ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

UDC 681.324 doi: 10.32620/reks.2026.2.13

Stanislav DOVGIY[1], Oleh KOPIIKA[1], Oleg BARABASH[2], Yurii KRAVCHENKO[3], Andriy MAKARCHUK[2]

[1] ***Institute of applied control systems of the National Academy of Sciences of Ukraine, Kyiv, Ukraine***
[2] ***National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine***
[3] ***Taras Shevchenko National University of Kyiv, Kyiv, Ukraine***

A MACHINE LEARNING-BASED METHOD FOR ASSESSING THE FUNCTIONAL STABILITY OF DISTRIBUTED INFORMATION SYSTEMS THROUGH TOPOLOGICAL STRUCTURE

The subject of this *research is the application of machine learning to detect the functional stability of distributed information systems based on the numerical characteristics of their topology.* ***This study aims*** *to improve the accuracy of verifying the condition whose fulfillment ensures the functional stability of considered distributed information systems.* ***The task*** *of this research is to improve the accuracy of machine learning-based verification of the condition whose fulfillment ensures the functional stability of a considered distributed information system.* ***The main result*** *of the study is the development of an ensemble classifier based on multiple heterogeneous machine learning models, enabling superior accuracy compared to that reported in related studies on the same or similar tasks. Unlike other studies on applying machine learning to assess information system functional stability, the proposed classifier architecture allows sequential use of sets of machine learning models. In other words, the first set of models receives a group of input parameters and estimates certain additional parameters. The obtained estimates are then passed to another set of models, which produces the final result. This approach enables the discovery of deeper dependencies between input and output parameters, which in turn yields higher accuracy. Such an approach differs significantly from the one commonly used in practice, where all models receive the same input parameters, produce the same type of output, and aggregate their outputs depending on the specific task.* ***Conclusions.*** *The study's outcome is a trained and tested ensemble machine learning model that, based on several numerical characteristics, detects whether the condition is met for the considered distributed information system to be regarded as functionally stable. The ensemble machine learning model obtained in this work can be applied at the design stage of information systems. When used as demonstrated here, the machine learning models trained in the study can be integrated into software for the design and simulation of information systems. The method for combining these machine learning models is of particular interest. As described in the paper, the approach to organizing models within the ensemble opens new possibilities for building highly accurate classifiers. The principal* ***scientific novelty*** *of the obtained results lies in a new approach to constructing ensemble machine learning models. Unlike classical ensembles, such as random forests, the ensemble described in this study employs a set of machine learning models to compute estimates of certain parameters, which are then used as inputs to another set of models. This particular approach to organizing the ensemble most likely improved the quality of solving the task, as demonstrated within the study.*

Keywords: *machine learning; ensembles; functional stability; information systems; statistical analysis.*

1. Introduction

1.1. Motivation

The active computerization of several important fields related to human activity requires the increasingly intensive use of different distributed information systems. However, as the use of distributed information systems expands, so do the requirements imposed on them. As demonstrated in practice, one of the most important requirements currently placed on distributed information systems is their functional stability. The functional stability of a system is its complex property, which describes the possibility of this system to solve given tasks, at least partially considering possible inner or outer destructive effects. In fact, this property may be critical in many cases because it includes the reliability, survivability, and redundancy of the considered system, which allows for a more complex analysis of the system.

At present, a number of metrics, commonly referred to in the literature as "functional stability indicators," indicate functional stability, as well as a set of conditions under which an information system may be regarded as functionally stable. One of the simplest such conditions can be formulated as follows:

Creative Commons Attribution NonCommercial 4.0 International

$$\lambda(G) \geq 2 \wedge \kappa(G) \geq 2, \quad (1)$$

where $\lambda(G)$ is a degree of edge connectivity of a graph G, which describes the topology of the considered distributed information system, and $\kappa(G)$ is a vertex connectivity of this graph. Condition (1) is arguably the most intuitively clear among those developed to date, as well as one of the most thoroughly studied. However, it also has its drawbacks, the most significant of which is the relatively large number of computations required for its verification. This has led to the use of machine learning [1]. Nevertheless, preliminary experiments with its application [1] demonstrate that typical classification models do not provide sufficient accuracy in determining the realization of condition (1) based on a set of numerical characteristics of the topology of the distributed information system under consideration. Therefore, an alternative machine learning-based approach that would overcome this drawback is needed.

Although many methods have been proposed for assessing the functional stability of information systems, their practical applicability remains limited. Classical analytical approaches require significant computational effort; therefore, they cannot be easily integrated into modern software tools used during system design. At the same time, machine-learning-based solutions typically address only isolated aspects of system behavior and do not provide a comprehensive view of system stability. The proposed method overcomes these limitations by relying solely on the numerical topological characteristics extracted automatically from the system model. This creates a direct connection between our results and real information systems, enabling seamless integration of the classifier into existing design and simulation environments to support automated stability assessment throughout the system lifecycle.

