НАВЧАЛЬНО-НАУКОВИЙ
ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

СИСТЕМНІ ДОСЛІДЖЕННЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

МІЖНАРОДНИЙ НАУКОВО–ТЕХНІЧНИЙ ЖУРНАЛ

№ 1 2026

ЗАСНОВАНО У ЛИПНІ 2001 р.

АДРЕСА РЕДАКЦІЇ:

03056, м. Київ,
просп. Берестейський, 37, корп. 35,
НН ІПСА КПІ ім. Ігоря Сікорського
Тел.: **204-81-44**; факс: **204-81-44**
E-mail: journal.iasa@gmail.com
http://journal.iasa.kpi.ua

У номері:

EDUCATIONAL AND RESEARCH
INSTITUTE FOR APPLIED SYSTEM ANALYSIS
OF THE NATIONAL TECHNICAL UNIVERSITY OF UKRAINE
"IGOR SIKORSKY KYIV POLYTECHNIC INSTITUTE"

SYSTEM RESEARCH AND INFORMATION TECHNOLOGIES

INTERNATIONAL SCIENTIFIC AND TECHNICAL JOURNAL

№ 1 2026

IT IS FOUNDED IN JULY 2001

THE EDITION ADDRESS:

03056, Kyiv,
av. Beresteiskyi, 37, building 35,
ER Institute for Applied System Analysis
at the Igor Sikorsky Kyiv Polytechnic Institute
Phone: **204-81-44**; Fax: **204-81-44**
E-mail: journal.iasa@gmail.com
http://journal.iasa.kpi.ua

In the issue:

Шановні читачі!

Навчально-науковий інститут прикладного системного аналізу Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського» видає міжнародний науково-технічний журнал

«СИСТЕМНІ ДОСЛІДЖЕННЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ».

Журнал публікує праці теоретичного та прикладного характеру в широкому спектрі проблем, що стосуються системних досліджень та інформаційних технологій.

Провідні тематичні розділи журналу:

Теоретичні та прикладні проблеми і методи системного аналізу; теоретичні та прикладні проблеми інформатики; автоматизовані системи управління; прогресивні інформаційні технології, високопродуктивні комп'ютерні системи; проблеми прийняття рішень і управління в економічних, технічних, екологічних і соціальних системах; теоретичні та прикладні проблеми інтелектуальних систем підтримання прийняття рішень; проблемно і функціонально орієнтовані комп'ютерні системи та мережі; методи оптимізації, оптимальне управління і теорія ігор; математичні методи, моделі, проблеми і технології дослідження складних систем; методи аналізу та управління системами в умовах ризику і невизначеності; евристичні методи та алгоритми в системному аналізі та управлінні; нові методи в системному аналізі, інформатиці та теорії прийняття рішень; науково-методичні проблеми в освіті.

Головний редактор журналу — завідувач кафедри системного проєктування НН ІПСА КПІ ім. Ігоря Сікорського, професор Мухін Вадим Євгенович.

Журнал «Системні дослідження та інформаційні технології» включено до переліку наукових фахових видань України (категорія «А»).

Журнал «Системні дослідження та інформаційні технології» входить до таких наукометричних баз даних: Scopus, EBSCO, Google Scholar, DOAJ, Index Copernicus, реферативна база даних «Україніка наукова», український реферативний журнал «Джерело», наукова періодика України.

Статті публікуються українською та англійською мовами.

Випусковий редактор **І.О. Пишнограєв**
Редакторка **Л.О. Тарин**
Комп'ютерна верстка **М.А. Марченко**

Рішення Національної ради України з питань телебачення і радіомовлення
№1794 від 21.12.2023. Ідентифікатор медіа R30-02404

Підписано до друку 31.03.2026. Формат 70×108 1/16. Папір офс. Гарнітура Times.
Спосіб друку – цифровий. Ум. друк. арк. 14,411. Обл.-вид. арк. 28,56. Наклад 150 пр. Зам. № 588/50

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Свідоцтво про державну реєстрацію: ДК № 5354 від 25.05.2017 р.
просп. Берестейський, 37, м. Київ, 03056.

Видавництво «Політехніка» КПІ ім. Ігоря Сікорського вул. Політехнічна, 14, корп. 15,
м. Київ, Україна, 03056, тел. (044) 204-81-78

Dear Readers!

Educational and Research Institute for Applied System Analysis of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" is published of the international scientific and technical journal

"SYSTEM RESEARCH AND INFORMATION TECHNOLOGIES".

The Journal is printing works of a theoretical and applied character on a wide spectrum of problems, connected with system researches and information technologies.

The main thematic sections of the Journal are the following:

Theoretical and applied problems and methods of system analysis; theoretical and applied problems of computer science; automated control systems; progressive information technologies, high-efficiency computer systems; decision making and control in economic, technical, ecological and social systems; theoretical and applied problems of intellectual systems for decision making support; problem- and function-oriented computer systems and networks; methods of optimization, optimum control and theory of games; mathematical methods, models, problems and technologies for complex systems research; methods of system analysis and control in conditions of risk and uncertainty; heuristic methods and algorithms in system analysis and control; new methods in system analysis, computer science and theory of decision making; scientific and methodical problems in education.

The editor-in-chief of the Journal is Head of the Department of Systems Design of ER IASA, Igor Sikorsky Kyiv Polytechnic Institute, professor Vadym Mukhin.

The articles to be published in the Journal in Ukrainian and English languages are accepted. Information printed in the Journal is included in the Catalogue of periodicals of Ukraine.

