Date of publication DEC-31, 2025, date of current version SEP-14, 2025.
www.computingonline.net / computing@computingonline.net

Print ISSN 1727-6209
Online ISSN 2312-5381
DOI 10.47839/ijc.24.4.4330

Fast Stacking Neuro-Neo-Fuzzy System for Inverse Modeling in Online Mode

YEVGENIY BODYANSKIY, OLEH ZOLOTUKHIN, ANDRIY YEROKHIN, MARYNA KUDRYAVTSEVA, MAKSYM YEROKHIN

Kharkiv National University of Radio Electronics, Kharkiv, Ukraine, 61166

Corresponding author: Yevgeniy Bodyanskiy (e-mail: yevgeniy.bodyanskiy@nure.ua), Oleh Zolotukhin (e-mail: oleg.zolotukhin@nure.ua), Andriy Yerokhin (e-mail: andriy.yerokhin@nure.ua), Maryna Kudryavtseva (e-mail: maryna.kudryavtseva@nure.ua), Maksym Yerokhin (e-mail: maksym.ierokhin@nure.ua).

This paper is part of the DIOR project that has received funding from the European Union's MSCA RISE programme under grant agreement No. 10100828.

ABSTRACT The article is devoted to the development of an approach based on hybrid systems of computational intelligence to solve Data Stream Mining tasks for complex technical objects. Such tasks arise during the identification, control and diagnostics of highly dynamic technical objects, for example, fiber-optic communication systems with rapidly growing traffic volumes. Often deep neural networks are used to solve such problems that require large amounts of training samples, a lot of time for their tuning and leads to the impossibility of effectively solving tasks in online mode with inverse modeling tasks being particularly complex. A new architecture of a stacking neuro-neo-fuzzy system is proposed that solves inverse modeling tasks, operates in online mode, has improved approximation capabilities, high speed and is characterized by simple computational implementation. The proposed stacking neuro-neo-fuzzy system is designed to solve the problem of inverse modeling of fiber optic communication systems in real time mode under conditions of limited training samples.

KEYWORDS artificial neural networks; artificial intelligence; computational intelligence; data visualization; inverse modeling; neuro-neo-fuzzy system; stacking hybrid systems; online adaptive learning; online ensemble learning; Python programming.

I. INTRODUCTION AND STATEMENT OF THE PROBLEM

Nowadays, artificial neural networks (ANN) have become widely used to solve a large class of information processing problems of a diverse nature and, above all, intelligent data analysis (Data Mining), including pattern recognition – classification, clustering, association, approximation, filtering, smoothing, prediction-extrapolation, optimization, processing and generation of natural language texts and videos, etc.

Currently, deep neural networks (DNN) are considered the most effective for solving these problems which really show impressive results, but require large amounts of training samples (which are not always available in real problems) and a lot of time for their setup and which takes place in multiepoch learning mode. Therefore, to solve the problems of Data Stream Mining, DNNs require the use of powerful computing equipment which is not always available for specific applications.

A separate important direction in the use of ANN is the identification, control and diagnostics of technical plants, which must be solved online in real time during the operation of the controlled object, information from which comes in the form of a data stream which in turn can be contaminated by disturbances of various nature. Here, the tasks of inverse modeling, which is used in some adaptive control systems of technical objects [1, 2] are particularly difficult. These tasks consist in the fact that for the object of control

$$y = f(x), \tag{1}$$

where y - $(m\times 1)$ – vector of the object's output signals,

x – $(n\times 1)$ – vector of input signals,

$f(\cdot)$ – some a priori unknown nonlinear operator that is subject to recovery in the identification process.

It is necessary to construct an inverse transformation

$$\hat{x} = f^{-1}(y), \tag{2}$$

however, since usually in real situations $m<n$, from a mathematical point of view this problem is incorrect.

Figure 1 shows the inverse modeling scheme.

Here the inverse modeling error $e(k) = x(k) - \hat{x}(k)$ is used to tune the inverse model which is most often realised by some artificial neural networks due to their approximating properties.