1.2. State of the art

System modeling, including distributed information systems, constitutes a logical component of their investigation before direct operation. This step enables the identification of the specific features of the system under consideration, including its functional stability.

Many studies devoted to functional stability estimation rely on well-established mathematical apparatus. For instance, in several works, the authors propose applying methods inherent to the theory of dynamical systems and related domains. For example, in [2, 3], authors describe a usage of methods, typical for dynamical systems theory, for analysis of functional stability, and in [4, 5], authors purpose to use more complex methods, based on general concepts of mathematical modeling. Some other works describe approaches that are typical for combinatorial analysis and discrete mathematics. For example, [6, 7] purpose ways of analysis of functional stability, similar to methods of graph theory, and [8] describes the use of methods of combinatorial analysis and set theory. However, such approaches have certain significant drawbacks, such as high computational complexity. Therefore, the transition to methods associated with artificial intelligence appears logical.

Several studies have been dedicated to tasks related to the functional resilience of information systems and their resolution with ML. For example, in [9, 10], the authors regard artificial intelligence as a tool for traffic analysis, which can help gain an overall understanding of the system's operational resilience. In a number of works, is applied to the detection of cybersecurity-related aspects that may influence functional stability. For example, in [11], the authors describe the methodological aspects of the usage of AI in this field, [12] describes the use of AI for analyzing social risks, and [13] describes how AI may be used for the analysis of the behavior of individual users. Some works address the use of machine learning in assessing a range of technical aspects associated with the resilience of information systems. For example, in [14], authors show how neural networks may be used in energetics, [15] describes the usage of artificial intelligence for modeling the architecture of information systems, and [16] describes possible problems related to the usage of artificial intelligence in digital hardware. However, all such studies, including those cited above, employ machine learning to examine only individual aspects of functional stability, rather than resilience as an integral property of the system.

The authors of [1] attempted to assess functional stability as a distinct property of an information system through machine learning. In that study, the authors conducted experiments with several machine learning models that demonstrated reasonable accuracy in detecting whether the information system under consideration is functionally stable. Nonetheless, the accuracy reported in [1] remains insufficient for application, particularly in critical infrastructure or during an information system's autonomous operation.

Thus, based on the analyzed state of the art, it can be concluded that while contemporary international studies extensively employ machine learning and artificial intelligence to evaluate network anomalies, data reliability, and infrastructural vulnerability, they typically treat stability parameters in isolation. Most existing frameworks focus on specialized engineering or cybersecurity aspects rather than addressing functional stability as an integral property of the information system determined by its underlying topological structure. Consequently, a critical research gap remains in developing comprehensive, low-complexity machine learning methods capable of fully identifying global structural stability conditions with high accuracy, underscoring the necessity of the alternative approach proposed in this study.

1.3. Objectives and tasks

To solve the described problem within the scope of the study, the following tasks were set:

– To select the structure of an ensemble learning model and the numerical parameters to be used as its input;

– To generate a dataset for training the chosen machine learning model and to analyze this dataset for possible relationships among the selected input parameters;

– To train the chosen machine learning model and compare its accuracy with existing machine-learning-based solutions to the stated problem.

Section 2 of the study is dedicated to describing the general approach to solving the problem. Section 3 describes the main result of the fork. In particular, subsection 3.1 provides a general description of the architecture of the machine learning model. Subsection 3.2 demonstrates an argumentation of the reasons for the usage parameters used in the study as input of the developed machine learning model. Subsection 3.3 provides details of the developed architecture (section 2) and compares the accuracy of the machine learning model's architecture with that of other machine learning models for analog problems.

1.4. Case-study

This study considers a structural evaluation scenario of a modern cloud-based distributed information system (DIS) to demonstrate the practical relevance and applicability of the proposed machine learning framework. Such a system structurally comprises numerous heterogeneous processing nodes (i.e., servers, database units, and edge-gateways) interconnected via high-speed communication lines. These components are mapped onto an unweighted undirected graph G in graph-theoretical modeling, where the vertex set V denotes the active computing nodes and the edge set E represents bidirectional information channels.

The DIS must maintain its functional stability under severe operational conditions, including hardware failures, localized cyberattacks, or accidental wire breaks, ensuring that critical data routing remains uncompromised. The verification condition (1) serves as an early-stage architectural filter during system deployment. If the topology yields $\kappa(G)=\lambda(G)$, the system guarantees maximal structural resilience under independent node or edge failures. This study systematically evaluates how the developed sequential ensemble handles diverse topological configurations (ranging from sparse tree-like networks to dense, highly redundant cluster topologies), demonstrating its ability to automate stability checks in real-time network simulators without executing computationally prohibitive exhaustive search routines.