СИСТЕМНІ ДОСЛІДЖЕННЯ ТА ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

1 • 2026

ЗМІСТ

SYSTEM RESEARCH AND INFORMATION TECHNOLOGIES

1 • 2026

CONTENT

АВТОМАТИЗОВАНІ СИСТЕМИ УПРАВЛІННЯ

UDC 681.513
DOI: 10.20535/SRIT.2308-8893.2026.1.01

THE PRINCIPLES OF SYNTHESIZING THE COORDINATING AUTOMATIC CONTROL SYSTEMS

A.A. GURSKIY, A.V. DENISENKO, S.M. DUBNA

Abstract. The problem linked to the model development of the coordinating automatic control system has been solved in this scientific work. Such a task is relevant, as it is linked to increase the operational efficiency of the technological object due to the quality improvement of ratio control. As an object the steam boiler has been considered, in which it is necessary to maintain the appropriate air-fuel ratio. The article describes the stages of the model development for the coordinating automatic control system ensuring the maintenance of a given ratio between controlled variables in transient modes. The support of the set ratio is ensured by the implementation of temporal subordination for control processes. Based on the analysis of modeling results, we have made the conclusion about the expediency of using the coordinating system of automatic control. Thus, the method of stepwise synthesis for coordinating control systems was further developed.

Keywords: coordinating automatic control system, ratio control, steam boiler, stepwise synthesis for the multi-level systems.

INTRODUCTION

Temporal subordination of control processes and coordination of transient is an important factor in order to increase the operational efficiency of some technological objects. In this case, increasing the operating efficiency of some objects is possible by using automatic coordinating control systems [1].

This class of automatic coordinating control systems has a certain feature of synthesis regarding the motions separation mode. At the same time in the systems, it must be provided the subordination of control processes according to temporal priority conditions.

The software development for automated synthesis of coordinating control systems linked with modern means of automation allows to realize the practical application of multi-level control systems in order to improve the operating efficiency of certain objects [2]. In a particular case, we have considered in this work a steam boiler (the DKVR series) for which it is advisable to synthesize the coordinating automatic control system. Such a coordinating system has been developed to improve the quality in control of the air-fuel ratio for efficient combustion of fuel.

PROBLEM STATEMENT

The automation of tuning up process for the multi-level automatic control systems on the temporal subordination of regulation processes is the important task due to the lack of sufficiently simple methods for development systems of the appropriate class [1, 3].

Analyzing the well-known works [1, 4], we can conclude that the calculations are made not based on the appropriate control qualities, but on the transformation of the original standard control algorithm while synthesizing automatic coordinating control systems. Thus, the automated implementation for the tuning up process of coordinating automatic control systems for given control qualities is a relevant task. Accordingly, in order to solve this problem, it has been considered the synthesis of a model for the steam boiler control system for which it is important to improve the quality of regulation for the gas-air ratio in comparison with a typical traditional control system. In this case, the value of the integral indicator for the quality of work (the integral performance indicator) J for the coordinating system ought to be significantly less than the value of the integral indicator J of the typical system. Where $J = \int_0^\infty (\alpha \cdot \left[\left| e(t) / e_{\max} \right| \right] + + \left| \phi(t) / \phi_{\max} \right|) dt$, α is coefficient indicating the temporal coordination of the control processes; $\phi(t)$ is deviations from the ratio of the values of regulated variables; $\phi_{\max}$ is maximum deviations from the ratio of the values of regulated variables; *e(t)* is the deviation of some variable in time from the given value.

Purpose of work is to increase the operational efficiency of the technological objects due to automating the synthesis process of multi-level systems which ensure the temporal subordination of control processes.

To achieve this purpose, it was necessary to conduct a series of experiments linked to the synthesis of multi-level automatic control systems. Such systems ensure coordinated change in the values of controlled variables in transient modes, in the particular case coordinated change of gas consumption and air consumption to the combustion chamber of the steam boiler.

REVIEW OF THE LITERATURE

Automation of the tuning processes for the systems of automatic control and coordination was presented in the work [2]. The refrigeration turbocompressor unit was considered as the control object. For the refrigeration turbocompressor unit it was necessary to change in a coordinated manner the diffuser blades pitch angle and the rotation speed of the turbocompressor shaft depending on cooling water consumption in the condenser.

This paper is considered a steam boiler in which it is necessary to coordinate the change in air consumption relative to the change in gas consumption in front of the combustion chamber during the transients.

There are various typical ratio control systems between fuel consumption and air consumption having both advantages and disadvantages.

The disadvantages of such methods for the regulating of gas-air ratio are that in order to set a given ratio between the gas and air consumption we have only

impact on the position of air consumption control element [5, 6]. The frequency controller can also be used to change the air supply fan performance into the burner.

The relevant systems use the most common air-fuel ratio control scheme in boiler units [6–8]. According to this scheme the fuel consumption is determined depending on the thermal load (steam consumption requirement) and the air consumption is regulated to provide the appropriate air-fuel ratio.

To improve the quality of regulation and liquidate the deviation from the set ratio we need to use not only one control action, as implemented in the standard control schemes. In this case it is necessary to implement an automatic system of coordinating control.

For example, a well-known scheme of the coordinating control system is presented in the scientific work by L.M. Boychuk [9]. This work shows the need to transform typical control schemes during the regulation of ratios and the need to decompose systems at the level when tuning up on the subordination of control processes. The disadvantage is that the coordinating system was not considered for the regulation of ratio between gas consumption and air consumption in the boiler unit. Also, in scientific work by L.M. Boychuk [9] it has not been considered in order to improve the ratio control quality due to simultaneous additional coordinating links between control actions with automatic optimization of the operating modes of objects.

