Fuzzy Online Bagging Using Adaptive F-transform

Sergiy Popov[1], Iryna Pliss[1], Olexii Holovin[2], Oleh Zolotukhin[1] and Larysa Chala[1]

[1] *Kharkiv National University of Radio Electronics, Nauky av., 14, Kharkiv, 61166, Ukraine*
[2] *Central Scientific Research Institute of Armament and Military Equipment of the Armed Forces of Ukraine, Kyiv, 03049, Ukraine*

Abstract

The ensemble multi-model bagging approach is considered. We propose a nonlinear adaptive bagging procedure which applies F-transform in its adaptive form to the results of a traditional weighting. This leads to further decrease of ensemble errors with low extra computational and time cost. Metamodel architecture and corresponding optimal learning algorithms are presented in details. Simulation based on short-term electric load forecasting problem confirms theoretical results and shows a significant decrease of the forecasting error in comparison to a linear approach.

Keywords

Nonlinear bagging, adaptive ensemble, optimal learning, F-transform

1. Introduction

Currently, Computational Intelligence (CI) systems such as Artificial Neural Networks (ANNs), both deep (DNNs) and traditional – shallow (SNNs), Neuro-Fuzzy Systems (NFS), Neo-Fuzzy Systems, etc. have been widely used to solve many Data Mining problems. This success is explained by their universal approximating and extrapolating capabilities and the ability to learn, i.e. adjust their parameters based on the data obtained from the observed object during its operation. At the same time, quite often there is a problem of choosing a specific system or network that can best cope with the problem being solved. Although DNNs provide high quality solutions to the problem, but require very large volumes of training samples and a lot of time for their training. SNNs such as Radial Basis Function Neural Networks (RBFNs) are inferior to DNNs in terms of accuracy, but are able to learn online, i.e. solve Data Stream Mining problems. Neuro-Fuzzy and Neo-Fuzzy Systems can effectively process non-stationary signals, etc. Therefore, choosing a specific system is a non-trivial task and usually requires considerable experience of the researcher.

To overcome the problems of choosing a specific system for a specific task, the ensemble multi-model bagging approach [1-10] is quite often used, when the task is concurrently solved using an ensemble of systems functioning in parallel. Their output signals are somehow combined using a so-called metamodel which forms the optimal result. Usually, weighted averaging is used, where the weights are calculated by the metamodel itself. As a rule, these are batch procedures working in offline mode, although adaptive linear online approaches are known [4-6, 9, 10] for solving Data Stream Mining problems. Non-linear bagging procedures practically do not exist with a few, but still offline mode exceptions [2].

Therefore, it is expedient to develop an adaptive nonlinear bagging metamodel that would combine and generalize the ensemble members' processing results in online mode with high speed and accuracy.

ProfIT AI 2024: 4th International Workshop of IT-professionals on Artificial Intelligence (ProfIT AI 2024), September 25–27, 2024, Cambridge, MA, USA

serhii.popov@nure.ua (S. Popov); iryna.pliss@nure.ua (I. Pliss), a_a_golovin@ukr.net (O. Holovin); oleg.zolotukhin@nure.ua (O. Zolotukhin), larysa.chala@nure.ua (L. Chala)

0000-0002-1274-5830 (S. Popov); 0000-0001-7918-7362 (I. Pliss); 0000-0003-4662-4559 (O. Holovin); 0000-0002-0152-7600 (O. Zolotukhin); 0000-0002-9890-4790 (L. Chala)

CEUR Workshop Proceedings (CEUR-WS.org)

2. Architecture of the adaptive nonlinear bagging metamodel

The architecture of the adaptive nonlinear bagging metamodel is shown in Figure 1.