2. Materials and research methods

This study aims to use machine learning to determine the functional stability of an information system based on its topology. Within the framework of the work, it is assumed that, to improve the results obtained for verifying the fulfillment of condition (1), it is possible to sequentially employ several models trained separately, i.e., some models use the outputs of others as inputs. A dataset was generated using the Monte Carlo method, in which preselected parameters were computed for randomly selected graphs. The analysis of the dataset, model training, and evaluation of results were performed in Python.

3. Results and Discussion

3.1. Description of an improved approach for detecting the fulfillment of condition (1)

As demonstrated in [1], achieving the required accuracy in identifying the fulfillment of condition (1) using a single machine learning model is rather challenging. Therefore, it is logical to employ not just one model but several. Accordingly, the following approach was adopted within the framework of this study. Let there be two ML models, one of which estimates $\frac{\lambda(G)}{n-1}$, where n is a number of machines in a considered information system, and second estimates $\frac{\kappa(G)}{n-1}$. In addition, let there be a third machine learning model, which uses estimates of $\frac{\lambda(G)}{n-1}$ and $\frac{\kappa(G)}{n-1}$, given by the first second model, and gives 1, if condition (1) is realized, and 0 in the opposite case. Visually, this schema may be presented as shown in Fig. 1.

Each model depicted in Fig. 1 as a rectangle is trained independently and may have its own architecture and hyperparameters. Unlike common practice, every model architecture and its hyperparameters are selected independently to guarantee maximum output accuracy.

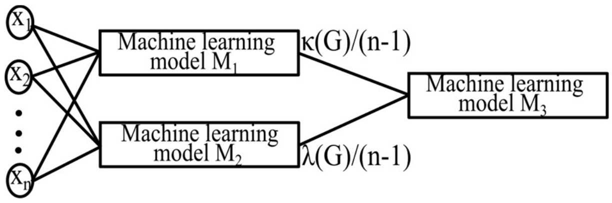

Fig. 1. Schema of modified way of estimation of realization of condition (1)

3.2 Definition of information system parameters that could be used as input for machine learning models

3.2.1. Selection of numerical parameters to be used as input for the first two machine learning models

Condition (1), like all formulated indicators and conditions of functional resilience, is closely related to the topology of the information system, as observed in several studies [1]. Therefore, it is logical to give to models M_1 and M_2 a number of numerical parameters of topology as input. Thus, by analyzing publications such as [1], it can be concluded that the following parameters should be supplied as inputs to the aforementioned models:

$$x_1 = \frac{d_{min}}{n-1}, x_2 = \frac{d_{max}}{n-1}, x_3 = S, \qquad (2)$$

where n is a number of machines (module, or something like this) in the considered information system, S is a saturation of graph G, what describes a topology of the considered information system, d_{min} and d_{max} are min and max degrees of a node of graph G, respectively. However, it is clear that these parameters may be insufficient. Therefore, there is a sense to additionally select some numeric parameters of a graph G, which could potentially affect the realization of condition (1), even if this may appear counterintuitive or unjustified at first glance. For example, to (2), one may add parameters

$$x_4 = \frac{e_{min}}{n-1}, x_5 = \frac{e_{av}}{n-1},$$
$$x_6 = \frac{e_{max}}{n-1}, x_7 = \frac{2^{\frac{n}{2}} \det A}{n^{\frac{n}{2}}}, \qquad (3)$$

where A is a adjacency matrix of a graph G, e_{min} and e_{max} min and max eigenvalues of this matrix, e_{av} is a mean of all eigenvalues of adjacency matrix and n is a number of machines in the considered information system.

3.2.2. Regression analysis to determine the influence of parameters (2)-(3) on the final result

To train the machine learning models, a dataset was generated by randomly selecting 3999996 connected graphs, which includes the parameters (2)–(3), parameters $\frac{\lambda(G)}{n-1}$ and $\frac{\kappa(G)}{n-1}$, and, in addition, a column, in which 1 means a realization of condition (1) for the selected graph and 0 in the opposite case.

Before directly training the aforementioned models, the generated dataset should be preliminarily analyzed to determine whether it is appropriate to use all parameters (2)–(3) as inputs or whether only a subset of them should be employed. For this purpose, we constructed a visualization of the correlation matrix (Fig. 2).

