UDC 004.896

SYNTHESIS OF NEURAL NETWORK MODELS FOR TECHNICAL DIAGNOSTICS OF NONLINEAR SYSTEMS

Leoshchenko S. D. – PhD, Associate Professor of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

Oliinyk A. O. – Dr. Sc., Professor, Professor of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

Subbotin S. A. – Dr. Sc., Professor, Head of the Department of Software Tools, National University "Zaporizhzhia Polytechnic", Zaporizhzhia, Ukraine.

Morklyanyk B. V. – Dr. Sc., Professor, Professor of the Department of Information Technology, Military Academy, Kyiv, Ukraine.

ABSTRACT

Context. The problem of synthesizing a diagnostic model of complex technical processes in nonlinear systems, which should be characterized by a high level of accuracy, is considered. The object of research is the process of synthesizing a neural network model for technical diagnostics of nonlinear systems.

Objective of the work is to synthesize a high-precision neural network model based on previously accumulated historical data about the system.

Method. It is proposed to use artificial neural networks for modeling nonlinear technical systems. First, you need to perform an overall assessment of the complexity of the task. Based on the assessment, a decision can be made on the best approach to organizing neuromodel synthesis. So, for the task, the level of 'random complexity' was chosen, because despite the relative structure of the data, their total array is quite large in volume and requires careful study in order to ensure high quality of the solution. Therefore, in the future, it was proposed to use a neuromodel based on recurrent networks of the GRU topology and use swarm intelligence methods for neurosynthesis, in particular the A3C method. The results obtained showed a high level of solution obtained, but due to the high level of resource intensity, the proposed approach requires further modifications.

Results. A diagnostic model of complex technical processes in nonlinear systems of optimal topology, characterized by a high level of accuracy, is obtained. The built neuromodel reduces the risks associated with ensuring human safety.

Conclusions. The conducted experiments confirmed the operability of the proposed approach and allow us to recommend it for further refinement in order to implement technical, industrial and operational process control systems in practice in automation systems. Prospects for further research may lie in optimizing the resource intensity of synthesis processes.

KEYWORDS: technical diagnostics, nonlinear systems, machine learning, neural network synthesis, indicator system, neuromodel, sampling, learning, error.

ABBREVIATIONS

A3C is an Asynchronous Advantage Actor-Critic method;

AdaGrad is an adaptive gradient algorithm;

ANN is an artificial neural net;

BTTT is a backpropagation throw the time method;

DNN is a deep neural network;

GBO is a gradient-based optimization method;

GRU is a gated recurrent unit;

ML is machine learning;

LSTM is a long short-term memory network;

RC is random complexity;

RL is a reinforcement learning;

RNN is recurrent neural network.

NOMENCLATURE

Inf_{Sample} is a general information of input data (data set);

K_{input} is a number of element types in the neural network;

K_{corrY} is a number of independent variables that strongly correlate with the original features;

K_{imp} is a number of the most significant independent variables among factors4

$K_{ntcorrX}$ is a number of independent variables that are weakly dependent on others or do not correlate with each other;

n is a number of input features that characterize sample instances;

N_i is a multiple neurons at the network input;

N_{i_l} is a neuron at the network input;

N_o is a multiple neurons at the network output;

N_{o_p} is a neuron at the network output;

N_h is a multiple neurons of the hidden network layer;

N_{h_r} is a hidden network layer neuron;

$Num_{elemtype}$ is a number of element types in the neural network;

NN is a neural network;

NN_{struct} is a structure of neural network;

l is a number of neurons at the network input;

$Lev_{accmeas}$ is a measurement accuracy level;

Lev_{fctr} is a level of significant and less significant and/or non-significant factors4

Lev_{manag} is a level of possible control and management;

Lev_{task} is a conditional difficulty level of the task;

DOI 10.15588/1607-3274-2025-2-11

$Lev_{smplfctn}$ is a level of possible simplification of the structure;

m is a number of dependent (categorical) features of sample instances;

p is a number of neurons at the network output;

$Param_T$ is additional and specificity parameters of task;

q is a number of connections between neurons in the network;

r is number of neurons in the hidden network layer;

RC is random complexity;

$Sample$ is a data set;

$Task$ is general represent of the modeling task;

w is a multiple of connections between neurons;

w_q is a connection between neurons in the network;

x_n is a independent attribute of the sample instance;

X is a set of independent attribute (variables);

y_m is a value of the dependent variable (attribute) of the sample instance;

Y is a set of values of dependent variables.