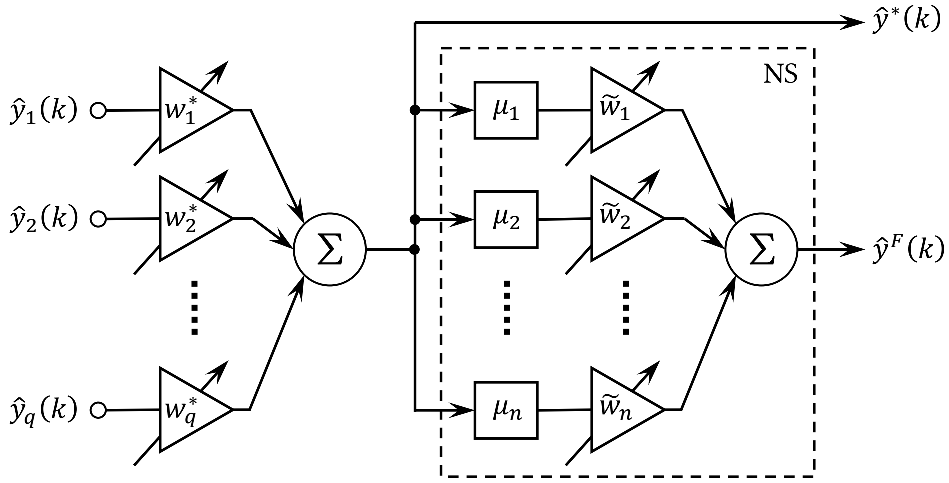

Figure 1: Adaptive nonlinear bagging metamodel architecture

It is readily seen that the metamodel's architecture is similar to F. Rosenblatt's elementary perceptron, but instead of a traditional activation function it contains a Nonlinear Synapse (NS) which is the main building block of a neo-fuzzy neuron [11-13] and implements the F-transform [14] in its adaptive version [15], i.e. it is essentially a universal approximator [16].

Output signals from q ensemble members which are solving the same problem $\hat{y}_1(k), \dots, \hat{y}_p(k), \dots, \hat{y}_q(k)$ (or in a vector form $\hat{y}(k) = \left(\hat{y}_1(k), \dots, \hat{y}_p(k), \dots, \hat{y}_q(k)\right)^T$, where $k = 1,2,\dots,N$ is the current discrete time index) are fed to the metamodel's inputs, then passed through adjustable synaptic weights $w_1^*, \dots, w_p^*, \dots, w_q^*$, and finally combined in the adder forming metamodel's intermediate output signal $\hat{y}^*(k)$ in the form

$$\hat{y}^*(k) = \sum_{p=1}^{q} w_p^* \hat{y}_p(k) \tag{1}$$

or in a vector form

$$\hat{y}^*(k) = \hat{y}^T(k) w^*, \tag{2}$$

where $w^* = \left(w_1^*, \dots, w_p^*, \dots, w_q^*\right)^T$.

The unbiasedness constraint is additionally imposed on the synaptic weights w^*

$$\sum_{p=1}^{q} w_p^* = I_q^T w^* = 1$$

(here I_q^T is a $(q \times 1)$ vector of ones). If we append inequality constraints on the non-negativity of the synaptic weights $0 \le w_p^* \le 1, \forall p$, these synaptic weights can be given the meaning of the degrees of membership of each of the signals $\hat{y}_p(k)$ to the optimal result.

Technically, the signal $\hat{y}^*(k)$ is already the solution of the optimization problem, however it can be improved by processing it in an adaptive Nonlinear Synapse (NS). NS is formed by n nonlinear membership functions $\mu_l\left(\hat{y}^*(k)\right), l = 1,2, \dots, n$. Traditional triangular constructions that satisfy the Ruspini partition of unity conditions are usually used, although it is possible to use more complex variants, e.g. B-splines, Gaussians, Epanechnikov kernels, etc. Each membership function output $\mu_l(\hat{y}^*(k))$ is multiplied by the corresponding adjustable weight $\tilde{w}_l$, and then summed in the second adder of the metamodel forming the output signal