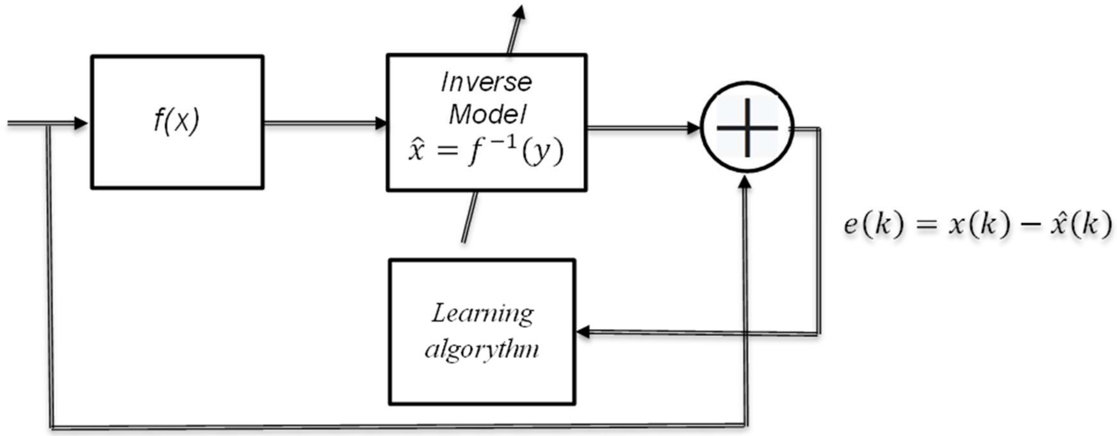


Figure 1. Inverse modeling scheme.

A striking example of effective inverse modeling is the fast neural network inverse model to maximize throughput in ultra wide-band wavelength-division multiplexed systems [3] where a four-layer neural network of the multilayer perceptron type with sigmoidal activation functions, containing several hundred thousand synaptic weights and trained on the basis of error backpropagation. This model has demonstrated high accuracy, but required 1300 epochs in multiepoch learning mode for its tuning. Thus, if really high speed is required to solve inverse modeling problems, multilayer neural networks (especially deep ones) are not always effective.

In our opinion, hybrid systems of computational intelligence (HSCI) especially neuro-fuzzy systems [4-11], may be the most suitable for such problems.

II. MATERIAL AND METHODS

A. ARCHITECTURE OF STACKING NEURO-NEO-FUZZY SYSTEM FOR INVERSE MODELING

Currently, HSCI have become widely used to solve a large class of problems processing data arrays of various nature, given both in the form of object-property tables and in the form of multidimensional time series.

Among those systems, neuro-fuzzy systems (NFS) [5, 6, 8, 11-13] have become the most widespread, due to their high approximating properties, simplicity of numerical implementation, and the possibility of online real-time learning. The tasks that are currently successfully solved using NFS include, first of all, prediction-forecasting, filtering, emulation-identification, and adaptive control of non-stationary, essentially nonlinear stochastic and chaotic objects. In terms of approximating properties, these systems are inferior to deep neural networks, but in the conditions of the need to process information in online mode and short training samples, NFS undoubtedly have advantages.

Therefore, in our opinion, it is advisable to use NFS in inverse modeling tasks, while to improve their approximating properties which is especially important in modeling tasks, to develop a hybrid system that would combine the advantages of both neuro-fuzzy approaches [14, 15] which is characterized by a high learning rate and effective approximating properties based on the F-transform [16] in its online version [17]. The hybridization of these two approaches has proven its effectiveness for solving pattern recognition problems [18].

Fig.2 shows the architecture of stacking neuro-neo-fuzzy system for solving inverse modeling tasks.

The first layer of the system is the fuzzification one formed by hm nonlinear membership functions (h membership functions for each input $y_j(k)$, here $k = 1,2,...$ – the observation number in the training sample or the current discrete time) which are usually used as Gaussians in the form

$$\varphi_{lj}(y_j, c_{lj}\sigma_{lj}) = \exp\left(-\frac{(y_j - c_{lj})^2}{2\sigma_{lj}^2}\right)$$
$$\forall l = 1,2,\dots,h;\ j = 1,2,\dots,m, \quad (3)$$

where c_{lj} – the parameter of the center of the membership function,

σ_{lj} – the parameter of its width.