INTRODUCTION

ML is widely used for diagnostics of complex technical processes, as it provides a number of advantages that are not available in classical approaches using manual control, rule systems, and the like. Today, ML is a powerful tool for solving such problems [1]–[4].

In nonlinear technical systems, there are quite complex patterns in the data received from Sensor Systems. Such data contains nonlinear relationships, i.e. technical processes often involve nonlinear interactions between several variables (for example, temperature, pressure, vibration). ML models can automatically recognize these patterns. In addition, most systems generate large amounts of sensor or operational data. ML can efficiently process large data sets and identify mission-critical functions [1]–[4].

The main further goal of using ML models is to automate diagnostics. ML-based models will allow you to perform real-time analysis. ML models can analyze data in real time, allowing instant fault detection and diagnostics. Moreover, the use of ML will reduce human intervention. Unlike manual diagnostics, ML systems can process data independently, reducing reliance on industry experts [1]–[4].

Processing noise in data and uncertainty. Complex systems often operate in environments where data is noisy or incomplete. ML models are designed to summarize imperfect data. In addition, ML models can estimate the probability of various failures or technological anomalies, providing probabilistic results that help with decision-making [1]–[4].

ML-based models are characterized by a high level of adaptability to new conditions. Technical processes often develop due to changes in operating conditions or system configuration. ML models can be retrained or modified to adapt to these changes. Pre-trained ML models can be adapted to new but similar processes with minimal additional training.

In industries such as manufacturing, power plants, or transportation, ML can perform diagnostic tasks for thousands of sensors and subsystems simultaneously. ML can be deployed on peripherals or in centralized systems to scale diagnostics in multiple locations [4].

ML models analyze patterns in historical data to predict when components might fail, preventing unplanned downtime. Uncontrolled ML models can detect deviations from normal operating conditions, warning of potential problems at an early stage.

ML models allow you to determine which variables or functions are most important for fault diagnosis, providing valuable information for system optimization. By studying patterns in misclassification or anomalies, ML can help pinpoint the root causes of problems [1].

Summing up, we should note the main advantages in terms of costs and efficiency, namely:

– reduced downtime: early detection of malfunctions minimizes production shutdowns and repair costs;

– reduced labor costs: automated diagnostics reduces the need for constant monitoring by operators;

– by providing accurate and timely information, ML allows you to make more informed decisions.

The object of study is synthesizing a high-precision neural network model based on previously accumulated historical data about the system.

The subject of the study is a neural network model of complex technical processes in nonlinear systems, which should be characterized by a high level of accuracy.

The purpose of the work is to construct and study neuromodels of complex technical processes in nonlinear systems, which should be characterized by a high level of accuracy with a preliminary definition of structural features based on the use of a system of indicators.

1 PROBLEM STATEMENT

Let it be that a data set $Sample$, containing data on mechanical parameters (e.g., vibration) recorded by specialized sensors and obtained during an operational study of a complex nonlinear technical system (e.g., helicopter transmission, car, or engine) is given. Then, $Sample = \langle X, Y \rangle$, where $X = \{x_1, x_2, x_3, ..., x_i\}$, where $i = \overline{1;1158}$, and $Y = \{y\}$ is the key output variable.

Then, it is necessary to determine such a set of $X*$, to ensure Y that the diagnostic error is minimized by a diagnostic model based on such a data set: $Error\left(Model_{diag}\left(X*\right)\right) \to \min$.