$$\hat{y}^F(k) = \sum_{l=1}^{n} \tilde{w}_l \mu_l(\hat{y}^*(k)), \tag{3}$$

or in a vector form

$$\hat{y}^F(k) = \mu^T(\hat{y}^*(k))\tilde{w}, \tag{4}$$

where $\mu(\hat{y}^*(k)) = \left(\mu_1(\hat{y}^*(k)), \dots, \mu_l(\hat{y}^*(k)), \dots, \mu_n(\hat{y}^*(k))\right)^T, \tilde{w} = (\tilde{w}_1, \dots, \tilde{w}_l, \dots, \tilde{w}_n)^T$.

Combining (1)-(4), finally we can write

$$\hat{y}^F(k) = \sum_{l=1}^{n} \tilde{w}_l \mu_l \left(\sum_{p=1}^{q} w_p^* \hat{y}_p(k) \right),$$

or

$$\hat{y}^F(k) = \mu^T(\hat{y}^T(k)w^*)\tilde{w},$$

Where adjusted synaptic weights vectors w^* and $\tilde{w}$ are subject to online learning.

3. Adaptive nonlinear bagging metamodel learning

Synaptic weights vector w^* can be adjusted by gradient optimization of the learning criterion

$$E(k) = \frac{1}{2}\left(y(k) - \sum_{p=1}^{q} w_p^* \hat{y}_p(k) \right)^2 = \frac{1}{2}(y(k) - \hat{y}^T(k)w^*)^2$$

(here $y(k)$ – external reference signal) subject to constraints $I_q^T w^* = 1$. It is achieved by searching for the saddle point of the Lagrange function

$$L(w^*, \lambda) = \frac{1}{2}(y(k) - \hat{y}^T(k)w^*)^2 + \lambda\left(I_q^T w^* - 1\right),$$

where λ – undefined Lagrange multiplier.

The Arrow-Hurwicz procedure can be used to find the saddle point in the following form

$$\begin{cases} w^*(k) = w^*(k-1) - \eta_w(k)\nabla_{w^*}L(w^*, \lambda), \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)\, \partial L(w^*, \lambda)/\partial\, \lambda, \end{cases}$$

or

$$\begin{cases} w^*(k) = w^*(k-1) + \eta_w(k)\left(e(k)\hat{y}(k) - \lambda(k-1)I_q\right), \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)\left(I_q^T w^*(k) - 1\right), \end{cases} \tag{5}$$

where $\eta_w(k), \eta_\lambda(k)$ – learning step parameters, $e(k) = y(k) - \hat{y}^T(k)w^*(k-1)$ – learning error.

It is mathematically proven that signal $\hat{y}^*(k) = \hat{y}^T(k)w^*(k)$ in terms of accuracy is not inferior to any of $\hat{y}_p(k), p = 1,2, \dots, q$ at the metamodel input.

The learning process can be optimized in terms of speed by the appropriate selection of the learning step parameter $\eta_w(k)$. If the following option is chosen

$$\eta_w(k) = \frac{e(k)}{e(k)\|\hat{y}(k)\|^2 - \lambda(k-1)I_q^T\hat{y}(k)},$$

the learning algorithm (5) can be written in the form

$$\begin{cases} w^*(k) = w^*(k-1) + \dfrac{e(k)\big(e(k)\hat{y}(k) - \lambda(k-1)I_q\big)}{e(k)\|\hat{y}(k)\|^2 - \lambda(k-1)I_q^T\hat{y}(k)}, \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)\big(I_q^T w^*(k) - 1\big). \end{cases} \tag{6}$$

When $\lambda(k) = 0$, it completely coincides with the adaptive speed-optimal Kaczmarz-Widrow-Hoff algorithm [17-20].