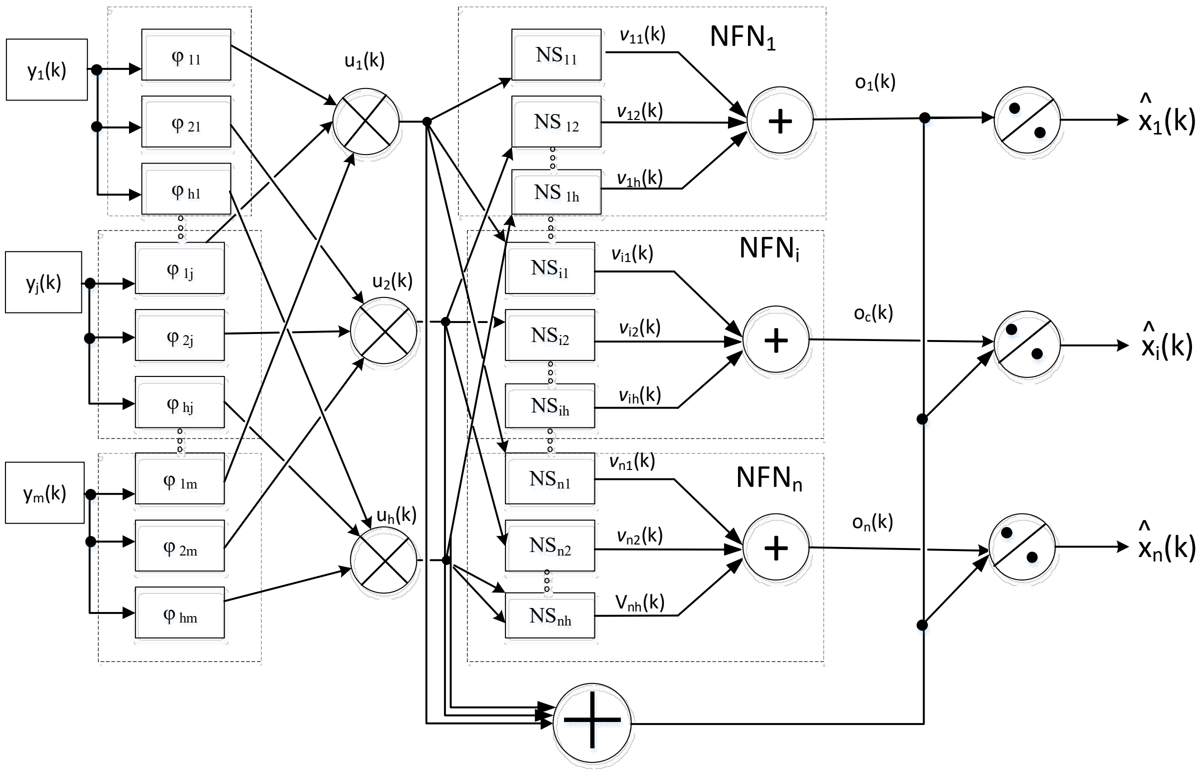
In most NFSs known for today, the centers c_{lj} are located evenly along the abscissa axes, while all input signals are previously encoded into some constraint interval $y_{min} \leq y_j(k) \leq y_{max}$. The parameters that define the width of the membership function σ_{lj}, are also given a priori and are assumed to be the same for all membership functions $\sigma_{lj} = \sigma$, although in principle both the centers and the width parameters can be adjusted during the training process of the system [5].

The second hidden layer of the system is an aggregation layer formed by h elementary multiplication blocks and forms multidimensional Gaussians in the form

$$\varphi_l\,(y, c_l, \sigma_l) = \prod_{j=1}^{m} \varphi_{lj}(y_j, c_{lj}, \sigma_{lj}) = \prod_{j=1}^{m} \exp\left(-\frac{(y_j - c_{lj})^2}{2\sigma_{lj}^2}\right) = \exp\left(-\frac{\| (y - c_l \|^2}{2\sigma_l^2}\right), \quad (4)$$

where $y = (y_1, \dots y_j, \dots, y_m)^T$, $c_l = (c_{l1}, \dots c_{lj}, \dots, c_{lm})^T$

In fact, the fuzzification-aggregation layers of almost all NFSs form a set of multidimensional Gaussians which completely coincides with the first layer of popular radial basis



function neural networks (RBFNs) with kernel activation functions and, like multilayer perceptrons, are universal approximators.

Figure 2. Stacking Neuro-Neo-Fuzzy System for inverse modeling.

At the same time, if classical RBFNs suffer from the undesirable effect of the "curse of dimensionality", NFSs are protected from this phenomenon due to a special partitioning of the input space.

Thus, as a result of fuzzification-aggregation operations, a set of signals is formed at the outputs of the second layer

$$u_l(k) = \exp\left(-\frac{\|y(k)-c_l\|^2}{2\sigma^2}\right), l = 1,2,\dots,h, \quad (5)$$

which is fed to the third hidden layer where the system training and tuning actually takes place.