Most of the tasks, for which is planning to use ML models, have a different nature and a high level of specificity ($Param_T$) [5]. However, when using the apparatus of neural networks, it is sufficient to have a comprehensive assessment of the complexity of the task: $Task = \{Param_T, Lev_{Task}\}$ [5]. Such a comprehensive assessment can be obtained on the basis of information about the input data of the task (a sample of data) and a group of criteria for evaluating the accuracy of the data

DOI 10.15588/1607-3274-2025-2-11

and the requirements for the model: $Lev_{Task} = \{Inf_{sample}, Lev_{smplfctm}, Lev_{fctr}, Lev_{accmeas}, Lev_{manag}\}$.

It was noted in [5] that a complex neuromodel based on a RNN and DNN topologies will be sufficient for tasks belonging to the RC category. Then such a model (NN) will consist of: a set of neurons $N = \{N_i, N_o, N_h\}$ consisting of subsets of input $N_i = \{N_{i_1}, N_{i_2}, ..., N_{i_l}\}, l = 1,2,...,|N_i|$, output $N_o = \{N_{o_1}, N_{o_2}, ..., N_{o_p}\}, p = 1,2,...,|N_o|$, and hidden neurons $N_h = \{N_{h_1}, N_{h_2}, ..., N_{h_r}\}, r = 1,2,...,|N_h|$. The number of neurons in the hidden layer ($N_h = \{N_{h_1}, N_{h_2}, ..., N_{h_r}\}$, $r = 1,2,...,|N_h|$) can be calculated based on analytical estimates of the input data [5].

After that, it can proceed to determining the weights of connections between neurons $w = \{w_q\}$, in other words, to parametric synthesis. Having determined the values of the elements of sets, we can consider the synthesis of ANN: complete [5].

Therefore, the first subtask will be to determine the exact category of complexity of the problem based on the values of the criteria $Lev_{Task} = \{Inf_{sample}, Lev_{smplfctm}, Lev_{fctr}, Lev_{accmeas}, Lev_{manag}\}$ and data about the data sample. The next subtask will be the calculation of the number of neurons in the hidden layer of the network $|N_h| = K_{input} - K_{corrY} - K_{imp} - K_{ntcorrX}$ [5].

2 REVIEW OF THE LITERATURE

GRU is a type of RNN designed for processing sequential data. GRUs are particularly effective for tasks related to time series or sequential data, where dependencies need to be fixed over time. Presented in [6], GRUs are a simplified version of LSTM, but retain comparable performance while reducing computational complexity.

The GRU consists of two main gateways [6], [7]:

a) update gate:

– decides how much past information should be saved;

– helps the network focus on up-to-date past information, while forgetting unnecessary details;

b) reset gate:

– manages how much past information should be deleted for the current time step;

– allows the network to reload its memory when new patterns or states appear [6], [7].

GRUs do an excellent job of fixing time dependencies in sequential data, such as sensor readings during technical processes. They can detect patterns or anomalies in time-dependent data, which is crucial for diagnosing failures in dynamic systems.

GRUs have fewer parameters compared to LSTM, which reduces computing costs. This speeds up GRU training and deployment, especially for large data sets or systems with real-time constraints [6], [7].

GRUs can study long-term dependencies in sequences, avoiding problems such as vanishing gradients faced by Standard RNNs. This is very important for diagnosing processes in which the consequences of past events (for example, earlier anomalies) affect the current state [8].

Complex technical processes often lead to noisy data. GRUs are resistant to such noise due to their closed mechanisms that selectively filter out irrelevant information [9].

GRU can efficiently process multidimensional time series of data. For example, power plant operation Diagnostics may include simultaneous analysis of temperature, pressure, and vibration data [7].

GRUs are computationally efficient and can be deployed for real-time fault detection and diagnostics, which is vital in mission-critical systems such as production lines or power grids [7].

GRUs adapt well to a variety of technical processes, making them versatile for applications in various industries, from Automotive to aerospace.

