Classification of the Technological Process Condition Based on Hybrid Neural Networks

Andrey Puchkov, Maxim Dli, and Yekaterina Lobaneva

Abstract The cascade architecture of neural networks, providing multichannel information processing as a part of a cyber-physical system, is presented. The architecture contains a hybrid network for aggregation of intermediate results obtained by an ensemble of deep recurrent neural networks based on fuzzy logic methods. The structure of the software which implements the presented architecture and being developed for conducting simulation experiments is described, their results are given.

Keywords Machine learning · Deep neural networks · Computer vision

1 Introduction

The improvement of the information support for cyber-physical systems now tends to be based on the use of machine learning methods which automate the procedure of extracting the necessary regularity from incoming data. Among these methods, deep neural networks gained the greatest popularity, which was facilitated by the emergence of new efficient neural network architectures and significantly increased abilities for computing technology in the middle price segment. This allowed many researchers to conduct algorithms learning and implement them in practice with the use of ready-made hardware solutions [1, 2].

The use of such solutions for complex industrial objects is faced with the need to process multi-channel technological information, therefore, in this case, ensembles of neural networks, each element of which takes into account the presentation

A. Puchkov (✉) · M. Dli · Y. Lobaneva
National Research University «Moscow Power Engineering Institute» (Branch), Energetichesky proyezd 1, g. Smolensk, Smolensk 2014013, Russia
e-mail: putchkov63@mail.ru

M. Dli
e-mail: MiDli@mail.ru

Y. Lobaneva
e-mail: lobaneva94@mail.ru

A. G. Kravets et al. (eds.), *Cyber-Physical Systems: Modelling and Intelligent Control*, Studies in Systems, Decision and Control 338,
https://doi.org/10.1007/978-3-030-66077-2_8

features and data format in a particular channel, are becoming very popular. The result combination for the work of channel neural networks for working out any generalizing solution, in this case, is carried out in the output analytical unit. The algorithms for the operation of such blocks are various, and they are based on additional cascades of neural networks [3, 4]; the following approaches are also used: fuzzy aggregation of ensemble output [5, 6]; additional information for ensemble learning [7]; self-organization of networks [8]; modifications of the Bayesian approach allowing compensation the imbalance of existing data [9]. Each of the above-mentioned approaches develops applied directions for neural networks usage taking into account a certain aspect of a particular application or presentation features and initial data form when combining the results for channel information processing. The existing wide variety of applied fields for neurotechnologies, which exist nowadays, ensures the relevance of research in this direction.

This work proposes a neural network architecture that provides multi-channel information processing as a part of a cyber-physical system in order to assess its condition. The architecture contains the input ensemble for the deep recurrent neural network (DRNN) and output hybrid neural network to aggregate the intermediate results of the presented data. Such a composition of neural networks allows predicting the parameters of a cyber-physical system due to DRNN and classifying its condition using an output hybrid (neuro-fuzzy) system which set of rules, generated automatically during learning, can be adjusted if necessary.

2 Materials and Methods

A specific feature of the applied aspect of neural networks in cyber-physical systems is the need to adapt the resulting solutions to the information-technological environment of the physical process, which is based on the experience of a developer both in the field of information and hardware. This process for neural networks consists of the analysis of the receiving result and further adjustment of network architectures and their hyperparameters (initial parameters of networks and optimization procedures which are unchanged during network training).

The absence of universal network architectures leads to the necessity to use various techniques to improve the results of data analysis. Boosting being one of them. It consists of the sequential aggregation of different algorithms in which each subsequent one uses the results of the previous ones for learning, this allows obtaining a stronger solution, high generalizing ability, and universality [10]. To a certain extent, the introduction of methods, which allow improving the understanding of how the neural network receives the result since this provides an opportunity for actions to improve it, can also be attributed to boosting. For example, in [11, in Russian] taking into account the symmetries in the incoming signals, we propose a grouping of neurons in a hidden layer according to the form of automorphism and the use of three-phase activation functions for each group. In many cases a similarity of a multi-agent approach is used: sets of relatively simple neural networks with different hyperparameters are created, and then their outputs are analyzed according to various schemes [12].