Thus, in the popular Wang-Mendel NFS, the third layer is formed by nh tunable synaptic weights which solve the piece-wise approximation problem.

It is clear that to achieve the required accuracy, the number of tuning parameters must be sufficiently large.

In the widely used Takagi-Sugeno-Kang NFS, the third layer is essentially a generator of polynomial functions, the order of which is chosen quite arbitrarily, i.e. it implements polynomial approximation, and the number of tuning parameters is determined by the chosen order of the polynomial.

To improve the approximating properties of the proposed neuro-neo-fuzzy system (NNFS), nonlinear synapses (NS) are used, which are universal approximators of one-dimensional functions of arbitrary type [16] and constituent elements of neo-fuzzy neurons (NFN) [14, 15].

The architecture of the nonlinear synapse NS_{il} of the proposed NNFS is shown in Fig. 3.

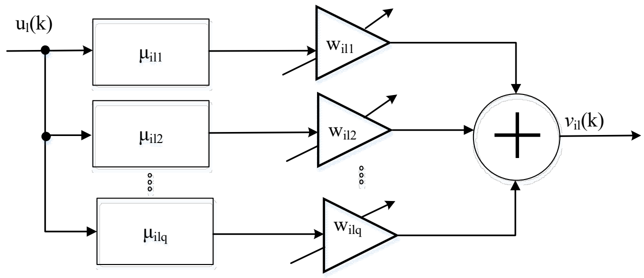


Figure 3. Nonlinear Synapse of the third layer of NNFS.

Each nonlinear synapse of the proposed NNFS, the total number of which is defined as nh, implements a nonlinear transformation

$$v_{il}(k) = \sum_{p=1}^{q} \mu_{ilp}\big(u_l(k)\big) w_{ilp}(k-1), \quad (6)$$

where $\mu_{ilp}\big(u_l(k)\big)$ – nonlinear membership functions, usually triangular, satisfying the Ruspini unity partitioning conditions, $w_{ilp}(k-1)$ – tuning synaptic weights obtained as a result of processing of previous k-1 observations x(k-1), y(k-1) from the controlled object.

The advantage of these functions is that at each time k of the system tuning, only two neighboring membership functions are fired, i.e., comparatively to the Wang-Mendel system, only two from many synaptic weights are tuned at each time which is a small "payment" for the universal approximating properties. To improve the approximating properties of nonlinear synapses, we propose to use parabolic Epanechnikov kernel functions instead of traditional triangular membership functions [19]:

$$\mu_{ilp}(u_l) = \left[1 - \frac{\left(c_{ilp} - u_l\right)^2}{r^2}\right]_+, \quad (7)$$

here $[\,\cdot\,]_+ = max\{0,\cdot\}$, the form of which is shown in fig.4.

It is clear that with this arrangement of membership functions, the number of tuning parameters at each training cycle is *2nh*.

Each of the NNFS outputs $\hat{x}_i,\ i = 1,2,\dots,n$, is associated with h nonlinear synapses NS_{il}, the outputs of which are summed, thus forming n neo-fuzzy neurons NFN_i which have proven their high efficiency for solving many real-world problems of prediction, filtering, and approximation [14].

And, finally, the fourth – the output layer of the proposed NNFS which is formed by a single adder and n division blocks, implements the defuzzification operation standard for all neuro-fuzzy systems.

$$\hat{x}_i(k) = \frac{\sum_{l=1}^{h} v_{il}(k)}{\sum_{l=1}^{h} u_l(k)} = \frac{o_i(k)}{\sum_{l=1}^{h} u_l(k)} = \frac{o_i(k)}{u(k)}, \quad (8)$$

which results in the formation of an estimate $\hat{x}_i(k)$ of the input signal of the real object $x_i(k)$.

Thus, the proposed system is a stacking system, since its first layers are formed by neuro-fuzzy elements, and the subsequent layers are neo-fuzzy systems which provide this system with high approximating properties and real-time online learning capabilities.