GRU in diagnostics is recommended to be used for:

a) fault detection:

– detection of anomalies in machine behavior by analyzing time series of sensor data

– example: detection of unusual vibrations in turbines or engines;

b) preventive maintenance:

– predict possible system failures by analyzing historical data;

– example: industrial equipment wear monitoring for maintenance planning;

c) process optimization:

– analysis of time dependencies to optimize operating parameters;

– example: configure the input data of a chemical enterprise based on sensor feedback to maximize performance;

d) root cause analysis:

– tracking patterns in time series data to identify the underlying cause of the malfunction;

– example: diagnostics of pressure fluctuations in pipelines.

Thus, GRUs are ideal for diagnosing complex technical processes, as they provide an optimal balance of computational efficiency, resistance to encrypted data, and the ability to capture complex time patterns. Their real-time capabilities and adaptability make them a one-stop solution for dynamic applications with large amounts of data [8].

A3C is an advanced RL method that can become an effective method for synthesizing GRU-based networks, especially for tasks that require high accuracy, such as diagnosing complex technical processes. The benefits of A3C are in good agreement with the requirements for GRU training networks to perform these tasks [9], [10].

A3C is a reinforcement learning system in which:

– many agents work asynchronously in parallel environments, collecting data and learning from different experiences;

– it uses two networks:

DOI 10.15588/1607-3274-2025-2-11

– actor: determines what actions should be taken (network of policies);

– critic: evaluates the quality of actions taken by evaluating the value function;

– the method optimizes performance by combining policy-based methods (learning policies) and value-based methods (evaluating the value of states or actions);

– for GRU networks: GRU can serve as a basic architecture for participating and / or critical networks to process sequential or temporary data.

A3C uses the term entropy-based regularization to encourage research, allowing GRU to learn from less common but important patterns [9], [10].

GRUs, combined with A3C, perfectly captures these dependencies, focusing on sequences that maximize benefits while effectively identifying the most significant patterns in the data.

A3C stabilizes the learning process by asynchronous updating of weighting factors by multiple agents. This asynchronous process smooths out gradients, resulting in faster convergence and a reduced risk of overtraining.

3 MATERIALS AND METHODS

As it was given in the previous section, the modeling task can be unified for a specific task after a certain comprehensive assessment of its complexity. Given that the structure of ANN ($NN = (struct, param)$) allows to most subtly encode the relationships between the input data ($X = \{x_1, x_2, ...x_n\}$) [5], it is necessary to accurately select the synthesis option for such a non-network model. Based on the values of the indicators to assess the complexity of the task ($Lev_{Task} = \{Inf_{sample}, Lev_{smplfctm}, Lev_{fctr}, Lev_{accmeas}, Lev_{manag}\}$), it can be chosen a way to synthesize the most acceptable structure [5].

The general scheme of chosen the category of complicity using indicators prepared as a formula (1).

$$Lev_{Task} = \begin{cases} Lev_{smplfctn} \le 0, Lev_{fctr} \le 0, Lev_{accmeas} = 0, Lev_{manag} = 0 \to OS \\ Lev_{smplfctn} > 0, Lev_{fctr} \le 0, Lev_{accmeas} = 0, Lev_{manag} = 0 \to OC \\ Lev_{smplfctn} \le 0 \,||\, Lev_{smplfctn} \ge 0, Lev_{fctr} > 0, Lev_{accmeas} > 0, Lev_{manag} = 0 \to CAwP \\ Lev_{smplfctn} > 0, Lev_{fctr} > 0, Lev_{accmeas} \ge 0, Lev_{manag} \ge 0 \to RC \end{cases}. \quad (1)$$

Therefore, we can conclude that it has a level of complexity of the RC category, that is why we will be using combine of GRU model and A3C method for training (Fig. 1).

A3C directly optimizes the expected reward (or a surrogate) by combining [11], [12]:

Actor Loss: Encourages the GRU to make better predictions for diagnosis.