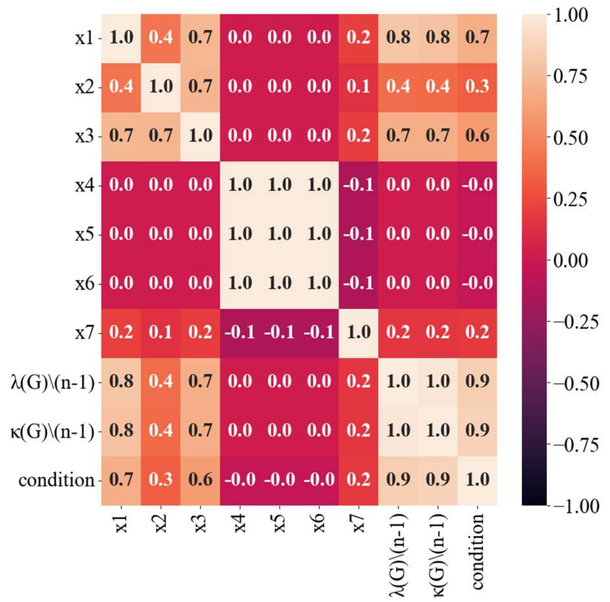

Fig. 2. Visualization of correlation matrix

Several important patterns can be clearly observed in Fig. 2. First, the parameters x_4, x_5 and x_6 are potentially strongly correlated with each other. Therefore, only one of these features should be provided as input to the machine learning model. In this way, the likelihood of redundancy in learning models, that predict, that predict values. $\frac{\lambda(G)}{n-1}$ and $\frac{\kappa(G)}{n-1}$, can be reduced. Secondly, in Fig. 1 it can be observed that these quantities also correlate strongly with each other, as well as with the validity of condition (1). This may raise certain concerns regard-

 ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

ing the appropriateness of the previously described approach. However, if the approach proves to be justified, this observation substantially simplifies the selection and calibration of the model that will perform predictions, based on $\frac{\lambda(G)}{n-1}$ and $\frac{\kappa(G)}{n-1}$, realization of condition (1). To verify the above considerations, we construct two sets of scatter plots: the first one for the parameters x_4, x_5 and x_6 (Fig. 3) and second one for the parameters $\frac{\lambda(G)}{n-1}$, $\frac{\kappa(G)}{n-1}$ and realization of condition (1) (Fig. 4). Considering the mathematical aspects of all described parameters, all scatter plots will be plotted in unit squares. This factor is practical for two reasons. First, it will help us to analyze the dataset used for model training and to extrapolate the results of this analysis on all possible cases. Second, this factor indicates that the selected parameters range from 0 to 1, which allows the experiment with the architectures of the model in the ensemble, which architecture is described above.

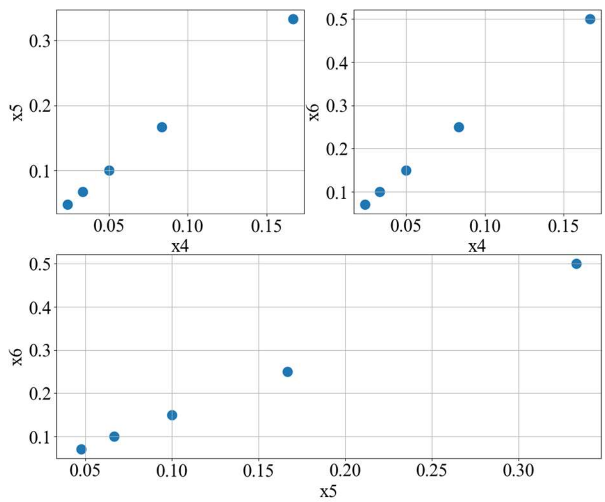

Fig. 3. Scatter plots for parameters x_4, x_5 and x_6

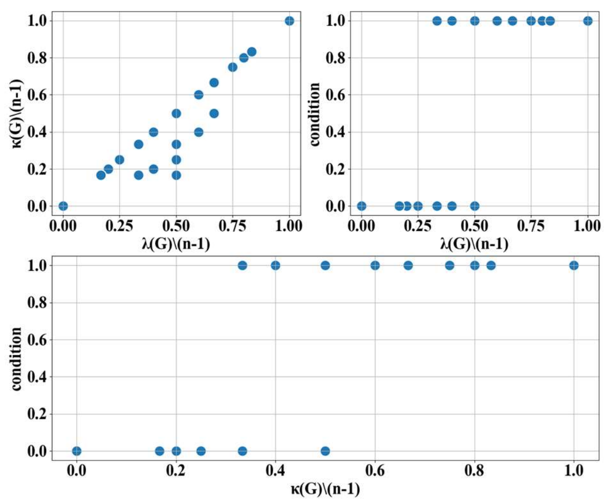

Fig. 4. Scatter plots for $\frac{\lambda(G)}{n-1}$, $\frac{\kappa(G)}{n-1}$ and realization of condition (1)

As shown in Fig. 3, there is a real linear dependency between parameters x_4, x_5 and x_6. This implies that to minimize the potential redundancy of the models M_1 and M_2 as input may be used only once (for example, x_4). On the other hand, on the first scatter plot in Fig. 4, we may see, that parameters $\frac{\kappa(G)}{n-1}$ and $\frac{\lambda(G)}{n-1}$, which will be predicted by models M_1 and M_2, respectively, do not have concrete dependencies. In addition, Fig. 4 shows that the selection of the architecture of model M_3 makes sense given its geometric interpretation.