B. ONLINE LEARNING OF STACKING NEURO-NEO-FUZZY SYSTEM IN INVERSE MODELING TASK

The training of the proposed NNFS for solving the inverse modeling problem is implemented by online optimization of the quadratic criterion traditional in the theory of identification

$$E(k) = \sum_{i=1}^{n} E_i(k) = \sum_{i=1}^{n} e_i^2(k) = \sum_{i=1}^{n}\big(x_i(k) - \\ - \hat{x}_i(k)\big)^2, \quad (9)$$

while this problem for each of the outputs of the system $\hat{x}_i(k)$ can be solved separately, taking into account the fact that

$$\hat{x}_i(k) = \frac{o_i(k)}{u(k)} = \frac{\sum_{l=1}^{h} v_{il}(k)}{\sum_{l=1}^{h} u_l(k)}, \quad (10)$$

Introducing the vector of membership function values at the на *k*-th learning cycle

$$\mu_i(k) = (\mu_{i11}(u_1(k)), \mu_{i12}(u_1(k)), \dots, \\ , \dots, \mu_{i1q}(u_1(k)), \mu_{i21}(u_2(k)), \dots, \\ \dots, \mu_{ilp}(u_l(k)), \dots, \mu_{ihq}(u_h(k)))^T, \quad (11)$$

and a vector of the same dimensionality of tuning synaptic weights calculated at the previous tuning cycle

$$w_i(k-1) = (w_{i11}(k-1), w_{i12}(k-1), \dots, \\ , \dots, w_{i1q}(k-1), w_{i21}(k-1), \dots, \\ \dots, w_{i2q}(k-1), w_{ilp}(k-1), \dots, w_{ihq}(k-1))^T, \quad (12)$$

we can write the equation of the tuning model in the form:

$$o_i(k) = w_i^T(k-1)\mu_i(k), \quad (13)$$

$$\hat{x}_i(k) = w_i^T(k-1)\frac{\mu_i(k)}{u(k)} = w_i^T(k-1)\,\tilde{\mu}_i(k), \quad (14)$$

and the training error

$$e_i(k) = x_i(k) - w_i^T(k-1)\,\tilde{\mu}_i(k). \quad (15)$$

To tune the parameters vector of this model in online mode, it is advisable to use an adaptive learning algorithm for a neuro-fuzzy system [20]

$$\begin{cases} w_i(k) = w_i(k-1) + r^{-1}(k)\big(x_i(k) - w_i^T(k-1)\,\tilde{\mu}_i(k)\big)\,\tilde{\mu}_i(k), \\ r(k) = \alpha r(k-1) + ||\tilde{\mu}_i(k)||^2 \end{cases}, \quad (16)$$

where $0 \le \alpha \le 1$ – smoothing parameter, which provides filtering properties of the learning process if the source data are contaminated with disturbances that are always present in real control objects.

It is easy to see that when $\alpha = 0$ we arrive at the Kaczmarz adaptive algorithm, popular in identification theory [21]

$$w_i(k) = w_i(k-1) + \frac{x_i(k) - w_i^T(k-1)\,\tilde{\mu}_i(k)}{||\tilde{\mu}_i(k)||^2}\,\tilde{\mu}_i(k) = \\ = w_i(k-1) + (x_i(k) - \\ = w_i^T(k-1)\,\tilde{\mu}_i(k)\,\tilde{\mu}_i^{+T}(k), \quad (17)$$

here $(\cdot)^+$ – symbol of the pseudoinversion operation) which is optimal in terms of rate of convergence in the class of gradient adaptive identification algorithms.

III. RESULTS

The computational experiment was performed using the Python programming language and aims to evaluate the performance of the proposed fast stacking neuro-neo-fuzzy system in solving inverse modeling tasks. The objectives include:

– assessing the approximation accuracy;

– evaluating the learning speed in online mode;

– comparing the NNFS with traditional inverse modeling approaches, such as deep neural networks and classical neuro-fuzzy systems.

The experiment used synthetic and real-world datasets HardFailure and SoftFailure that represent the inverse modeling problem in technical systems [22-24]. The datasets include:

– a synthetic nonlinear function with known inverse properties for validation;

– a dataset from fiber-optic communication systems where inverse modeling is required to optimize signal transmission.

The hardware required to conduct the experiment includes: Intel Core i7, 16GB RAM, NVIDIA RTX 3060 GPU. The software required to conduct the experiment includes: Python with TensorFlow and SciPy libraries.

Let's look at the evaluation metrics:

– Mean Squared Error (MSE) – to measure approximation accuracy.

– Root Mean Squared Error (RMSE) – to compare the magnitude of errors.

– Convergence Speed, measured in the number of iterations required to reach a predefined accuracy threshold.