The authors propose a hybrid architecture for the classifier of the technological process (TP) condition for the production of phosphorus from apatite-nepheline ores waste [13, in Russian], shared DRNN and ANFIS networks (adaptive neuro-fuzzy inference system is an adaptive network based on the fuzzy inference system of Takagi–Sugeno) [14]. This structure provides a mechanism to introduce into the "closed" neural network classifier the possibility of recording expert knowledge due to adjusting the rule base synthesized by the ANFIS system and allows making classification based on additional knowledge. The approach under consideration makes the understanding of classification results accessible since the output cascade of the neural networks ensemble the interpretation for the previous signals will be visible.

It should be noted, that hybrid approaches for the solver building in intelligent systems are often used in modifications of various subject areas and the ANFIS option is not the only one that can be used. One of the alternatives is the «coactive neuro-fuzzy inference system» (CANFIS), which also combines fuzzy logic and neural networks, achieving an increase in the power of intelligent systems through the use of verbal and numerical description of the subject area [15].

The proposed hybrid architecture contains the cascade inclusion of DRNN block and adaptive ANFIS networks, therefore, the output fuzzy classifier ANFIS gets a possibility to forecast the TP condition taking into account its behavior in the past due to the presence of such ability in DRNN.

Let the described architecture and its application be considered for the specific application for which it was specially created: for the processing of technological information of the system for phosphorus production from apatite-nepheline ores waste. The cyber-physical system consists of an intelligent information superstructure and three units (Fig. 1) implementing TP:

- a pelletizer (it forms raw pellets from ores waste);
- a multi-chamber indurating machine of a conveyer type (it provides high-temperature roasting of raw pellets);
- an ore-thermal furnace (pellets are melted in it with the release of gaseous phosphorus);
- The listed units are indicated in Fig. 1 as control objects CO1, CO2, and CO3, respectively.

DRNN application is justified by their ability to accumulate knowledge for some retrospective time period, which is in demand in the TP control, as it makes it possible to evaluate its condition not only by current parameters but also taking into account their past behavior. It is especially important for complex TP in which different stages are differentiated both in time and in space and implemented on various units. The DRNN ensemble provides the approximation of the historic behavior and spatial division of the controlled system into separate technological zones. The ANFIS block, in its turn, automates the receiving of final conclusions based on the results provided by recurrent networks.

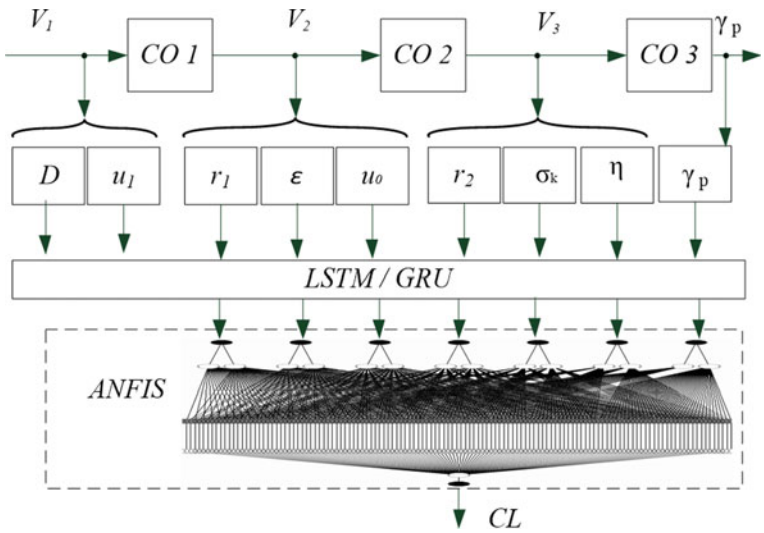


Fig. 1 The hybrid architecture of the information processing in the cyber-physical system

