..., w_p^{*[1]}, ..., w_q^{*[1]})^T$  is a vector of adapted parameters-synaptic weights on which additionally restrictions are set on unbiasedness:

$$\sum_{p=1}^{q} w_p^{*[1]} = I_q^T w^{*[1]} = 1,$$
(13)

where  $I_q - (q \times 1)$  is the vector of unities.

The problem of teaching the first metamodel is reduced to minimizing the standard quadratic criterion in the presence of additional constraints.

Thus, the problem of training the first metamodel can be solved using the standard method of penalty functions, which in this case reduces to minimizing the expression:

$$J(w^{*[1]},\delta) = (Y(N) - \overline{Y}^{[1]}(N)w^{*[1]})^{\mathrm{T}}(Y(N) - \overline{Y}^{[1]}(N)w^{*[1]}) + \delta^{-2}(I_q^T w^{*[1]} - 1), \qquad (14)$$

where  $Y(N) = (y(1), ..., y(k), ..., y(N))^{T}$  — is a vector,  $\overline{Y}^{1}(N) = (\hat{y}^{[1]}(1), ..., \hat{y}^{[1]}(k), ..., \hat{y}^{[1]}(N))^{T}$  —  $(N \times q)$  is a matrix,  $\delta$  is the penalty coefficient.

As a result of learning the first metamodel, the optimal signal  $y^{*[1]}(k)$  is formed at its output, as well as q signals  $\hat{y}_{p,\mu}^{[1]}(k)$  from which we choose n-1 (if  $q \ge n$ ) with the highest levels of fuzzy membership  $\mu_p^{[1]}$ , which subsequently in the form of  $(n \times 1)$  — vector are fed to the input of the second ensemble, the outputs of which go to the inputs of the second metamodel, and so on. The process of increasing the number of ensembles and metamodels continues until the required accuracy of the last metamodel with the output  $y^{*[s]}(k)$  is achieved, or the value of the criterion minimized for the bagging model begins to increase, i.e.  $\overline{\epsilon}^2 (y^{*[s+1]}(k)) \ge \overline{\epsilon}^2 (y^{*[s]}(k))$ .

#### DATA SET

Data on the dynamics of changes in the Dow Jones Industrial Average (DJIA) index from August 2, 2022 to July 31, 2024 were used for forecasting [16]. The dynamics of DJIA Close values is shown in the Fig. 5.



Fig. 5. Dynamics of the index Close DJIA

ISSN 1681–6048 System Research & Information Technologies, 2025, № 1



The correlogram of DJIA Close vales is presented in the Fig. 6.

Fig. 6. Correlogram of the index Close DJIA

Analyzing the presented curve, one may conclude that there is strong correlation between preceding and conceding values and even for lag 50 days the correlation is more than 0.7.

#### EXPERIMENTAL INVESTIGATION AND DISCUSSION

Experimental studies of the accuracy of index forecasting using networks were conducted: Back Propagation, LSTM, HMDH-neo-fuzzy, and HSCI-bagging. During the experiments, the values of the parameters presented in Table 1 were changed. The data set was split into three subsets: training, validation, and test. The test subset of data for all experiments had a fixed size (30 last points from the dataset) and was not used for training and validation.

<b>Table 1.</b> Experimental Paramet	ers
--------------------------------------	-----

Parameter	Values		
Interval	1; 3; 5; 7; 15; 20; 30		
Number of inputs	3; 4; 5		
Number of fuzzifiers	2; 3; 4		
Validation split	0.4; 0.3; 0.2		

After training, the accuracy of the models was checked on a test sample. For each network, the best results of the forecast accuracy according to the MAPE criterion were determined.

The first set of experiments was conducted with a back propagation network (2nd generation). It had three hidden layers: the first layer of 7 neurons, the second layer of 5 neurons, and the third layer of 3 neurons. The output was a single signal. The best prediction results of this network for all intervals are shown in Table 2.

Interval	Number of inputs	Validation split	MSE	MAPE
1	3	0.2	1644597.0402	2.9792
3	3	0.2	1824141.7299	3.0973
5	3	0.2	1820227.1563	3.1028
7	3	0.2	1847014.9194	3.1448
15	4	0.2	1926517.248	3.175
20	4	0.2	4176396.3173	4.8037
30	3	0.2	24150324.9138	12.1998

**Table 2.** The best results of the Back Propagation network

The second set of experiments investigated the prediction accuracy of the 3rd generation network — LSTM. It had the following structure: input signals determined by an experimental parameter, three hidden layers (32 neurons, 16 neurons, and 8 neurons), and one neuron with an output signal. For all forecasting intervals, the best results were determined and are shown in Table 3.