Let's move on to experiment implementation. It contains three procedures:

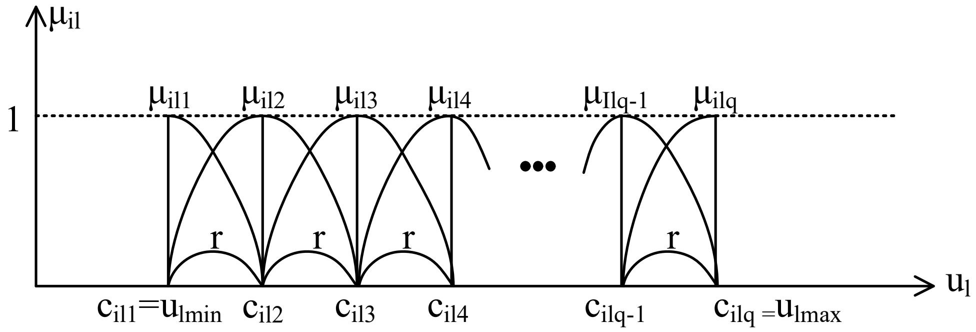


Figure 4. Epanechnikov membership function.

– training process;
– comparison with other models;
– results and analysis.

The NNFS was trained in an online mode using the adaptive learning algorithm [25, 26]:

1. Initialize parameters of membership functions and nonlinear synapses.
2. For each incoming data point:

– compute fuzzification and aggregation layer outputs;
– update synaptic weights using the adaptive rule;
– evaluate the inverse modeling error;
– adjust parameters using the Kaczmarz adaptive algorithm.

The NNFS is compared with [27-29]:

– a deep neural network (DNN) trained using backpropagation;

– a classical Takagi-Sugeno-Kang (TSK) neuro-fuzzy system.

Each model was trained on the same dataset, and performance is measured across multiple trials.

The results are presented in the tables: Table 1 Approximation Accuracy, Table 2 Learning Speed.

Table 1. Approximation Accuracy

Model	MSE (Synthetic)	MSE (Real-World)
NNFS	**0.0021**	**0.0035**
DNN	0.0045	0.0068
TSK	0.0052	0.0073

Table 2. Learning Speed

Model	Training Time per Sample (ms)	Convergence Iterations
NNFS	**0.32**	**1000**
DNN	1.45	1300
TSK	0.58	2000

Based on the conducted experiment the following conclusions can be drawn:

– the NNFS achieved the lowest MSE, indicating superior approximation capabilities.

– the NNFS demonstrated the fastest training per sample, making it suitable for real-time applications.

– the Kaczmarz-based adaptive learning significantly reduced the number of iterations needed for convergence.

IV. DISCUSSION

A fast stacking adaptive neuro-fuzzy system is proposed to solve the problem of inverse modeling of various control and management objects in online real-time mode.

This system has improved approximation properties compared to traditional neuro-fuzzy systems, is characterized by high learning speed and simplicity of numerical implementation. The results of numerical modeling confirm the correctness of the obtained theoretical results.

The proposed fast NNFS outperforms traditional deep neural networks and classical neuro-fuzzy systems in inverse modeling tasks [30-32]. It offers:

– high approximation accuracy;
– fast online learning;
– simple numerical implementation.

These advantages make NNFS a promising solution for real-time inverse modeling in dynamic technical systems.

V. CONCLUSIONS

The stacking neuro-neo-fuzzy system is proposed which is designed to solve the problem of inverse modeling of fiber optic communication systems in real time mode under conditions of limited training samples. The system combines the advantages of neuro-fuzzy systems and neo-fuzzy systems, which are the stacks that form it, and is characterized by high accuracy, speed, and simplicity of numerical implementation. It can be effectively used to solve problems of identification and adaptive control of technical objects for various purposes in real time mode.

REFERENCES

[1] J. Kufel et al., "What is machine learning, artificial neural networks and deep learning?—Examples of practical applications in medicine," *Diagnostics*, vol. 13, no. 15, Art. no. 2582, 2023. https://doi.org/10.3390/diagnostics13152582.

[2] M. G. M. Abdolrasol et al., "Artificial neural networks based optimization techniques: A review," *Electronics*, vol. 10, no. 21, Art. no. 2689, 2021, https://doi.org/10.3390/electronics10212689.