The application of mathematical models for transformation processes of the feedstock into the final product by each unit makes it possible to determine the condition variables at the input/output of each unit [13]. When preparing the learning sets variables are united into row vectors:

$$V_1 = [Du_1],\ V_2 = [r_1 \varepsilon u_0],\ V_3 = [r_2 \sigma_k \eta], \tag{1}$$

where D—a middle diameter for the waste particles of apatite-nepheline ores (mm); u_1—specific humidity of ores waste (%); r_1—a radius of a raw pellet (mm), ε—a pellet porosity (in fractions of the volume of the entire pellet), u_o—pellets humidity at the exit from the granulator (%); r_2—a radius of a roasted pellet (mm); σ_k—a pellet compressive strength (kN/pellet); η—a degree of response in decarbonization reactions (a fraction of the reacted substance from its total volume). It should be mentioned, that the output row vector V2 of the granulator is the input one for the indurating machine, and its output row vector V3 is the output one for the ore-thermal furnace, which output parameter is the purity of the resulting phosphorus γp (% from the general volume of the output fractions).

The listed parameters, taken at discrete intervals of time Δt, form an initial data matrix which is constantly updated with new values as the observation time increases. For the neural network training the data taken for a long time period, which provides a sufficient amount of the learning sample, is used. The data are stored in the form of a file with CSV format, which fragment reflecting the typical values for the TP parameters is shown in Fig. 2.

From the whole variety of DRNN for the considered architecture, two types of networks were selected, which have the best representative power and have a

Fig. 2 Data file fragment

	A	B	C	D	E	F
1	N_time, D, u1, r1, eps, u0, r2, sigma_k, etta, gamma_p					
2	0.0,1.032,0.513,11.05,0.417,12.09,10.92,2.97,0.608,34.40					
3	12.0,1.051,0.532,10.59,0.568,12.13,10.41,2.82,0.561,37.14					
4	24.0,1.049,0.501,11.10,0.612,11.43,11.12,2.71,0.483,42.65					
5	36.0,1.001,0.513,11.07,0.609,12.17,10.94,2.93,0.721,39.17					

wide range of practical applications. These networks are based on the algorithm of long short-term memory (LSTM, Long Short-Term Memory) [16, 17] and the network GRU (Gated Recurrent Unit) [18, 19], applied LSTM principles but having fewer numbers of filters and operations for calculations. LSTM and GRU layers allow coping with a problem of gradient attenuation observed in direct propagation networks with a large number of layers, as the number of layers increases, the network eventually becomes unlearning [20]. DRNN (in variants LSTM and GRU) in the proposed architecture are used for solving the problem of regression.

DRNN and ANFIS learning is carried out separately in two stages. At the first stage, DRNN is trained, at the second stage ANFIS is trained. Before learning, data are prepared and normalized. From the CSV file, the information is entered into the matrix: MV = {[V1 V2 V3 γp]i}, where i = 0, 1, 2, … Nmv, where Nmv—the number of rows in the table read from the file. From this matrix, learning and test sets are formed taking into account the following control parameters: lookback—the number of time intervals from the current moment, determining the number of rows in one learning data packet; delay—the interval of the forecast (if delay = 0, then the current condition is estimated); min_index and max_index—the indexes restricting data retrieval from MV; shuffle—a logic key, at shuffle = 1 rows are retrieved from MV with shuffling, at shuffle = 0 without shuffling; batch_size—a number of samples in one packet, according to which the network weights are adjusted; step—a period in intervals specifying the value of lines decimation from MV.

The LSTM/GRU unit (Fig. 1) uses several DRNN connected in parallel, which differ in the number of components in the input data vectors and the output parameter. For networks DRNN1–DRNN3 the input vector is the combined row vector [V1 V2], and the networks differ in what parameter is taken for them as the output one: r_1, ε, or u_o: r_1, respectively. For DRNN4–DRNN7 networks the sets of vectors [V2 V3] are the input parameters. DRNN4–DRNN7 are the output parameters in networks: r_2, σ_k, η, and γp respectively.