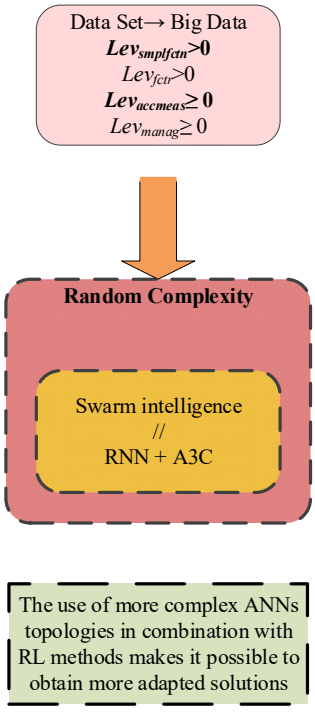

Figure 1 – Organized simplicity category of modeling task

Critic Loss: Ensures the network evaluates the quality of predictions accurately [13], [14].

This joint optimization ensures that the GRU learns both what actions to take and how good those actions are, leading to a more accurate and robust model.

In diagnosing complex processes, data is often noisy or changes over time.

A3C's parallel agents, combined with GRU's gating mechanisms, allow the system to:

– focus on relevant patterns;

– adapt to dynamic data distributions;

– improve robustness to outliers and transient conditions.

A3C utilizes multiple agents working independently, which reduces bottlenecks in learning long-term dependencies in GRUs.

This enables better model performance on large-scale problems with time-series data, where capturing subtle temporal patterns is essential [8]–[13].

GRUs already excel at modeling sequences, but A3C reinforces this by focusing on rewarded patterns:

– fault patterns that result in high rewards (accurate predictions) are emphasized;

– unimportant or noisy patterns are downplayed.

A3C agents experience diverse states and collect varied trajectories, effectively regularizing the training process. This diversity improves the GRU network's generalization, avoiding overfitting to specific fault scenarios.

Collect multivariate time-series data (in next section it will be noticed that this is vibration) from sensors.

A3C Setup:

– use GRUs for both the Actor and Critic networks:

– input: sequential sensor data;

DOI 10.15588/1607-3274-2025-2-11

– actor output: predicted fault class or action;

– critic output: value of the current sequence.

Training:

– deploy multiple agents in parallel, each exploring different sensor sequences and learning asynchronously.

Train GRUs to:

– recognize normal and fault patterns$

– predict future states or faults based on sequential dependencies.

Optimization:

– use the A3C loss function to refine GRU parameters:

– minimize actor loss for better fault classification.

– minimize critic loss to improve evaluation of prediction quality.

Deployment:

– the resulting GRU model can diagnose faults with high accuracy in real-time.

4 EXPERIMENTS

For testing was perpetrate next data set:

– 1158 Vibration Data Sets

– roughly 2/3s of the files are nominal data, while 1/3 have a gear fault.

Scoring: Percent Correct, total correct / 1158

Generally, for experimental comparing different strategies of ML approaches was using list of Python libraries and packages, as:

– TensorFlow Agents for RL methods;

– Keras for different ANNs models [15] (as DNN, RNN, CNN, etc.).

Table 1 – Model results based on test data

Model + method combination	Synthesis time, s	Accuracy on training part of data set	Accuracy on test part of data set
Perceptron + Backpropagation	5236	92.77	92.04
DNN + GBO	6286	96.52	95.72
DNN + AdaGrad	4632	97.24	96.25
GRU + Backpropagation Through Time	5965	96.39	95.76
GRU + A3C	7045	98.58	98.37
GRU + MGA	8906	98.73	98.66

5 RESULTS

Table 1 shows the results of models based on test data. During compression special attention was concentrated on accuracy of models for different depended features.

Accuracy was calculated as:

$$E = \frac{error_{class}}{Number_{sampl}} \cdot 100\% .$$

In multiclass classification, accuracy is a standard metric that measures the proportion of correctly predicted labels out of the total predictions made. It is particularly straightforward when each instance belongs to one of multiple classes, and only one label is correct.