[3] F. Aguirre et al., "Hardware implementation of memristor-based artificial neural networks," *Nature Communications*, vol. 15, Art. no. 1974, 2024, https://doi.org/10.1038/s41467-024-45670-9.

[4] A. Goel, A. K. Goel, and A. Kumar, "The role of artificial neural network and machine learning in utilizing spatial information," *Spatial Information Research*, vol. 31, pp. 275–285, 2023, https://doi.org/10.1007/s41324-022-00494-x.

[5] M. S. Roza Dastres and M. S. Soori, "Artificial neural network systems," *International Journal of Imaging and Robotics*, vol. 21, no. 2, pp. 13–25, 2021.

[6] L. Messeri and M. J. Crockett, "Artificial intelligence and illusions of understanding in scientific research," *Nature*, vol. 627, pp. 49–58, 2024, https://doi.org/10.1038/s41586-024-07146-0.

[7] S.-C. Vanegas-Ayala, J. Barón-Velandia, and E. Romero-Riaño, "Systematic review of forecasting models using evolving fuzzy systems," *Computation*, vol. 12, no. 8, Art. no. 159, 2024, https://doi.org/10.3390/computation12080159.

[8] M. E. Dogan, T. Goru Dogan, and A. Bozkurt, "The use of artificial intelligence (AI) in online learning and distance education processes: A systematic review of empirical studies," *Applied Sciences*, vol. 13, no. 5, Art. no. 3056, 2023, https://doi.org/10.3390/app13053056.

[9] S. Zhang et al., "Online adaptive CNN: A session-to-session transfer learning approach for non-stationary EEG," *Proceedings of the IEEE Symposium Series on Computational Intelligence (SSCI)*, Singapore, 2022, pp. 164–170, https://doi.org/10.1109/SSCI51031.2022.10022274.

[10] I. Gligorea et al., "Adaptive learning using artificial intelligence in e-learning: A literature review," Education Sciences, vol. 13, no. 12, Art. no. 1216, 2023, https://doi.org/10.3390/educsci13121216.

[11] P. V. C. Souza, "Fuzzy neural networks and neuro-fuzzy networks: A review of the main techniques and applications used in the literature," *Applied Soft Computing*, vol. 92, Art. no. 106275, 2020, https://doi.org/10.1016/j.asoc.2020.106275.

[12] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-15, no. 1, pp. 116–132, 1985. https://doi.org/10.1109/TSMC.1985.6313399.

[13] J.-S. Jang, C.-N. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, New York, NY, USA: Prentice Hall, 1977.

[14] E. Uchino and T. Yamakawa, "Soft computing based signal prediction, restoration, and filtering," in *Intelligent Hybrid Systems*, D. Ruan, Ed. Boston, MA, USA: Springer, 1997, pp. 331–349, https://doi.org/10.1007/978-1-4615-6191-0_14.

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

[16] I. Perfilieva, "Fuzzy transforms: Theory and applications," *Fuzzy Sets and Systems*, vol. 157, no. 8, pp. 993–1023, 2006. https://doi.org/10.1016/j.fss.2005.11.012.

[17] Ye. V. Bodyanskiy and N. O. Teslenko, "Adaptive F-transform using general regression neural networks," *Adaptive Systems of Automatic Control*, no. 10 (30), pp. 25–31, 2007.

[18] S. Popov, "Nonlinear visualization of incomplete data sets," in *Computer Science – Theory and Applications*, D. Grigoriev, J. Harrison, and E. A. Hirsch, Eds. Berlin, Heidelberg: Springer, 2006, pp. 476–484, https://doi.org/10.1007/11753728_53.

[19] V. A. Epanechnikov, "Non-parametric estimation of a multivariate probability density," *Theory of Probability and its Applications*, vol. 14, no. 1, pp. 153–158, 1969. https://doi.org/10.1137/1114019.

[20] Ye. Bodyanskiy, V. Kolodyazhniy, and A. Stephan, "An adaptive learning algorithm for a neuro-fuzzy network," in *Computational Intelligence: Theory and Application*, B. Reusch, Ed. Berlin, Heidelberg, New York, NY, USA: Springer, 2001, pp. 68–75. https://doi.org/10.1007/3-540-45493-4_11.

[21] S. Kaczmarz, "Approximate solution of systems of linear equations," *International Journal of Control*, vol. 53, no. 6, pp. 1269–1271, 1993. https://doi.org/10.1080/00207179308934446.