3.3. Direct training of machine learning models and analysis of the obtained results

The study suggests, that (see Fig. 1): model M_1 is feed-forward neural network with three hidden layers, every of which have 750 neurons and $\sinh(x)$ as activation function, and output neuron with $\sigma(x)$ as activation function; model M_2 is similar to M_1, but first and third hidden layers have 725 neurons, and the second has 775 neurons; model M_3 is based on SVM with quadratic kernel. The result of the training of the described models and its testing on a separately generated dataset was given as follows: if we try to predict the realization of condition (1) using modes M_1, M_2 and M_3, what will be connected to each other as shown in Fig. 1, the prediction accuracy will be 96.8%.

To comprehensively validate the efficiency of the developed sequential framework and confirm the achievement of the research objectives, its classification metrics must be critically compared not only with the baseline models originally structured in [1] but also with alternative state-of-the-art machine learning configurations and advanced ensemble techniques commonly applied to topological feature processing, such as optimized logistic Regression Ensembles described in recent methodological literature [17]. All evaluated classification paradigms, including standalone decision trees, classical random forests, multilayered feed-forward neural networks (FNN), and multi-model architectural combinations, were trained, calibrated, and validated using the same large-scale dataset generated from 3,999,996 topological graph samples, providing a mathematically rigorous and fair benchmark. The comparative performance metrics are presented in Table 1.

Let us compare the accuracy of the way prediction of the realization of condition (1), described in this work, with a way of solving the same problem, but described in [1] (Table 1).

Table 1
Comparative analysis of classification accuracy for condition (1) verification across different machine learning frameworks

Model Architecture	Operational Framework / Source Origin	Accuracy, %	False positive rate, %	False positive rate, %
Decision tree	Baseline topology evaluator [1]	85.9	11	17
Random forest	Classical uniform voting paradigm [1]	86.7	11.9	14.8
Feed forward neural network	Standard direct feature mapping [1]	86.5	11.4	15.3
Logistic regression ensemble	Optimized invariant approximation [17]	88.2	9.8	12.1
Proposed sequential ensemble	Developed multi-layer Framework	96.8	3.6	2.7

As shown in Table 1, the method developed in this work for detecting the realization of condition (1) performs significantly better than the machine learning models discussed in [1].

4. Discussion

The experimental results summarized in Table 1 reveal that the developed sequential machine learning architecture provides substantial performance gains over both the standalone baseline models investigated in [1] and the uniform ensemble techniques described in the related literature [17]. The noticeable increase in accuracy (96.8%) primarily stems from the unique structural decoupling within the ensemble framework. Instead of directly mapping highly nonlinear topological graph descriptors to a binary stability condition, the first layer of the ensemble (models M_1 and M_2) transforms raw spectral and structural features into intermediate normalized invariants ($\frac{\kappa(G)}{n-1}$ and $\frac{\lambda(G)}{n-1}$). This sequence substantially simplifies the optimization landscape for the final SVM-based classifier (M_3), which performs decision boundary separation within a reduced, geometrically bounded unit square. From an engineering perspective, minimizing False Positive Rate (FPR) down to 3.6% and False Negative Rate (FNR) to 2.7% holds profound practical implications for DIS synthesis. A low FNR ensures that architectures flagged as structurally stable are not mistakenly identified as vulnerable, preventing catastrophic failures in production environments. However, certain computational trade-offs must be outlined. The deployment and inference times of the trained ensemble are nearly instantaneous, making it suitable for rapid automated CAD design tools. However, the offline training phase exhibits elevated complexity. Training three independent, distinct learning models requires a strictly calibrated, balanced dataset. Currently, this dataset generation heavily relies on exhaustive Monte Carlo graph simulations. Therefore, future research directives will focus on evaluating the framework's behavior under dynamic, scale-free graph topologies and investigating approximation techniques [17] to alleviate the database compilation constraints without decaying the achieved classification precision.