Without dwelling on the algorithms of the networks, we only note that the output yk k-th DRNN, k = 1, 2, …, 7, is formed in accordance with the expression:

$$y_k = f\,(state\_t_k \circ U_k + BV_k \circ W_k + b_k), \tag{2}$$

where $state\_t_k$, U_k, b_k—learning parameters of k-th LSTM for the current moment of time t, ∘—a symbol of Hadamard matrices (component-wise product), BVk—the input packet of learning data.

It should be noted, that LSTM/GRU contains a sequentially located set of recurrence units and can only process a parameter row [20]. However, the learning algorithms provide for a packet supply of initial data to the network input (in accordance with the above-mentioned list of control parameters), therefore, state state_t_k, U_k, and b_k t_k, U_k, are matrixes.

3 Application and Results

DRNN networks were trained in the mode of sequence to end, when not complete results sequences were returned for all time intervals, but only the last result for each input sequence. GRU application in the architecture shown in Fig. 1 allows reducing the computational cost for network learning due to their simpler structure, compared with LSTM, as noted above. The use of GRU together with LSTM is justified by the appropriateness of testing simpler solutions if they allow achieving the specified accuracy for TP condition estimation.

In the second stage, the ANFIS network is trained, this network has seven inputs (by the number of DRNNs) and one CL output having a range of values from 0 to 1. The network solves the problem of classifying the condition for TP based on the results of its evaluation of DRNN. It was accepted that the condition of the TP can be divided into four classes (depending on the value of the parameter γp): "nominal value", "nominal value excess", "small deviation down from the nominal value" and "big deviation down from the nominal value" (Table 1).

The entire range (from 0 to 100%) is a technologically permissible spread in values of the parameter γp, the exceeding of which is already monitored by instrumentation.

The program that implements the architecture provides a control parameter delay, depending on it the current condition can be evaluated (delay = 0) or forecasted (delay > 0). Unlike DRNN, ANFIS learning requires a significantly smaller training sample due to the relatively small number of adjustable network parameters [15].

To test the proposed neural network architecture simulation experiments were performed in MatLab 2019b environment, which has a specialized machine learning

Table 1 Classes numbers of the output parameter for the ore-thermal furnace

Class number	Class name	Interval of values for a parameter γ_p, %	ANFIS output, appropriate to class
1	Nominal value	40–60	0.75
2	Exceeding the nominal value	61–100	1.00
3	Small deviation down from the nominal value	30–39	0.50
4	Big deviation down from the nominal value	0–29	0.25

Fig. 3 Hybrid network layer scheme for one channel of data processing

library Deep Network Designer. Before learning, the data were normalized using the operations of subtraction the mean and roof-mean-square deviation. A simplified scheme of the channels layers for processing TP parameters is shown in Fig. 3.

Each of seven LSTM was trained during 250 epochs, the number of samples in the training dataset was 50,000, and in the testing one, it was 10,000. To avoid retraining the decimation mechanism was used. The training quality estimation (Fig. 4) was carried out with the use of two metrics: the root-mean-square error and loss function [21].

ANFIS configuration: the number of inputs—7; the number of membership functions for each input—2; the type of membership functions—trimf; the number of outputs—1; the type of membership functions—linear. In the process of ANFIS training ANFIS, a set of rules was generated, the structure of which is reflected in the fragment shown in Fig. 5.

The neural networks were trained on the following hardware: ASUS TUF Gaming FX705DT-AU039 notebook, AMD Ryzen 7 3750H CPU, 2.3 GHz, NVIDIA GeForce GTX 1650 4G GPU, 1024 CUDA cores (provide parallel computing). Classification results are presented in Fig. 5.