For different ML models was using different training methods.

6 DISCUSSION

From the test results, it can be seen that the GRU-based approach in combination with RL is one of the slowest approaches. Therefore, the difference in comparison even with DNN sometimes almost 2 times indicates a high resource intensity in terms of time, because RL methods do not differ in high speed. Moreover, setting up GRU metaparameters requires additional processing of the gate value, which also slows down the synthesis time. On the other hand, when using the more classical BPTT, the time has been reduced, but this approach to training, firstly, is quite difficult to parallelize, which significantly slows it down for working on high-performance computing systems, and secondly, despite the more optimized synthesis time, the accuracy of work is still not high.

In addition, despite the large number of computing nodes, DNN also does not provide an acceptable level of accuracy. And given the importance of ensuring the safety of people during technical processes, it is not possible to compromise accuracy. Also, it is worth noting that despite the optimization of the synthesis time, when working with such a model, the load on computing resources is also high.

An ordinary Perceptron was taken for analysis simply to ensure that even the simplest neural network models show a high level of accuracy (more than 90%), but still do not fully allow abstracting complex data connections.

The use of MGA [16] for GRU synthesis was also considered separately as a stochastic approach. Despite the highest time indicators, this particular solution demonstrated the highest accuracy, but the synthesis process itself, in addition to high time requirements, also imposes high requirements on computing power.

CONCLUSIONS

The urgent scientific and applied problem of synthesizing a diagnostic model of complex technical processes in nonlinear systems, which should be characterized by a high level of accuracy, is considered is solved.

The scientific novelty lies in the study of the use of a system of criteria for determining the structural features of a neural network model. Based on the assessment of the complexity of the task and the system of indicators, it was possible to obtain neuromodel with a high level of accuracy of work.

The practical significance lies in the fact that the developed neural network models can be used during the implementation of real technical processes in production facilities. Their use will significantly reduce production costs and automate the modeling process.

Prospects for further research and development are mechanisms of optimization of computing resources using.

DOI 10.15588/1607-3274-2025-2-11

ACKNOWLEDGEMENTS

The work was carried out with the support of the state budget research project of the state budget of the National University "Zaporozhzhia Polytechnic": "Intelligent information processing methods and tools for decision-making in the military and civilian industries" (state registration number 0124U000250).

REFERENCES

1. Ameisen E. Building Machine Learning Powered Applications, Going from Idea to Product. California, O'Reilly Media, 2020, 260 p.
2. Bonaccorso G. Mastering Machine Learning Algorithms, Expert techniques to implement popular machine learning algorithms and fine-tune your models. Birmingham, Packt Publishing, 2018, 576 p.
3. Patan K. Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Process. Berlin, Springer, 2008, 112 p. DOI: 10.1007/978-3-540-79872-9
4. Leoshchenko S., Oliinyk A., Subbotin S., Zaiko T. Using Modern Architectures of Recurrent Neural Networks for Technical Diagnosis of Complex Systems, *2018 International Scientific-Practical Conference Problems of Info-communications. Science and Technology (PIC S&T), Kharkiv, 9–12 October 2018, proceedings*. Kharkiv, IEEE, 2018, pp. 411–416. DOI: 10.1109/INFOCOMMST.2018.8632015
5. Leoshchenko S., Subbotin S., Oliinyk A., Narivs'kiy O. Implementation of the indicator system in modeling complex technical systems. *Radio Electronics, Computer Science, Control*, 2021, Vol. 1, pp. 117–126. DOI: 10.15588/1607-3274-2021-1-12
6. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, URL: https://arxiv.org/abs/1406.1078
7. Ahmadian A., Salahshour S. Soft Computing Approach for Mathematical Modeling of Engineering. London, Chapman and Hall (CRC Press), 2021, 222 p.
8. Sayyaadi H. Modeling, Assessment, and Optimization of Energy Systems. Cambridge, Academic Press, 2020, 558 p.
9. Koulamas C., Lazarescu M.T. Real-Time Sensor Networks and Systems for the Industrial IoT. Basel, Mdpi AG, 2020, 242 p. DOI: 10.3390/books978-3-03943-431-2
10. Senge P. M. The Fifth Discipline, The Art & Practice of The Learning Organization. New York, Doubleday, 2006, 445 p.
11. Bruce P., Bruce A. Practical Statistics for Data Scientists, 50 Essential Concepts. California, O'Reilly Media, 2017, 318 p.
12. Finch W.H. Exploratory Factor Analysis. California, SAGE Publications, 2019, 144 p.
13. Rencher A. C., Christensen W. F. Methods of Multivariate Analysis. New Jersey, John Wiley & Sons, 2012, 800 p.
14. Dean A., Voss D., Draguljić D. Design and Analysis of Experiments (Springer Texts in Statistics), 2nd Edition. Berlin, Springer, 2017, 865 p. DOI: 10.1007/978-3-319-52250-0
15. Sewak M. Deep Reinforcement Learning, Frontiers of Artificial Intelligence. Berlin, Springer, 2020, 220 p. DOI: 10.1007/978-981-13-8285-7
16. Leoshchenko S., Oliinyk A., Subbotin S., Lytvyn V., V. Shkarupylo V. Modification and parallelization of genetic algorithm for synthesis of artificial neural networks. *Radio Electronics, Computer Science, Control*, 2019, Vol. 4, pp. 68–82. DOI: 10.15588/1607-3274-2019-4-7.