5. Conclusions

As a result of this study, a new classifier was built to predict the realization of condition (1) for an information system. As shown in the Fig. 1, this classifier consists of three machine learning models: two neural networks that evaluate the values of the variables $\frac{\kappa(G)}{n-1}$ and $\frac{\lambda(G)}{n-1}$, where G is a graph that describes a topology of the considered information system, $\lambda(G)$ is a degree of edge connectivity of graph G, $\kappa(G)$ is a degree of vertex connectivity of this graph and n is a number of machines in the considered information system, and a model based on SVM with quadratic kernel that makes a direct prediction of the realization of condition (1). As can be seen from (2)-(3), Figures 2-3, and Table 1, if the following parameters are passed to the neural networks as input

$$\frac{d_{min}}{n-1}, \frac{d_{max}}{n-1}, S, \frac{e_{min}}{n-1}, \frac{2^{\frac{n}{2}} \det A}{n^{\frac{n}{2}}},$$

ISSN 1814-4225 (print)
ISSN 2663-2012 (online)

the classifier described in the paper will provide a fairly high accuracy.

The primary advantage of the resulting classifier is its high accuracy. In contrast to the single-model approach proposed in [1] for this task, our method, which employs multiple models where the output of one serves as the input to another, achieves substantial accuracy improvements, as shown in Table 1. Nevertheless, this classification method has a major drawback: a significant increase in computational complexity, particularly during training. This issue highlights the need for additional research to reduce computational load during the training and operation of these machine learning models without compromising accuracy.

A potential solution to this problem may be the usage of approximation methods [17] as alternatives to models M_1 and M_2. A potential possibility of this approach may be argued by the existence of results in the usage of machine learning based methods in approximation [18]. However, this task and related tasks for implementing the obtained results [19, 20] may be a topic for future research, especially in the context of the research described in previous works [21, 22].

Contributions of authors: conceptualization, methodology – **Andriy Makarchuk;** formulation of tasks, analysis **– Oleh Kopiika, Oleg Barabash, Stanislav Dovgiy;** development of model, software, verification **– Andriy Makarchuk;** analysis of results, visualization **– Andriy Makarchuk, Yurii Kravchenko;** writing – original draft preparation, writing – review and editing **– Oleg Barabash, Andriy Makarchuk.**

Conflict of Interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship, or otherwise, that could affect the research and its results presented in this paper.

All the authors have read and agreed to the published version of this manuscript.

Financing

This study was conducted without financial support.

Data Availability

Data will be made available upon reasonable request.

Use of Artificial Intelligence

The authors confirm that they did not use artificial intelligence technologies when creating the current work.

References

1. Barabash, O., Makarchuk, A., Musienko, A., & Maistrov, O. Prediction of fulfilling the conditions of functional reliability of information systems using machine learning models. Proceedings of the 7th International Conference on Actual Problems of Unmanned Aerial Vehicles Development (APUAVD 2024), Kyiv, Ukraine, IEEE, 2024, pp. 179–182. DOI: 10.1109/APUAVD64488.2024.10765875.

2. Narendra, K. S., & Annaswamy, A. M. Stable Adaptive Systems. Dover Publications, 2012, 512 p. Available at: https://books.google.com.ua/books/about/Stable_Adaptive_Systems.html?id=CRJhmsAHCUcC&redir_esc=y (accessed 12 February 2025).

3. Shmatko, O., Yevseiev, S., Milov, O., Sporyshev, K., Opirskyy, I., Glukhov, S., Rudenko, Y., Nalyvaiko, A., Dakov, S., & Sampir, O. Development of a model of the information and analytical system for making decisions on detecting failures of information transmission channels. Eastern-European Journal of Enterprise Technologies, 2024, vol. 3, no. 9 (129), pp. 28–36. DOI: 10.15587/1729-4061.2024.306179.

4. Mikhail, M., Ouali, M.-S., & Yacout, S. A data-driven methodology with a nonparametric reliability method for optimal condition-based maintenance strategies. Reliability Engineering & System Safety, 2024, vol. 241, article no. 109668. DOI: 10.1016/j.ress.2023.109668.

5. Khalil, H. K. Nonlinear Systems. Prentice Hall, 2002, 750 p. Available at: https://www.egr.msu.edu/~khalil/NonlinearSystems/index.html (accessed 12 February 2025).

6. Aalirezaei, A., Kabir, G., & Khan, S. A. Dynamic predictive analysis of the consequences of gas pipeline failures using a Bayesian network. International Journal of Critical Infrastructure Protection, 2023, vol. 43, article no. 100638. DOI: 10.1016/j.ijcip.2023.100638.

7. Orda, A., & Rom, R. Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. Journal of the ACM, 1990, vol. 37, no. 3, pp. 607–625. DOI: 10.1145/79147.214078.

8. Fu, T., Zhang, Y., & Li, C. Two typical analytic models for reverse bond percolation on real networks. Physica A: Statistical Mechanics and its Applications, 2023, vol. 625, article no. 129029. DOI: 10.1016/j.physa.2023.129029.

9. Javaid, A., Niyaz, Q., Sun, W., & Alam, M. A Deep Learning Approach for Network Intrusion Detection. Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies, 2016, pp. 1–6. DOI: 10.4108/eai.3-12-2015.2262516.

10. Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. A Detailed Analysis of the KDD CUP 99 Data Set. Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009, pp. 1–6. DOI: 10.1109/CISDA.2009.5356528.

11. Shone, N., Tran Nguyen, N., Vu Dinh, P., & Shi, Q. A Deep Learning Approach to Network Intrusion Detection. IEEE Transactions on Emerging Topics in Computational Intelligence, 2018, vol. 2, no. 1, pp. 41–50. DOI: 10.1109/TETCI.2017.2772792.

12. Chen, H., Chen, P., Wang, B., Yu, X., Chen, X., Ma, D., & Zheng, Z. Graph neural network based robust anomaly detection at service level in SDN driven microservice system. Computer Networks, 2024, vol. 239, article no. 110135. DOI: 10.1016/j.comnet.2023.110135.

13. Eke, C. I., Norman, A. A., Shuib, L., & Nweke, H. F. A Survey of User Profiling: State-of-the-Art, Challenges, and Solutions. IEEE Access, 2019, vol. 7, pp. 144907–144924. DOI: 10.1109/ACCESS.2019.2944243.

14. Ma, Z., Jiang, G., & Chen, J. Physics-informed ensemble learning with residual modeling for enhanced building energy prediction. SSRN Electronic Journal, 2024, pp. 1–40. DOI: 10.2139/ssrn.4885240.

15. Dovgiy, S., Kopiika, O., & Kozlov, O. Architectures for the information systems, network resources, and network services. Proceedings of the Cybersecurity Providing in Information and Telecommunication Systems, 2021, pp. 293–301. Available at: https://ceur-ws.org/Vol-3187/short9.pdf (accessed 12 February 2025).

16. Boche, H., Fono, A., & Kutyniok, G. Limitations of deep learning for inverse problems on digital hardware. IEEE Transactions on Information Theory, 2023, vol. 69, no. 12, pp. 7887–7908. DOI: 10.1109/TIT.2023.3326879.

17. Papchenko, O., Kuzikov, B., & Shovkoplias, O. Enhancing functional efficiency in information-extreme machine learning with logistic regression ensembles. Radioelectronic and Computer Systems, 2023, no. 4 (108), pp. 65–74. DOI: 10.32620/reks.2023.4.06.

18. Caragea, A., Petersen, P., & Voigtlaender, F. Neural network approximation and estimation of classifiers with classification boundary in a Barron class. Annals of Applied Probability, 2023, vol. 33, no. 4, pp. 3039–3079. DOI: 10.1214/22-AAP1884.

19. Diestel, R. Graph Theory. 5th ed. Springer, 2017, 428 p. DOI: 10.10007/978-3-662-53622-3.

20. Liberzon, D. Switching in Systems and Control. Springer, 2003, 233 p. DOI: 10.10007/978-1-4612-0017-8.

21. Kendall, A., Badrinarayanan, V., & Cipolla, R. Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding. Proceedings of the British Machine Vision Conference (BMVC), 2017, pp. 57.1–57.12. DOI: 10.5244/C.31.57.

22. Barabash, O., Sobchuk, V., Sobchuk, A., Musienko, A., & Laptiev, O. Topological aspects of designing functionally robust wireless sensor networks. Advanced Information Systems, 2025, vol. 9, no. 4, pp. 28–38. DOI: 10.20998/2522-9052.2025.4.05.

Received 21.09.2025, Received in revised form 24.11.2025
Accepted date 18.07.2026, Published date 31.07.2026

МЕТОД ВИЗНАЧЕННЯ ВИКОНАННЯ УМОВ ФУНКЦІОНАЛЬНОЇ СТІЙКОСТІ РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНИХ СИСТЕМ ЗА ТОПОЛОГІЧНОЮ СТРУКТУРОЮ ІЗ ЗАСТОСУВАННЯМ МАШИННОГО НАВЧАННЯ