The crosses indicate the output of the hybrid architecture classifier, the dots indicate the levels corresponding to the true classes. The forecasting parameter delay = 0 was suggested, which means the recognition of the current condition, however,

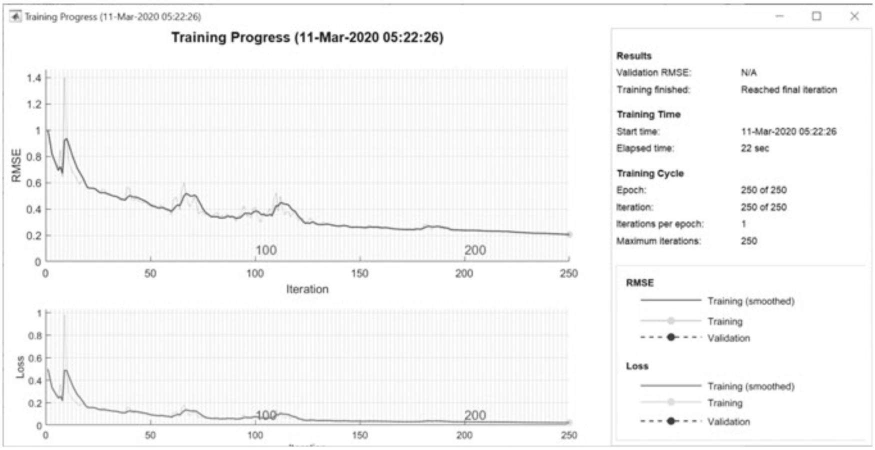
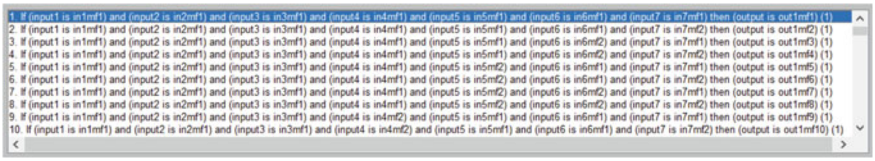


Fig. 4 LSTM metrics dynamics in the training process



1. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf1) and (input7 is in7mf1) then (output is out1mf1) (1)
2. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf1) and (input7 is in7mf2) then (output is out1mf2) (1)
3. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf2) and (input7 is in7mf1) then (output is out1mf3) (1)
4. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf1) and (input6 is in6mf2) and (input7 is in7mf2) then (output is out1mf4) (1)
5. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf1) and (input7 is in7mf1) then (output is out1mf5) (1)
6. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf1) and (input7 is in7mf2) then (output is out1mf6) (1)
7. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf2) and (input7 is in7mf1) then (output is out1mf7) (1)
8. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf1) and (input5 is in5mf2) and (input6 is in6mf2) and (input7 is in7mf2) then (output is out1mf8) (1)
9. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf1) and (input7 is in7mf1) then (output is out1mf9) (1)
10. If (input1 is in1mf1) and (input2 is in2mf1) and (input3 is in3mf1) and (input4 is in4mf2) and (input5 is in5mf1) and (input6 is in6mf1) and (input7 is in7mf2) then (output is out1mf10) (1)

Fig. 5 A fragment of ANFIS rules set

this does not affect the quality of the classification performed by ANFIS, since the LSTM/GRU unit is responsible for the forecast accuracy in this architecture (its accuracy characteristics are shown in Fig. 4). We note that in the experiment the replacement of LSTM by GRU practically did not affect the accuracy of the forecast, this fact can be due to a large number of training epochs or the features of the training data set.

The grouping of the classifier output values near the true class values in Fig. 6 can indicate the operability of the proposed hybrid architecture and the ability of its application to solve the problem for classifying the TP conditions. To obtain an integer class number from Table 1, it is possible to specify a valid range for the spread of the ANFIS output values. Some of the ANFIS outputs, underlined in Fig. 6, have