As noted above, the synaptic weights at the inputs of the metamodel can be given the meaning of the degrees of membership of each of the input signals $\hat{y}_p(k)$ to the optimal signal, which should theoretically coincide with the reference signal $y(k)$. In this case, the learning task consists in the minimization of the criterion

$$E(k) = \frac{1}{2}\left(y(k) - \sum_{p=1}^{q} w_p^\mu \hat{y}_p(k)\right)^2 = \frac{1}{2}\left(y(k) - \hat{y}^T(k)w^\mu\right)^2$$

subject to constraints

$$I_q^T w^\mu = 1, \qquad 0 \le w_p^\mu \le 1, \forall p.$$

Introducing the Lagrange function

$$L(w^\mu, \lambda, \rho) = \frac{1}{2}\left(y(k) - \hat{y}^T(k)w^\mu\right)^2 + \lambda\big(I_q^T w^\mu - 1\big) - \rho^T w^\mu,$$

(here ρ – vector of non-negative indefinite Lagrange multipliers) and the Kuhn-Tucker system

$$\begin{cases} \nabla_{w^\mu} L(w^\mu, \lambda, \rho) = 0_q, \\ \partial L(w^\mu, \lambda, \rho)/\partial \lambda = 0, \\ \rho_p \ge 0, p = 1,2, \dots, q, \end{cases}$$

it is easy to write the Arrow-Hurwicz-Uzawa gradient procedure for finding the saddle point of the Lagrangian in the form

$$\begin{cases} w^\mu(k) = w^\mu(k-1) - \eta_w(k)\nabla_{w^\mu} L(w^\mu, \lambda, \rho), \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)\, \partial L(w^\mu, \lambda, \rho)/\partial \lambda, \\ \rho(k) = \big[\rho(k-1) - \eta_\rho(k)\nabla_\rho L(w^\mu, \lambda, \rho)\big]_+ \end{cases}$$

(here $[\cdot]_+$ is the projector on the positive orthant),
or

$$\begin{cases} w^\mu(k) = w^\mu(k-1) + \eta_w(k)\left(e(k)\hat{y}(k) - \lambda(k-1)I_q - \rho(k-1)\right), \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)\big(I_q^T w^\mu(k) - 1\big), \\ \rho(k) = \big[\rho(k-1) - \eta_\rho(k) w^\mu(k)\big]_+. \end{cases} \tag{7}$$

Similarly to (5) and (6), the learning algorithm (7) can also be optimized for speed. The optimized procedure has the following final form

$$\begin{cases} w^{\mu}(k) = w^{\mu}(k-1) + \dfrac{e(k)\left(e(k)\hat{y}(k) - \lambda(k-1)I_q - \rho(k-1)\right)}{e(k)\|\hat{y}(k)\|^2 - \lambda(k-1)I_q^T\hat{y}(k) + \rho^T(k-1)\hat{y}(k)}, \\ \lambda(k) = \lambda(k-1) + \eta_\lambda(k)\left(I_q^T w^{\mu}(k) - 1\right), \\ \rho(k) = \left[\rho(k-1) - \eta_\rho(k) w^{\mu}(k)\right]_+. \end{cases} \tag{8}$$

Thus, algorithms (6), (8) are designed for online adjustment of metamodel parameters w^* or w^{μ} and ensure high accuracy of the obtained results.

As already mentioned, triangular-shaped functions are usually used as membership functions in the nonlinear synapse NS:

$$\mu_l\left(\hat{y}^*(k)\right) = \begin{cases} \dfrac{\hat{y}^*(k) - c_{l-1}}{c_l - c_{l-1}}, \text{if } \hat{y}^*(k) \in [c_{l-1}, c_l], \\ \dfrac{c_{l+1} - \hat{y}^*(k)}{c_{l+1} - c_l}, \text{if } \hat{y}^*(k) \in [c_l, c_{l+1}], \\ 0 \text{ otherwise,} \end{cases}$$

where c_{l-1}, c_l, c_{l+1} – parameters of the centers of adjacent membership functions which are usually either uniformly distributed over the abscissa axis or can be found using clustering procedures [20, 21].