С. О. Довгий, О. В. Копійка, О. В. Барабаш, Ю. В. Кравченко, А. В. Макарчук

Предметом *дослідження є застосування машинного навчання для виявлення функціональної стійкості розподілених інформаційних систем на основі числових характеристик їхньої топології.* ***Метою*** *даного дослідження є вдосконалення раніше отриманих результатів щодо точності встановлення виконання умови, що забезпечує функціональну стійкість розподілених інформаційних систем.* ***Завданням*** *дослідження є підвищення точності перевірки умови, виконання якої забезпечує функціональну стійкість розглянутої розподіленої інформаційної системи, на основі машинного навчання. Основним* ***результатом*** *дослідження є розроблений ансамблевий класифікатор, заснований на використанні кількох різнотипних моделей машинного навчання, що дало змогу отримати точність, вищу за ту, що отримана в аналогічних роботах, присвячених цій же задачі або суміжним. На відміну від інших робіт, пов'язаних із використанням машинного навчання для оцінки функціональної стійкості інформаційних систем, запропонована в роботі архітектура класифікатора передбачає послідовне використання наборів моделей машинного навчання: перша множина моделей отримує набір вхідних параметрів та оцінює певні інші параметри, а отримані оцінки передаються на вхід інших моделей, які вже й видають кінцевий результат. У такий спосіб можна виявити глибші залежності між вхідними та вихідними параметрами, що й дає змогу отримувати вищу точність. Такий підхід значно відрізняється від того, який, як правило, використовується на практиці, де всі моделі приймають на вхід одні й ті ж параметри, видають на виході одне й те саме, а їх вивід «збивається» в один залежно від того, яка задача вирішується.* ***Висновки.*** *Результатом дослідження є розроблена, навчена та верифікована ансамблева модель машинного навчання, яка на основі вектора кількісних параметрів прогнозує дотримання критерію функціональної стійкості розподіленої інформаційної системи. Отримана в роботі ансамблева модель машинного навчання може бути використана на етапі проєктування інформаційних систем. Моделі машинного навчання, навчені в межах дослідження та використані так, як показано в* *роботі, можуть бути інтегровані в програмне забезпечення, орієнтоване на проєктування та симуляцію роботи інформаційних систем. Не менший інтерес викликає спосіб поєднання цих моделей машинного навчання. Описаний у роботі підхід до організації моделей усередині ансамблю відкриває нові можливості для створення високоточних класифікаторів. Головною* ***науковою новизною*** *отриманих результатів є новий підхід до побудови ансамблевих моделей машинного навчання. На відміну від класичних ансамблів, як-от, наприклад, випадкові ліси, ансамбль, описаний у цій роботі, передбачає застосування набору моделей машинного навчання для обчислення оцінок певних параметрів, які виступають входом для іншого набору моделей машинного навчання. Найімовірніше, саме цей підхід до організації ансамблевої моделі машинного навчання й забезпечив підвищення якості вирішення поставленої задачі, продемонстроване в межах дослідження.*

Ключові слова: *машинне навчання; ансамблі; функціональна стійкість; інформаційні системи; статистичний аналіз.*

Довгий Станіслав Олексійович – д-р фіз.-мат. наук, проф., академік НАН України, зав. відділом, Інститут прикладних систем управління Національної академії наук України, Київ, Україна.

Копійка Олег Валентинович – д-р техн. наук, проф., чл.-кор. НАН України, директор, Інститут прикладних систем управління Національної академії наук України, Київ, Україна.

Барабаш Олег Володимирович – д-р техн. наук, проф. каф. інженерії програмного забезпечення в енергетиці, Національний технічний університет України «Київський Політехнічний Інститут імені Ігоря Сікорського», Київ, Україна.

Кравченко Юрій Васильович – д-р техн. наук, проф., зав. каф. мережевих та інтернет технологій, Київський національний університет імені Тараса Шевченка, Київ, Україна.

Макарчук Андрій Валентинович – асист. каф. інженерії програмного забезпечення в енергетиці, Національний технічний університет України «Київський Політехнічний Інститут імені Ігоря Сікорського», Київ, Україна.

Stanislav Dovgyi – Doctor of Physical and Mathematical Sciences, Professor, Head of the Department of the Institute of Applied Control Systems of the National Academy of Sciences of Ukraine, Kyiv, Ukraine,
e-mail: s.dovgii@gmail.com, ORCID: 0000-0003-1078-0162, Scopus Author ID: 57215821999.

Oleh Kopiika – Doctor of Technical Sciences, Professor, Director of the Institute of Applied Control Systems of the National Academy of Sciences of Ukraine, Kyiv, Ukraine,
e-mail: okopiyka@gmail.com, ORCID: 0000-0003-0189-3915, Scopus Author ID: 59755261800.

Oleg Barabash – Doctor of Technical Sciences, Professor, Professor at the Department of Software Engineering for Power Industry, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine,
e-mail: bar64@ukr.net, ORCID: 0000-0003-1715-0761, Scopus Author ID: 36724076700.

Yurii Kravchenko – Doctor of Technical Sciences, Professor, Head of the Department of Network and Internet Technologies, Taras Shevchenko National University of Kyiv, Kyiv, Ukraine,
e-mail: kr34@ukr.net, ORCID: 0000-0002-0281-4396, Scopus Author ID: 57189330864.

Andriy Makarchuk – PhD Student, Assistant of the Department of Software Engineering for Power Industry, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine,
e-mail: andreymakarh2@gmail.com, ORCID: 0000-0002-6422-7488, Scopus Author ID: 57000526500.