Thus, we have developed in this work a model of the control system for the steam boiler, which has the appropriate coordinating links between the control loops, taking into account the correction of the ratio and optimization of the operating mode of the boiler unit.

MAIN PART

It is known that in fairly frequent cases we strive for autonomy of control loops through dynamic decoupling of control channels. However, independence (i.e., the dynamic decoupling) of control channels from each other may be unnecessary, in particular when ensuring coordination of controlled variables during transients. The term “coordination” is found not only in the field of automatic control and in the field of control the term “coordination” is ambiguous in turn. But in this case coordination is a type of control, along with stabilization and tracking which is associated with the coordination of transients. We are talking exclusively about the coordinating automatic control systems.

The example of various transients is shown in Fig. 1 [1]. Such processes presented in Fig. 1, *b* are difficult to receive in the class of tracking automatic control systems or program system. It is obvious that the transient processes presented in Fig. 1, *b* can occur with a coordinating automatic control system. Such transients in the system can occur if there is a subordination of control processes or a motions separation mode. The different motion trajectory of systems in the space of controlled variables X_1 and X_2 is shown in Fig. 1, *c*. In the motions separation mode at first from the initial point X_0 there is a mode of fast motions to the manifold m_1 and then in a mode of slow motions to the end point $X_к$. In this case the manifold m_1 is determined by the ratio between the variables X_1 and X_2.

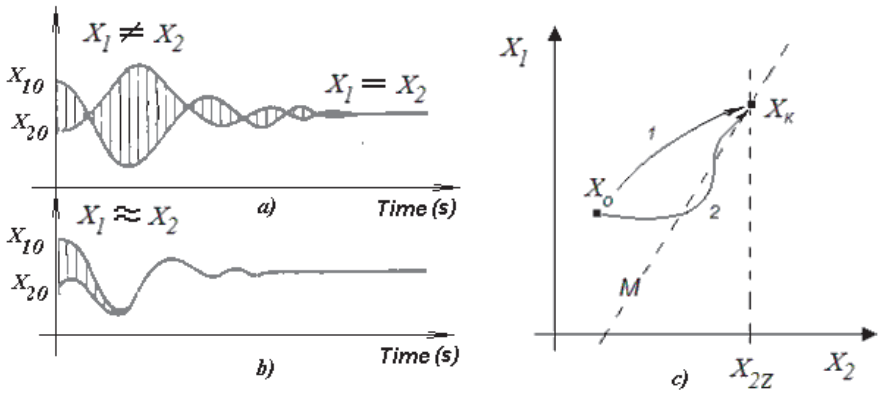


Fig. 1. Examples of transients in a typical automatic control system and in the coordinating automatic control system

It is shown in Fig. 2 the classic (typical) diagram of an automatic control system in which is implemented maintenance the ratio between the controlled variables X_1 and X_2. But the coordinated change of controlled variables X_1 and X_2 is impossible to carry out in the dynamically unconnected system. In order to implement the subordination of control processes and ensure the motions separation mode we need availability cross-links between the control loops shown by the dotted lines in Fig. 2.

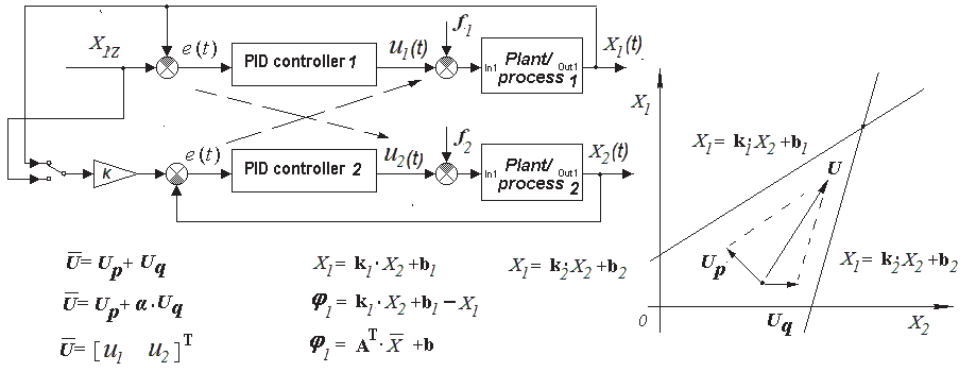


Fig. 2. Structural diagram of the typical control system and geometric formalization of the control task

In such a system we can possess two or more given ratios represented by equations of the form: $X_1 = k \cdot X_2 + b$, where b is a constant. Accordingly, deviations from the ratio is $\phi = k \cdot X_2 - X_1 + b$ or in vector form is $\phi = A^T \cdot \bar{X} + b$, where A^T is the matrix of ratio coefficients. As shown in Fig. 2, in the space of variables X_1 and X_2 the ratios are presented in the form of straight lines possessing one common intersection point M. In such a system (in a system with cross-links) we can distinguish two subsystems. One system is named varying and the other system is stabilizing. One system is linked to elimination of deviations φ_1 from the ratio, the other is connected to deviation φ_2. Herewith the control vector U_p of the varying system is directed towards Manifold One m_1 and the vector U_q of the stabilizing subsystem is directed towards Manifold Two m_2 of regulated ratios. Then the composition of these two subsystems and the addition of these two

vectors lead to the emergence vector U directed to the point for intersection M of the lines. When adding two vectors $U_p + U_q$ we cannot to obtain the subordination of control processes, therefore we need to add coefficient a with this operation. So, in a system configured for the motions separation mode we can show $U = Up + \alpha \cdot Uq$, where α is the coefficient which the temporal subordination of control processes and respectively the motions separation mode.