The main advantage of such functions is that at each learning cycle only two adjacent functions are activated and accordingly only two synaptic weights $\tilde{w}_{l-1}, \tilde{w}_l$ or $\tilde{w}_l, \tilde{w}_{l+1}$ are adjusted which simplifies and speeds up nonlinear synapse tuning process.

A standard quadratic criterion can be used to tune the nonlinear synapse:

$$\tilde{E}(k) = \frac{1}{2}\left(y(k) - \sum_{l=1}^{n} \tilde{w}_l \mu_l\left(\hat{y}^*(k)\right)\right)^2 = \frac{1}{2}\left(y(k) - \mu^T\left(\hat{y}^*(k)\right)\tilde{w}\right)^2,$$

which is minimized using a gradient procedure

$$\tilde{w}(k) = \tilde{w}(k-1) - \eta_{\tilde{w}}(k)\left(y(k) - \mu^T\left(\hat{y}^*(k)\right)\tilde{w}(k-1)\right)\mu\left(\hat{y}^*(k)\right), \tag{9}$$

where the step parameter $\eta_{\tilde{w}}(k)$ is chosen either using the Kaczmarz-Widrow-Hoff procedure [17, 19] or using other approaches [22, 23] which provide additional filtering properties of the learning process.

4. Simulation results

As a test case, we apply the proposed bagging approach to the short-term electric load forecasting problem (STLF), specifically 1-step ahead forecasting of daily electric load of one of regional power systems of Ukraine. We have the original series with $N = 337$ samples and $q = 6$ forecast series (337 samples each) generated by 6 different independent computational intelligence models. We treat the time series as a data stream, i.e. forecasting and metamodel operation is performed in online mode, therefore the whole dataset is processed only once (sample by sample, $k = 1, 2, \dots, N$) and there is no need to divide it into training, validation and test sets.

The original series (Figure 2) has several trends corresponding to different seasons, periodic (mostly weekly) patterns, sudden changes and outliers. Obviously, there is a strong random component, because electric load in large systems depends on many external factors, some of which have true random or chaotic nature, e.g. weather conditions [24]. So, the time series is nonstationary and noisy by its nature, hence its forecasting is quite challenging and usually different forecasting models/methods perform better than others on particular parts of the series and are inferior on other parts. One model/method is rarely better than all others on the whole series. It is exactly the case when bagging methods come into play and can improve overall forecasting accuracy attempting to take the best from all models/methods in the ensemble.