For the Back Propagation and LSTM networks, the structure was selected using Cross-Validation and Grid Search methods. As a result, optimal structures were obtained and used that do not have too many hidden layers. This approach made it possible to use minimal computational costs to obtain sufficiently high accuracy of the results.

Interval	Number of inputs	Validation split	MSE	MAPE
1	5	0.2	96500.7026	0.582
3	4	0.2	258966.107	0.979
5	5	0.2	474929.5015	1.3487
7	5	0.2	537666.5732	1.4243
15	3	0.2	1159702.2855	2.0442
20	4	0.2	2847643.2154	3.7911
30	4	0.3	4233239.0843	4.6857

Table 3. The best results of the LSTM network

The third set of experiments was conducted to determine the forecasting accuracy of the 4th generation network — GMDH-neo-fuzzy. The structure of the network was synthesized during training and in most cases had two hidden layers and one layer with the output signal. After comparing the obtained forecasting results on the test subsample, Table 4 was created with the best results and optimal network parameters for all intervals.

Table 4. The best results of the GMDH-neo-fuzzy

Int.	Number of inputs	Number of fuzzifiers	Validation split	MSE	MAPE
1	3	2	0.3	155027.4886	0.7004
3	3	3	0.2	332615.1181	1.1576
5	5	3	0.3	417632.8307	1.198
7	5	4	0.3	394795.8022	1.2579
15	5	3	0.4	622501.8958	1.6785
20	4	3	0.3	1086960.4794	2.204
30	4	3	0.3	1138011.8375	2.353

The last series of experiments was conducted to evaluate the prediction accuracy of the HSCI-bagging network, which also belongs to the 4th generation. The previous networks were used for its implementation. The best forecasting results are presented in Table 5.

Interval	Number of inputs	Validation split	MSE	MAPE	
1	5	0.2	80602.0465	0.5054	
3	4	0.2	248445.23	0.9685	
5	4	0.2	384022.2476	1.1568	
7	4	0.2	409905.6681	1.2263	
15	5	0.2	595730.7206	1.6062	
20	5	0.2	827577.2912	2.011	
30	4	0.2	978388.1249	2.1217	

Table 5. The best results of the HSCI-bagging

Based on the data from Tables 2–5, Table 6 was created to compare the forecasting results according to the MAPE criterion.

Interval	<b>Back Propagation</b>	LSTM	GMDH-neo-fuzzy	HSCI-bagging
1	2.9792	0.582	0.7004	0.5054
3	3.0973	0.979	1.1576	0.9685
5	3.1028	1.3487	1.198	1.1568
7	3.1448	1.4243	1.2579	1.2263
15	3.175	2.0442	1.6785	1.6062
20	4.8037	3.7911	2.204	2.011
30	12.1998	4.6857	2.353	2.1217

**Table 6.** Comparative table of the best forecasting results

For the convenience of analyzing the results, a comparative graph of the average forecast accuracy according to the MAPE criterion for all the investigated networks at each of the intervals was also constructed. This graph is shown in Fig. 7.



Fig. 7. Comparative diagram of forecast accuracy according to the MAPE criterion

Thus, the results of the conducted studies show that the Back Propagation network showed the worst results at all intervals. The best forecasting accuracy according to the MAPE criterion was obtained using the 4th generation HSCI-bagging network, its slightly better than hybrid GMDH-neo-fuzzy network. The LSTM recurrent network showed good results on short-term intervals, but starting from interval 5, it is inferior to the GMDH-neo-fuzzy network.

#### CONCLUSION

This article considers the problem of short- and middle-term forecasting in the financial sector using the Dow Jones Industrial Averagse (DJIA) dataset.

Experimental investigations of the forecasting accuracy of neural networks of different generations were conducted: Back Propagation (2nd generation), LSTM (3rd generation), GMDH-neo-fuzzy (4th generation) and HSCI-bagging (4th generation).

During the experiments, at each of the short- and medium-term intervals, the optimal parameters for each of the networks were determined, at which it demonstrated the best forecasting results.

The accuracy of forecasts by the MAPE criterion of all networks at short and medium-term intervals was compared. The best forecasting results were obtained using HSCI-bagging, and the GMDH-neo-fuzzy hybrid network showed slightly worse results, but better than other studied networks of previous generations.

The results of the investigation show that, in general, the forecasting accuracy increases with the generation of neural networks. In addition, the latest generations of artificial neural networks have shown better results on medium-term intervals.