As an example, we have considered the classic scheme of the automatic air-fuel ratio control system in the steam boiler while stabilizing the pressure of steam in the boiler drum (shown in Fig. 3).

The parameters of the steam boiler control channel are known. The control channel "control action u_{1g} by gas consumption – steam pressure Pn" is described by the 2nd order transfer function in this form $W_{up}(s) = \dfrac{k_1 \cdot e^{-\tau \cdot s}}{(T_1 \cdot s + 1) \cdot (T_2 \cdot s + 1)}$, where τ is time delay; *T1, T2* are the time constants for the boiler. The dynamic properties of the boiler we can also describe by transfer function of the 1st order aperiodic link with a delay $W_{up}(s) = \dfrac{k_1 \cdot e^{-\tau \cdot s}}{T \cdot s + 1}$, where, T = 200–300 sec., τ/T = 0.025 [10–12]. Accordingly, the inertia of the air consumption control channel is significantly less than the inertia of the steam pressure control channel. The air consumption control channel is "the control action u_{2V} on the air consumption linked with a change in the angle of the guide blades for the blower fan – the air consumption G_V before the economizer and the boiler furnace". This channel is described by the nonlinear static characteristic of a blowing centrifugal fan.

The tuning parameters for the boiler PID controller we can determine based on the parameters of the control channels and optimize by the appropriate criterion for the quality of system operation.

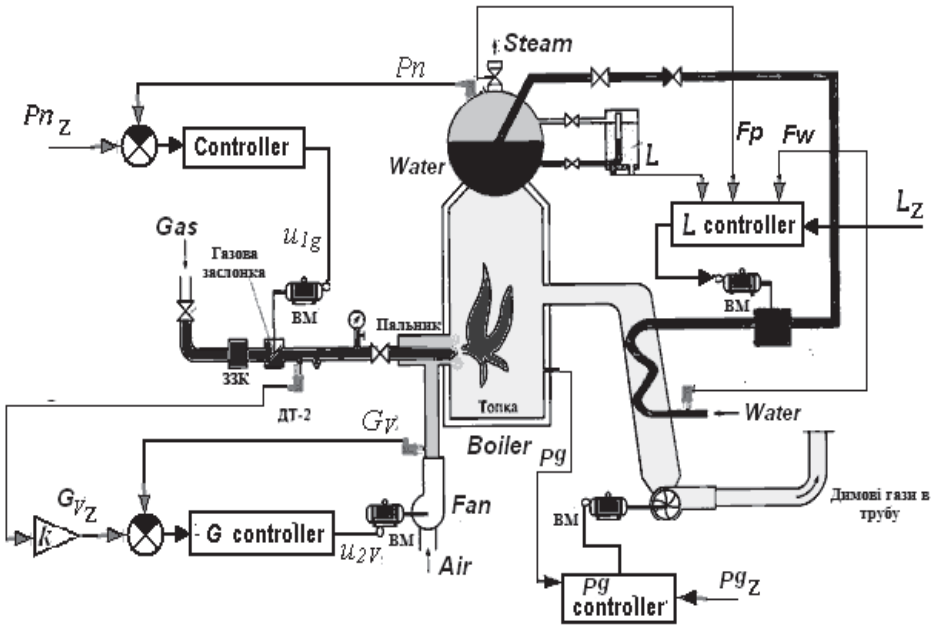
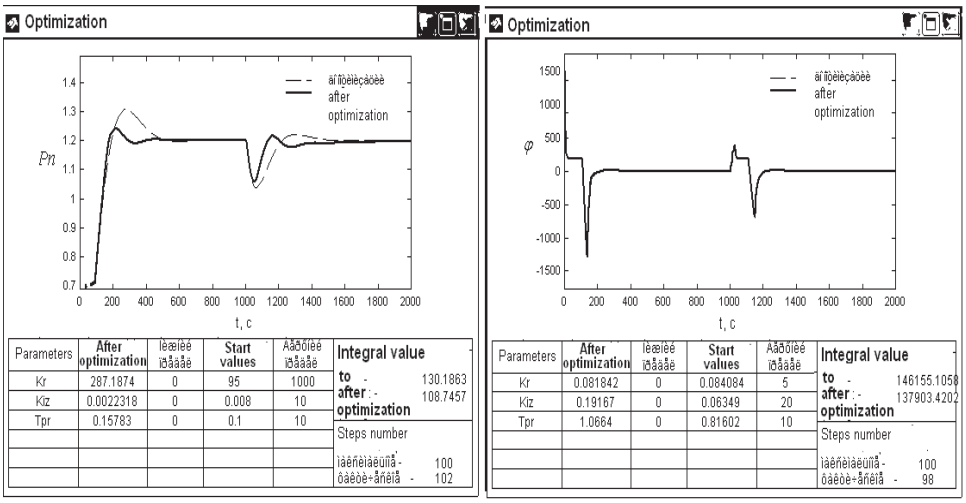


Fig. 3. Structural and technological diagram of the steam boiler

Thus, in the MATLAB\Simulink 5.2 software environment we have optimized the steam pressure PID controller parameters, then we have carried out optimization of the parameters for the ratio controller according to such integral criterion of system quality: $J_{02} = \int_0^\infty |e(t)|dt$, where *e(t)* is the difference between the actual and set values of the controlled variable (Fig. 4).



a *b*

Fig. 4. The optimization results of parameters for the PID steam pressure controller in the boiler drum (*a*) and the ratio controller (*b*)

In order to synthesize such coordinating automatic control system, we can decompose the coordinating system into two systems. System one is the single-level coordinating automatic control system presented in Fig. 5. System Two is the stabilizing system and also the 2nd level of the coordinating system.