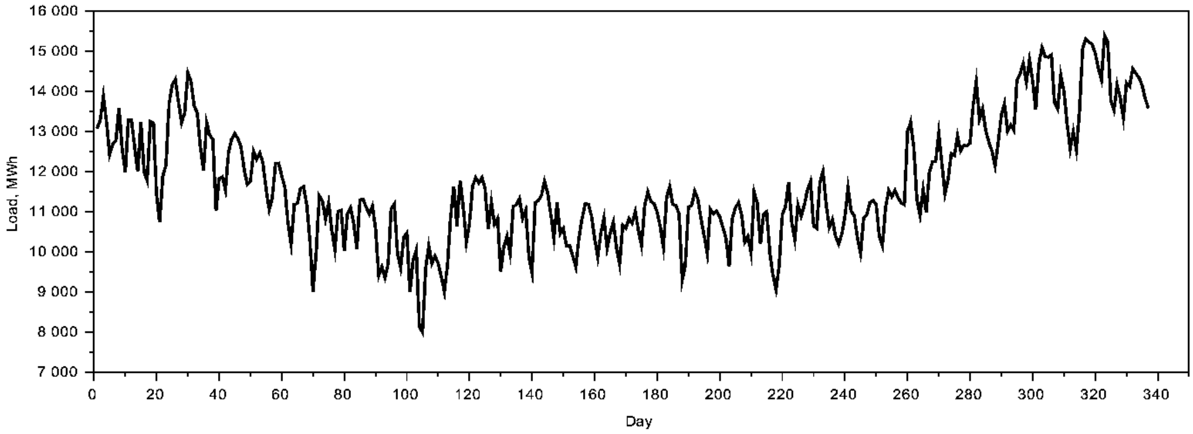

Figure 2: Daily electric load time series

We employ 6 specialized STLF models in the ensemble that have different inputs and structures. Such a diversity is aimed at capturing different properties of different parts of the series under consideration. Figure 3 shows last 30 days of the time series with the corresponding forecasts. We can see that long-term trends are more or less well captured by all models, but short-term changes pose a problem to all of them so that no single model is significantly better than the others.

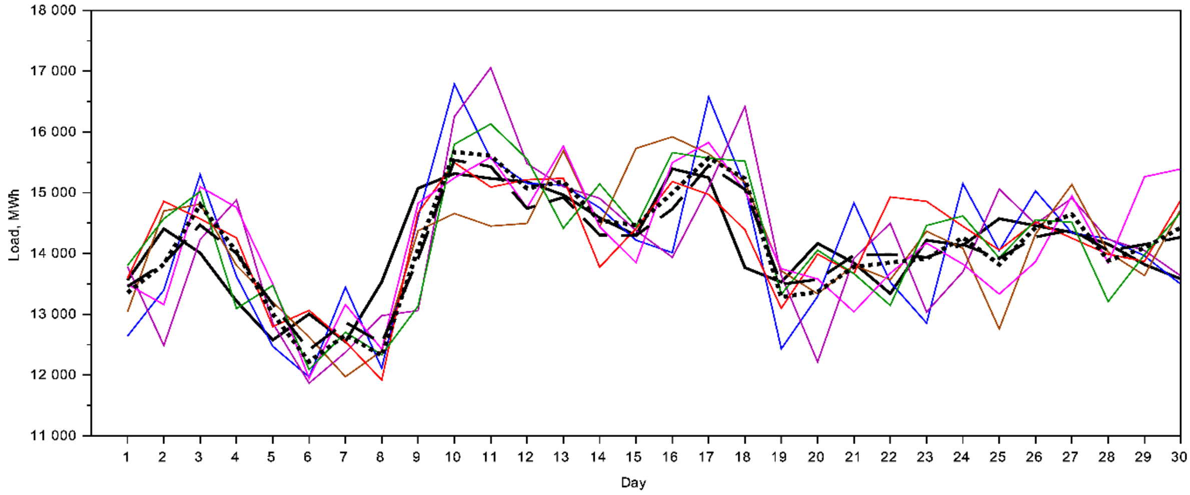

Figure 3: Forecasting results: true electric load (solid black line), 6 independent 1-day ahead forecasts (color lines), two metamodel forecasts: $\hat{y}^*(k)$ (dotted black line), $\hat{y}^F(k)$ (dashed black line).

We apply model (2) with algorithm (6) to obtain an optimal linear combination $\hat{y}^*(k)$ of the 6 forecasts from the ensemble member models. Just by a visual inspection of the plot it is obvious that $\hat{y}^*(k)$ is generally closer to the true series $y(k)$, which is also confirmed by corresponding errors comparison in Table 1. We employ Mean Absolute Percentage Error (MAPE) criterion that is widely used in short-term electric load forecasting research and has a clear physical sense.

Table 1
1-day ahead forecasting errors for all models and the ensemble outputs

Models	#1	#2	#3	#4	#5	#6	$\hat{y}^*(k)$	$\hat{y}^F(k)$
MAPE	6.8171%	7.5066%	4.8580%	5.0311%	4.8827%	5.1015%	**4.1440%**	**3.9595%**

The best of ensemble member models provides MAPE of 4.858%, which is reduced to 4.144% by the linear bagging procedure. Then we additionally apply to $\hat{y}^*(k)$ the adaptive F-transform (4) in order to exploit any possible remaining nonlinearities which cannot be approximated by the linear model (2). In this simple test case, the nonlinear synapse has 10 triangular membership functions $\mu_l(\hat{y}^*(k))$ whose centers c_l are uniformly distributed between the minimum and maximum values of the time series $y(k)$. NS parameters $\tilde{w}(k)$ are tuned by procedure (9). This additional nonlinear processing step further reduces the bagging error to 3.9595%, which is 1.23 times less than the lowest error provided by the best ensemble member alone.