### REFERENCES

- 1. Frank Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.
- David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986. doi: 10.1038/323533a0
- G.V. Cybenko, "Approximation by Superpositions of a Sigmoidal function," Mathematics of Control, Signals and Systems, vol. 2, no. 4, pp. 303–314, 1989, doi: 10.1007/BF02551274
- 4. I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. Available: http://www.deeplearningbook.org
- 5. A.G. Ivakhnenko, V.G. Lapa, *Cybernetic forecasting devices* (in Ukrainian). K.: Naukova Dumka, 1965, 216 p.
- A.G. Ivakhnenko, G.A. Ivakhnenko, and J.A. Mueller, "Self-organization of the neural networks with active neurons," *Pattern Recognition and Image Analysis*, vol. 4, no. 2, pp. 177–188, 1994.
- 7. Jürgen Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, pp. 85–117, 2015. doi: 10.1016/j.neunet.2014.09.003
- 8. Purwono Purwono et al., "Understanding of Convolutional Neural Network (CNN): A Review," *International Journal of Robotics and Control Systems*, vol. 2, no. 4, pp. 739–748, 2023. doi: 10.31763/ijrcs.v2i4.888
- 9. B. Hammer, "On the approximation capability of recurrent neural networks," *Neuro-computing*, vol. 31, pp. 107–123, 1998. doi: 10.1016/S0925-2312(99)00174-5
- 10. S. Hochreiter, J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735

- 11. C. Olah, *Understanding LSTM networks*. 2020. Available: https://colah.github.io/ posts/2015-08- Understanding-LSTMs
- Yu. Zaychenko, Galib Hamidov, "The Hybrid Deep Learning GMDH-neo-fuzzy Neural Network and Its Applications," *Proceedings of 13-th IEEE International Conference Application of Information and Communication Technologies-AICT2019.* 23–25 October 2019, Baku, pp. 72–77. doi: 10.1109/AICT47866.2019.8981725
- Evgeniy Bodyanskiy, Yuriy Zaychenko, Olena Boiko, Galib Hamidov, and Anna Zelikman, "Structure Optimization and Investigations of Hybrid GMDH-Neo-fuzzy Neural Networks in Forecasting Problems," *System Analysis & Intelligent Computing*; Eds. Michael Zgurovsky, Natalia Pankratova. Book Studies in Computational Intelligence, SCI, vol. 1022. Springer, 2022, pp. 209–228.
- T. Yamakawa, E. Uchino, T. Miki, and H. Kusanagi, "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.
- Ye. Bodyanskiy, O. Kuzmenko, He. Zaichenko, and Yu. Zaychenko, "Hybrid System of Computational Intelligence based on Bagging and Group Method of Data Handling," *System Research and Information Technologies*, no. 1, pp. 75–85, 2024. doi: 10.20535/SRIT.2308-8893.2024.1.06
- "DJIA Dow Jones Industrial Average Historical Prices," WSJ. Accessed on: August 1, 2024. [Online]. Available: https://www.wsj.com/market-data/quotes/index/DJIA/ historical-prices

Received 04.09.2024

#### **INFORMATION ON THE ARTICLE**

Yevgeniy V. Bodyanskiy, ORCID: 0000-0001-5418-2143, Kharkiv National University of Radio Electronics, Ukraine, e-mail: yevgeniy.bodyanskiy@nure.ua

**Yuriy P. Zaychenko,** ORCID: 0000-0001-9662-3269, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: zaychenkoyuri@ukr.net

**Helen Yu. Zaichenko**, ORCID: 0000-0002-4630-5155, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: zaichenko.helen@lll.kpi.ua

**Oleksii V. Kuzmenko,** ORCID: 0000-0003-1581-6224, Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: oleksii.kuzmenko@ukr.net

ДОСЛІДЖЕННЯ ЕФЕКТИВНОСТІ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ (ШНМ) РІЗНИХ ПОКОЛІНЬ У ЗАДАЧІ ПРОГНОЗУВАННЯ У ФІНАНСОВІЙ СФЕРІ / Є.В. Бодянський, Ю.П. Зайченко, О.Ю. Зайченко, О.В. Кузьменко

Анотація. Розглянуто ШНМ різних поколінь. Досліджено ефективність використання обчислювального інтелекту в задачах коротко- та середньострокового прогнозування у фінансовій сфері. Для дослідження обрано повнозв'язну мережу прямого поширення (Back Propagation), рекурентну мережу (LSTM), гібридну мережу глибокого навчання на основі самоорганізації (GMDH-neofuzzy) та гібридну систему обчислювального інтелекту на основі беггінгу та методу групового урахування аргументів (HSCI-bagging). Як експериментальні параметри обрано інтервал прогнозування, кількість входів, відсоток валідаційних даних у навчальній вибірці та кількість фазифікаторів (для GMDH-neofuzzy). Проведено експерименти та порівняно найкращі результати, отримані для різних інтервалів прогнозування. Визначено оптимальні параметри мереж та доцільність їх використання в задачі прогнозування на різних інтервалах.

Ключові слова: покоління ШНМ, Back Propagation, LSTM, GMDH neo fuzzy, HSCI bagging.