Received 26.01.2025.
Accepted 04.04.2025.

УДК 004.896

СИНТЕЗ НЕЙРОМЕРЕЖЕВИХ МОДЕЛЕЙ ДЛЯ ТЕХНІЧНОГО ДІАГНОСТУВАННЯ НЕЛІНІЙНИХ СИСТЕМ

Леощенко С. Д. – д-р філософ., доцент кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя Україна.

Олійник А. О. – д-р техн. наук, доцент, професор кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

Субботін С. О. – д-р техн. наук, професор, завідувач кафедри програмних засобів Національного університету «Запорізька політехніка», Запоріжжя, Україна.

Морклянник Б. В. – д-р техн. наук, професор, професор кафедри інформаційних технологій, Воєнна академія, м. Київ, Україна

АНОТАЦІЯ

Актуальність. Розглянуто задачу синтезу діагностичної моделі складних технічних процесів у нелінійних системах, що має відрізнятися високим рівнем точності. Об'єктом дослідження є процес синтезу нейромережевої моделі для технічного діагностування нелінійних систем.

Мета роботи полягає у синтезі нейромережевої моделі високої точності, на основі попередньо накопичених історичних даних про систему.

Метод. Запропоновано використовувати штучні нейронні мережі для моделювання нелінійних технічних систем. По-перше, необхідно виконати загальну оцінку складності задачі. На основі оцінки можна прийняти рішення про подальший підхід до організації синтезу нейромоделі. Від так, для поставленої задачі було обрано рівень складності безладна складність, адже не зважаючи на відносну структурованість даних, їх загальний масив є досить великим за об'ємом та вимагає ретельного опрацювання з метою забезпечення високої якості рішення. Тому в подальшому було запропоновано використовувати нейромодель на основі рекурентних мереж топології GRU та використати для нейросинтезу методи ройового інтелекту, зокрема метод АЗС. Отримані результати засвідчили високий рівень отриманого рішення, проте через високий рівень ресурсоємності запропонований підхід вимагає подальших модифікацій.

Результати. Отримано діагностичну модель складних технічних процесів у нелінійних системах оптимальної топології, що відрізняється високим рівнем точності. Побудована нейромодель знижує ризики пов'язані зі забезпеченням людської безпеки.

Висновки. Проведені експерименти підтвердили працездатність запропонованого підходу і дозволяють рекомендувати його для подальшого допрацювання з метою імплементації на практиці в системи автоматизації систем контролю технічних, промислових та експлуатаційних процесів. Перспективи подальших досліджень можуть полягати в оптимізації ресурсоємності процесів синтезу.