The aforementioned processing steps can be summarized as a pseudo-code below.

Algorithm 1
Adaptive nonlinear bagging procedure performed on each time step k

```
Step 1. Receive input signals from $q$ ensemble members $\hat{y}_1(k), \dots, \hat{y}_q(k)$.
Step 2. Calculate the intermediate output signal $\hat{y}^*(k)$ as a linear combination of inputs (2).
Step 3. Apply adaptive F-transform (4) to obtain the output signal $\hat{y}^F(k)$.
Step 4. Update weights $w^*(k)$ using learning algorithm (6).
Step 5. Update NS parameters w˜(k) with procedure (9).
```

5. Conclusions

A fuzzy nonlinear online bagging procedure is proposed. It provides optimal results of the ensemble of computational intelligence systems for solving Data Stream Mining problems when the data are received for processing in real time and are non-stationary in nature. The proposed approach has a simple numerical implementation and high processing rate.

Simulations confirm theoretical results. Optimal linear combination provides errors lower than the lowest error among the ensemble member models. Nonlinear F-transform provides additional decrease of the error, overall by 1.23 times in comparison to the best model in the ensemble.

Future research on the topic would focus on fine tuning of the nonlinear synapse parameters (membership function types, their centers initialization and adaptation, etc.) and including other types of nonlinearities in the metamodel.

References

[1] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 126–140. https://doi.org/10.1007/BF00058655.

[2] J. H. Friedman, P. Hall, On bagging and nonlinear estimation, Journal of Statistical Planning and Inference 137.3 (2007) 669–683. https://doi.org/10.1016/j.jspi.2006.06.002.

[3] Z.-H. Zhou, J. Wu, W. Tang, Ensembling neural networks: Many could be better than all, Artificial Intelligence 137.1-2 (2002) 239–263. https://doi.org/10.1016/S0004-3702(02)00190-X.

[4] Ye. Bodyanskiy, P. Otto, I. Pliss, S. Popov, An Optimal Algorithm for Combining Multivariate Forecasts in Hybrid Systems, in: V. Palade, R.J. Howlett, L. Jain (Eds) Knowledge-Based Intelligent Information and Engineering Systems. KES 2003, volume 2774 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2003. https://doi.org/10.1007/978-3-540-45226-3_132.

[5] Ye. Bodyanskiy, S. Popov, Fuzzy Selection Mechanism for Multimodel Prediction. in M.G. Negoita, R.J. Howlett, L.C. Jain (Eds), Knowledge-Based Intelligent Information and Engineering

Systems. KES 2004, volume 3214 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2004, pp. 772–778. https://doi.org/10.1007/978-3-540-30133-2_101.

[6] A. Bifet, G. Holmes, B. Pfahringer, R. Gavaldà, Improving Adaptive Bagging Methods for Evolving Data Streams, in: ZH. Zhou, T. Washio (Eds), Advances in Machine Learning. ACML, volume 5828 of Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2009, pp. 23–37. https://doi.org/10.1007/978-3-642-05224-8_4.

[7] C. Zhang, Y. Ma (Eds) Ensemble Machine Learning. Methods and Applications, Springer New York, NY, 2012. https://doi.org/10.1007/978-1-4419-9326-7.

[8] D. Sarkar, V. Natarajan, Ensemble Machine Learning Cookbook, Packt Publishing, Birmingham, 2019.

[9] N. C. Oza, Online bagging and boosting, in Proc. 2005 IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, HI, USA, 2005, Vol. 3, pp. 2340-2345. https://doi.org/10.1109/ICSMC.2005.1571498.