[22] V. Filatov, O. Zolotukhin, A. Yerokhin, and M. Kudryavtseva, "The methods for the prediction of climate control indicators in Internet of Things systems," in CEUR Workshop Proceedings, 2021, pp. 391–400, https://doi.org/10.5281/zenodo.14526027.

[23] S. Popov, "Large-scale data visualization with missing values," *Technological and Economic Development of Economy*, vol. 12, no. 1, pp. 44–49, 2006, https://doi.org/10.3846/13928619.2006.9637721.

[24] J.-S. R. Jang, "ANFIS: Adaptive-network-based fuzzy inference system," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp. 665–685, 1993, https://doi.org/10.1109/21.256541.

[25] A. Abraham, "Neuro-fuzzy systems: State-of-the-art modeling techniques," in *Computational Intelligence, IntechOpen*, 2004, pp. 1–24, https://doi.org/10.5772/5555.

[26] S. Vieira, U. Kaymak, and J. M. C. Sousa, "Fuzzy neural networks and neuro-fuzzy networks: A review of the main techniques and applications used in the literature," *Applied Soft Computing*, vol. 92, art. no. 106275, 2020, https://doi.org/10.1016/j.asoc.2020.106275.

[27] E. Lughofer, *Evolving Fuzzy Systems: Methodologies, Advanced Concepts and Applications*, Heidelberg, Germany: Springer, 2011, https://doi.org/10.1007/978-3-642-18087-3.

[28] N. Kasabov, *Evolving Connectionist Systems: The Knowledge Engineering Approach*, 2nd ed. London, U.K.: Springer, 2007.

[29] J. Kacprzyk and S. Zadrożny, "A hybrid systematic review approach on complexity issues in data-driven fuzzy inference systems development," *Informatica*, vol. 29, no. 4, pp. 595–613, 2018.

[30] G. Bucci, F. Ciancetta, E. Fiorucci, S. Mari, and A. Fioravanti, "Measurements for non-intrusive load monitoring through machine learning approaches," *ACTA IMEKO*, vol. 10, no. 4, pp. 90–96, 2021, https://doi.org/10.21014/acta_imeko.v10i4.1184.

[31] V. Golovko, Y. Savitsky, T. Laopoulos, A. Sachenko, and L. Grandinetti, "Technique of learning rate estimation for efficient training of MLP," *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks (IJCNN)*, Como, Italy, 2000, vol. 1, pp. 323–328, https://doi.org/10.1109/IJCNN.2000.857856.

[32] R. Prasad and P. K. Shukla, "Indeterminacy handling of adaptive neuro-fuzzy inference system using neutrosophic set theory: A case study for the classification of diabetes mellitus," *International Journal of Intelligent Systems and Applications*, vol. 15, no. 3, pp. 1–15, 2023, https://doi.org/10.5815/ijisa.2023.03.01.

YEVGENIY BODYANSKIY Education: Kharkiv National University of Radio Electronics, Automatic and Remote Control. Current position: Professor of Artificial Intelligence Department at the Kharkiv National University of Radio Electronics, Dr.Sc., Professor. Areas of scientific interests: computational intelligence, neural networks, deep learning.

***OLEH ZOLOTUKHIN** Education: Kharkiv National University of Radio Electronics, Artificial Intelligence Systems, Computer System Analyst. Current position: Acting Dean of Computer Science Faculty, Head of Artificial Intelligence Department at the Kharkiv National University of Radio Electronics, PhD, Associated Professor. Areas of scientific interests: artificial intelligence, machine learning, deep learning.*

***ANDRIY YEROKHIN** Education: Kharkiv National University of Radio Electronics, Designing and manufacturing of radio equipment. Current position: First Vice Rector atf the Kharkiv National University of Radio Electronics, Dr.Sc., Professor. Areas of scientific interests: artificial intelligence, fiber optics data transfer.*

***MARYNA KUDRYAVTSEVA** Education: Kharkiv National University of Radio Electronics, Information Control Systems and Technologies, Computer System Analyst. Current position: PhD, Professor of Artificial Intelligence Department at the Kharkiv National University of Radio Electronics. Areas of scientific interests: generative artificial intelligence, AR/VR, neural networks.*

***MAKSYM YEROKHIN** Education: University of Tartu, Estonia, Computer Science. Current position: PhD Student, Assistant Professor of Artificial Intelligence Department at the Kharkiv National University of Radio Electronics. Areas of scientific interests: software engineering, data science.*

.