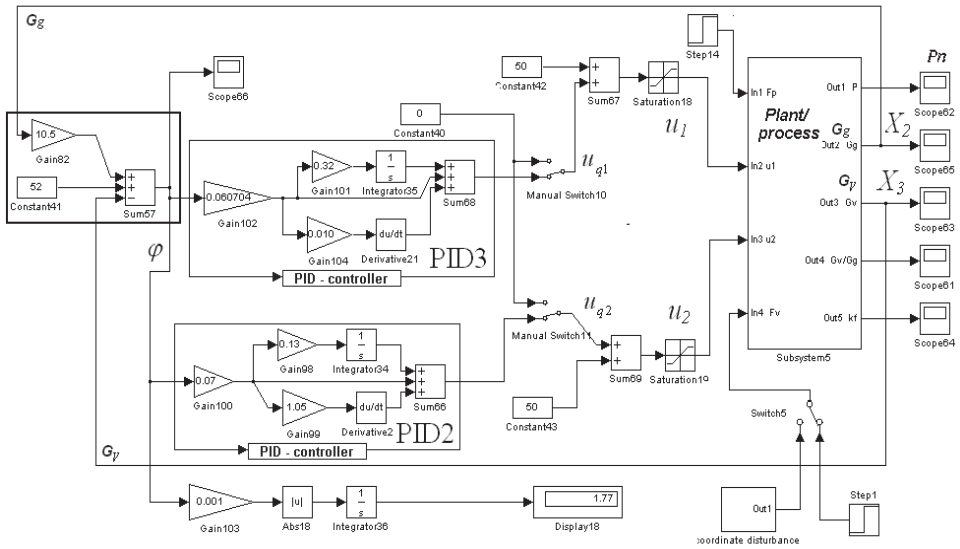


Fig. 5. Block diagram of the model for a single-level coordinating automatic control system presented by using MATLAB\Simulink

As shown in Fig. 5, two PID controllers are used simultaneously to process the deviation from the ratio of controlled variables. This is a distinctive feature of systems for this class. It is obvious that if one of the controllers (*PID2 or PID3*) is excluded from the system and one of the control actions is fixed at a certain value, then the quality of the ratio control will noticeably deteriorate. In this case such system will convert from a coordinating system into a tracking one or into a regular stabilization one. But when two PID-controllers operate simultaneously and the values of two controlled variables change towards decreasing of deviation from the given ratio, then the quality of control ratio is noticeably higher.

The control law of a single-level coordinating system is as follows:

$$\overline{u}_q = \begin{bmatrix} u_{q1} \\ u_{q2} \end{bmatrix} = \begin{bmatrix} k_1 \cdot (1 + (k_{11}/p) + k_{12} \cdot p \\ k_2 \cdot (1 + (k_{21}/p) + k_{22} \cdot p \end{bmatrix} \cdot \phi, \quad (1)$$

where $\phi = A^T \cdot \overline{X} + b$, $A^T = \begin{bmatrix} -1 & k \end{bmatrix}$, $\overline{X} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ is a vector of controlled variables, k_1, k_2, k_{12}, k_{11}, k_{21}, k_{22} are the tuning parameters of PID controllers.

The tuning parameters for the *PID2* controller are determined as in the previous case based on the parameters of the control channel $u_2 - X_3$, and the tuning parameters for the *PID3* are determined based on the parameters of the channel $u_1 - X_2$.

As a result of modeling for a single-level coordinating system we have determined that depending on the initial conditions or under the various actions ${}^{<2>}u_1$, ${}^{<2>}u_2$, deviations from the ratio are eliminated for 12–15 seconds and whilst various steady-state values of the controlled variables are implemented. We can see it in Fig. 6 showing the graphs of various transients by the deviation ϕ from the ratio and by air consumption X_3.

In order to ensure stabilization of the controlled variables at the specified value, it is necessary to add to the system an extra stabilizing level, in which the deviations from another specified ratio can be processed.

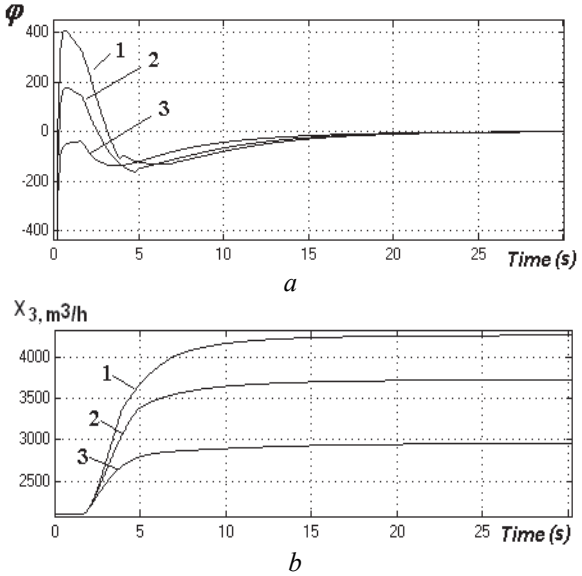


Fig. 6. Curves of transients by deviation from the specified ratio between the controlled variables (*a*) and step responses curves of the controlled variable X3 (*b*)

The structural diagram of the stabilization system model is presented in Fig. 7. In the stabilizing system of this model we can see some additional connection in addition to the PID controller. This connection sets the required value of the controlled variable X_3 (air consumption), relative to the control action u_1 on the controlled variable X_2 (gas consumption). Such additional cross-link with the transmission coefficient (k_u) improves the quality of control by steam pressure in the boiler drum (according to the variable X_1).