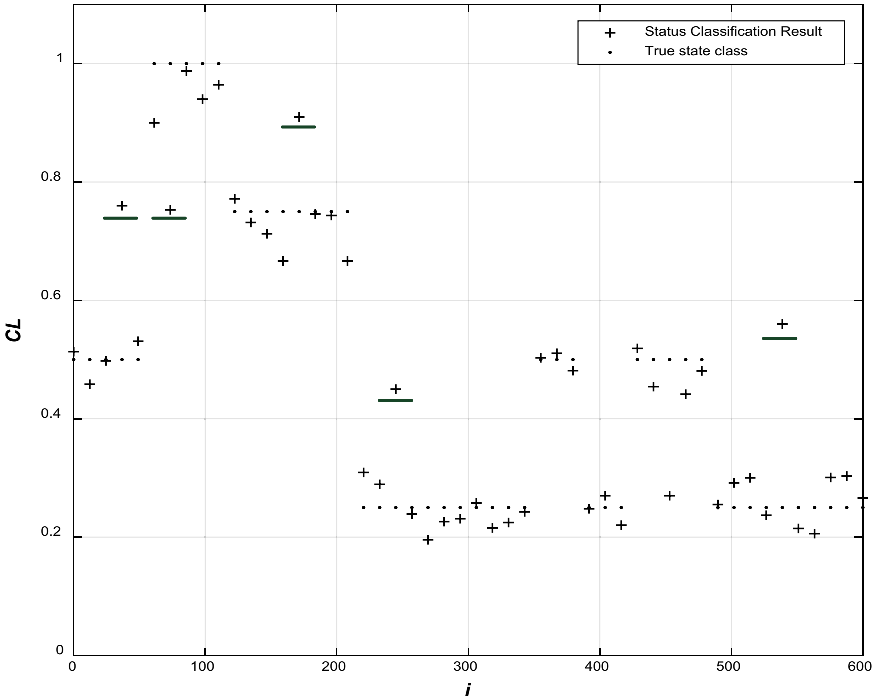


Fig. 6 Classification results

significant deviations from the true values, which reflect classification errors that can be reduced by modification of the hybrid architecture hyperparameters, for example, the number of ANFIS input membership functions can be increased to three.

4 Conclusion

The proposed hybrid architecture makes it possible to take advantage of two methodologies for constructing neural networks: to perform a retrospective analysis of time series using the DRNN ensemble and to generalize the results of their work with the ANFIS system providing an additional opportunity to make adjustments to the automatically created knowledge base in case of need.

The conducted simulation experiment showed the ability of the proposed hybrid architecture to carry out the classification of the process condition based on the stage-by-stage processing of the initial information using the DRNN and ANFIS systems.

The obtained results can be used in the development of knoware and software for systems of intelligent data analysis in various subject areas.

Acknowledgements The reported study was funded by RFBR according to the research projects No 18-29-24094 and No 19-01-00425.

References

1. Ivanavičius, A., Simonavičius, H., Gelšvartas, J., et al.: Real-time CUDA-based stereo matching using Cyclops2 algorithm. J Image Video Proc. **2018**, 12 (2018). https://doi.org/10.1186/s13640-018-0253-2
2. Moreno, J.J., Ortega, G., Filatovas, E., et al.: Using low-power platforms for evolutionary multi-objective optimization algorithms. J Supercomput **73**, 302–315 (2017). https://doi.org/10.1007/s11227-016-1862-0
3. Frazão X., Alexandre L.A.: Weighted Convolutional Neural Network Ensemble. In: Bayro-Corrochano, E., Hancock, E. (eds.) Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2014. Lecture Notes in Computer Science, vol. 8827. Springer, Cham (2014)
4. Koitka S., Friedrich C.M.: Optimized convolutional neural network ensembles for medical subfigure classification. In: Jones, G., et al. (eds.) Experimental IR Meets Multilinguality, Multimodality, and Interaction. CLEF 2017. Lecture Notes in Computer Science, vol. 10456. Springer, Cham (2017)
5. Puchkov A., Dli M.I., Lobaneva E., Vasilkova M.: Choice of a deep neural networks architecture to monitor the dynamics of an object state. In: Proc. of the 14th International Conference on Interactive Systems: Problems of Human–Computer Interaction (2019). http://ceur-ws.org/Vol-2475/paper12.pdf
6. Kulyasova, E.V., Kulyasov, N.S., Puchkov, A.Y.: The appliance of deep neural networks in the process of managing chemical enterprises. IOP Conf. Ser.: J. Phys. Conf. Ser. **1260**, 032024 (2019). https://doi.org/10.1088/1742-6596/1260/3/032024