КЛЮЧОВІ СЛОВА: технічне діагностування, нелінійні системи, машинне навчання, синтез нейронних мереж, система індикаторів, нейромодель, вибірка, навчання, помилка.

DOI 10.15588/1607-3274-2025-2-11

ЛІТЕРАТУРА

1. Ameisen E. Building Machine Learning Powered Applications: Going from Idea to Product / E. Ameisen. – California: O'Reilly Media, 2020. – 260 p.
2. Bonaccorso G. Mastering Machine Learning Algorithms: Expert techniques to implement popular machine learning algorithms and fine-tune your models / G. Bonaccorso. – Birmingham : Packt Publishing, 2018. – 576 p.
3. Patan K. Artificial Neural Networks for the Modelling and Fault Diagnosis of Technical Processe / K. Patan. – Berlin : Springer, 2008. – 112 p. DOI: 10.1007/978-3-540-79872-9
4. Using Modern Architectures of Recurrent Neural Networks for Technical Diagnosis of Complex Systems / [S. Leoshchenko, A. Oliinyk, S. Subbotin, T. Zaiko] // 2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T), Kharkiv, 9–12 October 2018 : proceedings. – Kharkiv: IEEE, 2018. – P. 411–416. DOI: 10.1109/INFOCOMMST.2018.8632015
5. Implementation of the indicator system in modeling complex technical systems / [S. Leoshchenko, S. Subbotin, A. Oliinyk, O. Narivs'kiy] // Radio Electronics, Computer Science, Control. – 2021. – Vol. 1. – P. 117–126. DOI: 10.15588/1607-3274-2021-1-12
6. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation [Electronic resource]. – Access mode: https://arxiv.org/abs/1406.1078
7. Ahmadian A. Soft Computing Approach for Mathematical Modeling of Engineering / A. Ahmadian, S. Salahshour . – London: Chapman and Hall (CRC Press), 2021. – 222 p.
8. Sayyaadi H. Modeling, Assessment, and Optimization of Energy Systems / H. Sayyaadi. – Cambridge : Academic Press, 2020. – 558 p.
9. Koulamas C. Real-Time Sensor Networks and Systems for the Industrial IoT / C. Koulamas, M. T Lazarescu. – Basel: Mdpi AG, 2020. – 242 p. DOI: 10.3390/books978-3-03943-431-2
10. Senge P. M. The Fifth Discipline: The Art & Practice of The Learning Organization / P. M. Senge. – New York : Doubleday, 2006. – 445 p.
11. Bruce P. Practical Statistics for Data Scientists: 50 Essential Concepts / P. Bruce, A. Bruce. – California : O'Reilly Media, 2017. – 318 p.
12. Finch W. H. Exploratory Factor Analysis: 1st Edition / W. Holmes Finch. – California : SAGE Publications, 2019. – 144 p.
13. Rencher A. C. Methods of Multivariate Analysis / A. C. Rencher, W. F. Christensen. – New Jersey : John Wiley & Sons, 2012. – 800 p.
14. Dean A. Design and Analysis of Experiments (Springer Texts in Statistics) / A. Dean, D. Voss, D. Draguljić. – Berlin : Springer, 2017. – 865 p. DOI: 10.1007/978-3-319-52250-0
15. Sewak M. Deep Reinforcement Learning: Frontiers of Artificial Intelligence / M. Sewak. – Berlin : Springer, 2020. – 220 p. DOI: 10.1007/978-981-13-8285-7
16. Modification and parallelization of genetic algorithm for synthesis of artificial neural networks / [S. Leoshchenko, A. Oliinyk, S. Subbotin et al.] // Radio Electronics, Computer Science, Control. – 2019. – № 4. – P. 68–82. DOI: 10.15588/1607-3274-2019-4-7

DOI 10.15588/1607-3274-2025-2-11