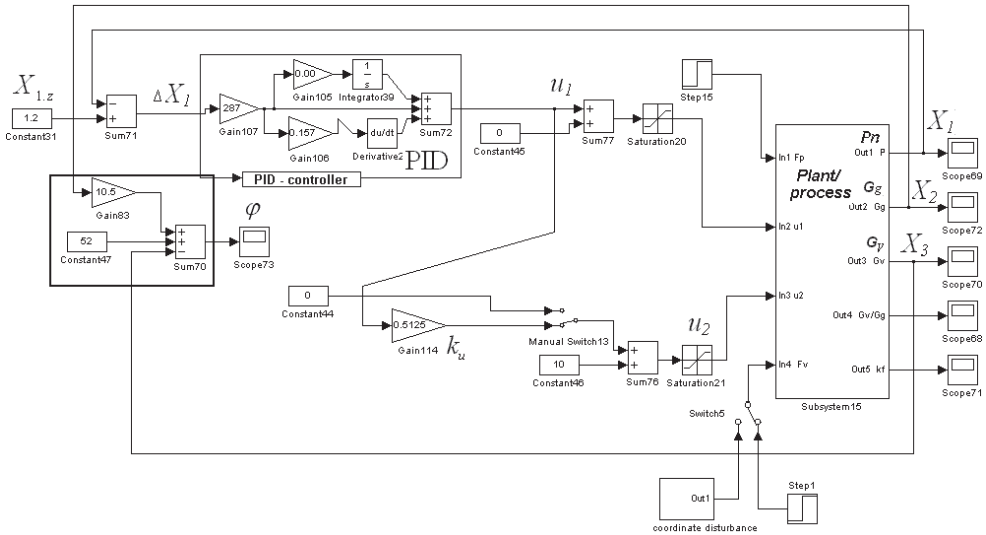


Fig. 7. Block diagram of the model for a single-level stabilizing control system presented by using MATLAB\Simulink

The systems shown in Figs. 5 and 7 have the same control object (plant or process) therefore we can combine them into one system. The structural diagram for the model of such a combined system is presented in Fig. 8.

To synthesize such a system and to move from the coordinating level of the system to the stabilizing level we need to select the leading control variable and, respectively the slave control one. In this case, X_2 is as the leading control variable and, respectively X_3 is the slave control one. Thus, another control loop appears in order to stabilize the variables at a set value. By Fig. 8, such a control loop appears when the switch KL1 is closed. But with such a system the control loop of steam pressure with the *PID* controller is the stabilizing subsystem, and when the KL1 contact is closed, we can see an additional internal control loop to stabilize the variable X_2.

The control law of this coordinating system is two-level and it can be represented as follows:

$$\overline{u} = \overline{u}_q + \overline{u}_p = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \overline{u}_q + \begin{bmatrix} k_3(1 + (k_{31}/p) + k_{32} \cdot p) \\ k_{u1-u2} \cdot k_3(1 + (k_{31}/p) + k_{32} \cdot p) \end{bmatrix} \cdot (X_{1.z} - X_1),$$

where $\overline{u}_q$ is the control vector of the 1st level, according to expression (1), k_3, k_{31}, k_{32} are the tuning parameters of the 2nd level, where k_3, k_1, k_2 are the parameters of the 1st control level, according to expression (1). They determine the time subordination of the control processes and, respectively, the mode separation of motions in the system.

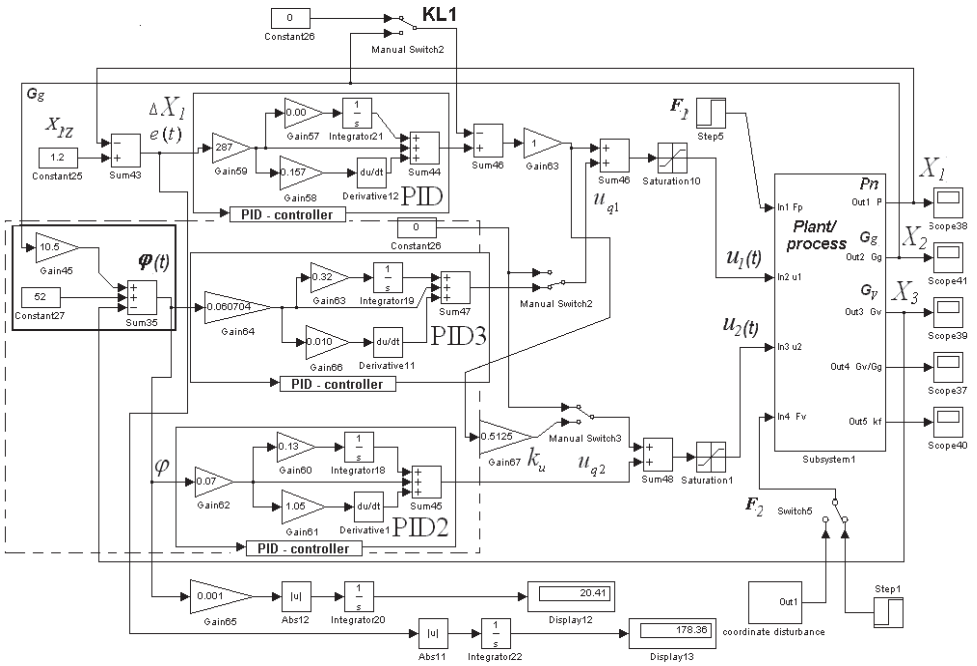
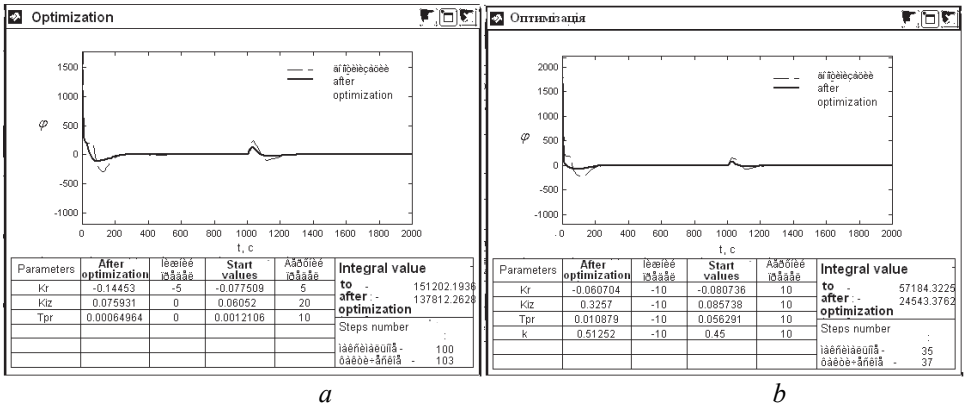