7. Guo, Y., Wang, X., Xiao, P., et al.: An ensemble learning framework for convolutional neural network based on multiple classifiers. Soft Comput. **24**, 3727–3735 (2020). https://doi.org/10.1007/s00500-019-04141-w
8. Ståhl, N., Falkman, G., Mathiason, G., Karlsson, A.: A self-organizing ensemble of deep neural networks for the classification of data from complex processes. In: Medina J., Ojeda-Aciego, M., Verdegay, J., Perfilieva, I., Bouchon-Meunier, B., Yager, R., (eds.) Information Processing and Management of Uncertainty in Knowledge-Based Systems. Applications. IPMU 2018. Communications in Computer and Information Science, vol. 855. Springer, Cham (2018)
9. Lázaro, M., Herrera, F., Figueiras-Vidal, A.R.: Classification of binary imbalanced data using a Bayesian ensemble of Bayesian neural networks. In: Iliadis, L., Jayne, C. (eds.) Engineering Applications of Neural Networks. EANN 2015. Communications in Computer and Information Science, vol. 517. Springer, Cham (2015)
10. Dembski, J.: Multiclass AdaBoost classifier parameter adaptation for pattern recognition. In: Choraś, R. (ed.) Image Processing and Communications Challenges 8. IP&C 2016. Advances in Intelligent Systems and Computing, vol. 525. Springer, Cham (2017)
11. Lozhkin, A.G., Majorov, K.N.: The symmetry mechanism application in neural networks. In: Conference Proceedings: Seventeenth National Conference on Artificial Intelligence with International Participation of CII-2019, vol. 2, pp. 119–126. (2019)
12. Zhang, W., Jiang, J., Shao, Y., et al.: Snapshot boosting: a fast ensemble framework for deep neural networks. Sci. China Inf. Sci. **63**, 112102 (2020). https://doi.org/10.1007/s11432-018-9944-x
13. Puchkov, AYu., Dli, M.I., Lobaneva, E.I., Bliznyuk, O.A.: Optimization of energy consumption for a multistage phosphorus production process. Conf. Proc.: Math. Meth. Technol. Tech. **4**, 68–71 (2019)
14. Arun, K.S., Govindan, V.K.: A hybrid deep learning architecture for latent topic-based image retrieval. Data Sci. Eng. **3**, 166–195 (2018)
15. Tahmasebi, P., Hezarkhani, A.A.: Hybrid neural networks-fuzzy logic-genetic algorithm for grade estimation. Comput. Geosci. **42**, 18–27 (2012). https://doi.org/10.1016/j.cageo.2012.02.004. Accessed 09 Mar 2020
16. Liu, J., Chen, S.: Non-stationary multivariate time series prediction with selective recurrent neural networks. In: Nayak, A., Sharma, A. (eds.) PRICAI 2019: Trends in Artificial Intelligence. PRICAI 2019. Lecture Notes in Computer Science, vol. 11672. Springer, Cham (2019)
17. Althelaya, K.A., El-Alfy, E.M., Mohammed, S.: Stock market forecast using multivariate analysis with bidirectional and stacked (LSTM, GRU). In: 2018 21st Saudi Computer Society National Computer Conference (NCC), pp. 1–7. Riyadh (2018)
18. Ke, K., Hongbin, S., Chengkang, Z., et al.: Short-term electrical load forecasting method based on stacked auto-encoding and GRU neural network. Evol. Intel. **12**, 385–394 (2019). https://doi.org/10.1007/s12065-018-00196-0
19. Islam, M.M., Lam, A., Fukuda, H., et al.: An intelligent shopping support robot: understanding shopping behavior from 2D skeleton data using GRU network. Robomech J. **6**, 18 (2019). https://doi.org/10.1186/s40648-019-0150-1
20. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
21. Boutaba, R., Salahuddin, M.A., Limam, N., et al.: A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. J. Internet Serv. Appl. **9**, 16 (2018). https://doi.org/10.1186/s13174-018-0087-2