[10] E. Lughofer, M. Pratama, I. Skrjanc, Online bagging of evolving fuzzy systems, Information Sciences 570 (2021) 16–33. https://doi.org/10.1016/j.ins.2021.04.041.

[11] T. Yamakawa, E. Uchino, T. Miki, H. Kusanagi, A neo fuzzy neuron and its applications to system identification and prediction of the system behavior, in Proc. 2nd Int. Conf. on Fuzzy Logic and Neural Networks, 1992, pp. 477-483.

[12] E. Uchino, T. Yamakava, Neo-fuzzy neuron based new approach to system modeling with application to actual system, in Proc. Sixth International Conference on Tools with Artificial Intelligence, New Orlean, LA, USA, 1994, pp.564-570.

[13] T. Miki, T. Yamakawa, Analog implementation of neo-fuzzy neuron and its on-board learning, in Computational Intelligence and Applications, WSES Press, Piraeus, 1999, pp. 144-149.

[14] I. Perfilieva, Fuzzy transforms: Theory and applications, Fuzzy Sets and Systems 157.8 (2006) 993-1023. https://doi.org/10.1016/j.fss.2005.11.012.

[15] Ye. Bodyanskiy, N. Teslenko, Adaptyvne F-peretvorennya na isnovi uzagal'nenoi regresiynoi neyronnoi merezhi, Adaptyvni systemy avtomatychnogo upravlinnya 10 (2007) 25–31. (In Ukrainian)

[16] Ye. Bodyanskiy**,** S. Kostiuk Neuron based on an adaptive fuzzy transform for modern artificial neural network models, International Scientific Technical Journal "Problems of Control and Informatics" 68.6 (2023) 94–105. https://doi.org/10.34229/1028-0979-2023-6-7.

[17] S. Kaczmarz, Angenaherte auflosung von systemen linearer gleichungen, Bulletin International de l'Academie Polonaise des Sciences et des Lettres 35 (1937) 355–357. (In German)

[18] S. Kaczmarz, Approximate solution of systems of linear equations, International Journal of Control 57.6 (1993) 1269–1271. https://doi.org/10.1080/00207179308934446.

[19] B. Widrow, M.E. Hoff, Adaptive switching circuits, in: 1960 IRE WESCON Convention Record, Part 4, IRE, New York,1960, pp. 96–104.

[20] T. Kohonen, Self-Organizing Maps, Springer-verlag, Berlin, 1995.

[21] Ye. Bodyanskiy, E. Vynokurova, A. Dolotov, Self-Learning Cascade Spiking Neural Network for Fuzzy Clustering Based on Group Method of Data Handling, Journal of Automation and Information Sciences 45.3 (2013) 23–33. https://doi.org/10.1615/JAutomatInfScien.v45.i3.30.

[22] P. Otto, Ye. Bodyanskiy, V. Kolodyazhniy, A new learning algorithm for a forecasting neuro-fuzzy network, Integrated Computer Aided Engineering 10 (2003) 399-409. https://doi.org/10.3233/ICA-2003-10409.

[23] Ye. Bodyanskiy, O. Vynokurova, I. Pliss, G. Setlak, P. Mulesa, Fast learning algorithm for deep evolving GMDH-SVM neural network in data stream mining tasks, in Proc. 2016 IEEE First International Conference on Data Stream Mining & Processing (DSMP), Lviv, Ukraine, 2016, pp. 257–262. https://doi.org//10.1109/DSMP.2016.7583555.

[24] Ye. Bodyanskiy, S. Popov, T. Rybalchenko, Feedforward neural network with a specialized architecture for estimation of the temperature influence on the electric load, in Proc. 2008 4th International IEEE Conference Intelligent Systems, Varna, Bulgaria, 2008, pp. 7-14–7-18. https://doi.org//10.1109/IS.2008.4670444.