Fig. 8. Block diagram of the model for a two-level coordinated automatic control system presented by using the MATLAB\Simulink environment

In such a system the tuning parameters of the *PID* and *PID2* controllers have already optimized at the previous stage during the synthesis of the classical (typical) system. At the next stage the tuning parameters of the *PID 3* controller have optimized by the following integral criterion of quality for system operation: $J_{02} = \int_0^\infty \left(|\phi(t)| + \alpha \cdot |e(t)|\right)dt,$ where α is the weighting factor.

The results of the optimization for the PID 3 controller parameters are presented in Fig. 9. The presented transient process (Fig. 9) demonstrates how significant the quality of the air-gas ratio control has improved when comparing with the results presented in Fig. 4.

At the last stage the value of the coefficient k_{u1-u2} of the cross-link between the control actions u_1 and u_2 has optimized by the previous integral criterion, which has provided an extra improvement in the quality of control.



a *b*

Fig. 9. The parameter optimizing results for the coordinating level of the system

RESEARCH RESULTS

In this scientific article it has been carried out the modeling of various automatic control systems in the MATLAB\Simulink 5.2 software environment. The simulation results are presented in Fig. 10. It is evident from Fig. 10, *b* that in terms of the quality of ratio φ regulation the typical automatic control system (graph 1) functions significantly worse than the coordinating system (curves 2, 3), which contains additional cross-links between the control loops. However, the quality of steam pressure control is insignificantly worse while coordinating the controlled variables, unlike the typical control system. It is noticeable by comparing curves 1 and 3 in Fig. 10, *a*.

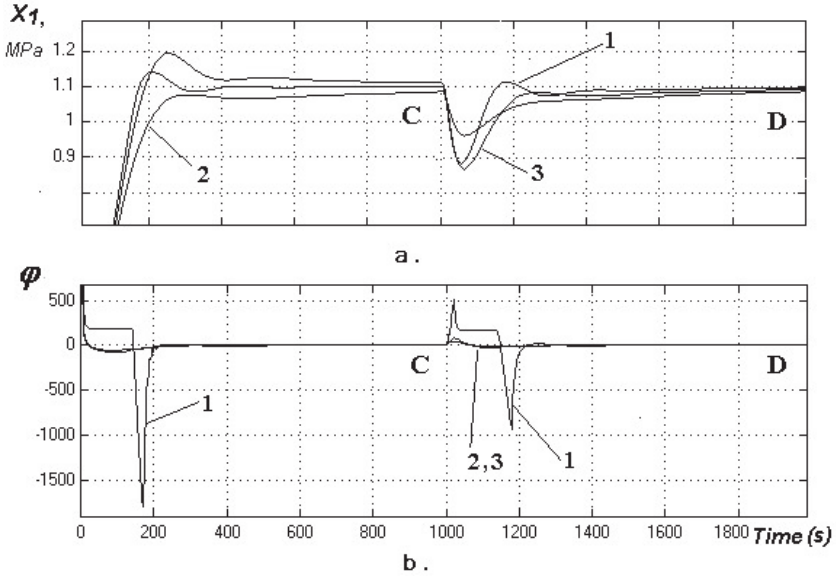
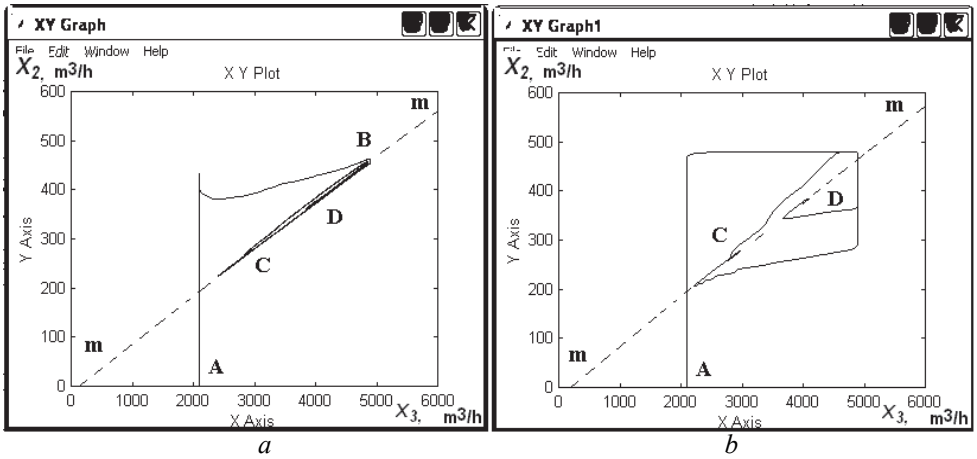


Fig. 10. The transients in various control systems, 1 is dynamic characteristic in the traditional control system; 2, 3 are dynamic characteristics in coordinating automatic control systems

The motions separation mode is clearly seen when considering motion of the system in space of the controlled variables X_2, X_3 (gas-air) shown in Fig. 11, *a*, *b*. The motion trajectory of the coordinating system is shown in Fig. 11, *a*. As shown in Fig. 11, *a* as per the motions separation mode at first the mode of fast motions comes from point A to point B located on the manifold m submitted by the dashed line. Then in the m ode of slow motions along the manifold m of the controlled ratio we can see the movement from point B to point C representing the final steady state. Due to the disturbing effect in such system the movement occurs from point C to point D along the manifold *m.* In Fig. 10 we have indicated points C and D in the graphs of the transients appropriating to steady conditions.

The motions separation mode is absent in a typical system of the steam boiler control. This is noticeable when considering the movement of the system in the space of controlled variables X_2, X_3. It is presented in Fig. 11, *b*. As shown in Fig. 11, *b*, any transients and, hence, any movement of the system is associated with away from the manifold m (submitted by the dashed line) which represents the set of optimal states for gas combustion in the boiler furnace.



a *b*

Fig. 11. Movement of the coordinating (*a*) and typical systems (*b*) in the space of controlled variables

DISCUSSION

As a result of studies and based on simulation results, we have determined that during the coordinating automatic control system the quality of regulation for the gas-air ratio is significantly higher. But at the same time the coordinated change of gas consumption relative to air consumption and vice versa has not worsened significantly the regulation quality of steam pressure in the boiler drum with appropriate cross-link. This cross-link exists with the transfer coefficient k_u between the control actions for gas and air consumption control. Thus, as per the optimization results presented in Fig. 9, *a*, *b* we have the cross-link with the coefficient k_{u1-u2} allowing reducing significantly the value of the appropriate integral indicator for quality of system operation.

Ultimately, we have got the structural diagram of automatic control system shown in Fig. 12 as per the diagram of a model for the coordinating system presented by means of MATLAB\Simulink 5.2 (Fig. 8). In Fig. 12 we can see such a system differing mainly from the typical control system of the steam boiler by the presence of cross-links with links 3 and 4 and the peculiarity of tuning on the temporal subordination of control processes appropriate the motions separation mode. The diagram is also shown the automatic optimizer (AO) correcting the gas-air ratio.

Hence, the coefficient of the gas-air ratio (determining the required air flow depending on the G_g gas consumption) can be correcting through the signal u_o which has come from the output of the automatic optimizer 8. The automatic optimizer 8 forms the u_o correction signal depending on the oxygen content v_o in the flue gases and in the indicator of gas (fuel) combustion efficiency. So automatic optimizer 8 is contained in the optimization level of the system, PID controller 2 and unit 3 are contained in the stabilization level, and PID 4 and 5 controllers are contained in the coordination level of the system.

Notably, as per the structural diagram presented in Fig. 12, the complete composition of the control system must contain the underpressure *Py* control loops in the boiler furnace and the control loops of water level in the boiler drum.

Such control loops cause certain disturbances in the considered system for the air and gas consumption in the boiler furnace taken into account during the synthesis of system models.

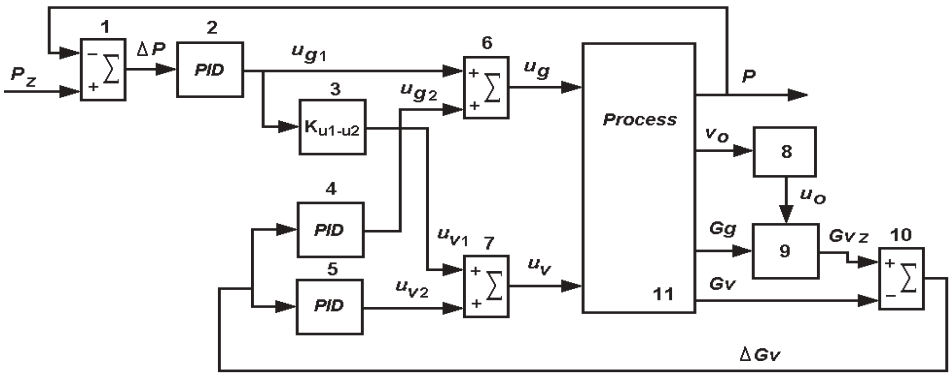


Fig. 12. Block diagram of the coordinating automatic control system

CONCLUSIONS

This article focuses on the development of a stepwise synthesis for the multi-level coordinating system providing the air-gas (air-fuel) ratio control in the steam boiler. It has been considered this stepwise synthesis of the control system, which is necessary for subsequent verification of the automation for tuning process of the appropriate systems. So, the method of synthesizing the coordinating automatic control systems has been further developed.

These conducted studies have confirmed the fundamental suitability of the algorithms for step-by-step tuning of the coordinating systems on the motions separation mode. The developed models allow approaching the definite solution of a practical problem linked to the automated tuning of control systems based on the up-to-date programs connected with the latest automation tools.

Thus, the further research should be directly related to the automation for tuning process of the coordinating automatic control systems.

REFERENCES

1. L.M. Boichuk, *Sintez koordinirujushhih sistem avtomaticheskogo upravlenija [Synthesis of coordinating automatic control systems]*. M.: Energoatomizdat, 1991, 160 p.
2. A.A. Gurskiy, A.E. Goncharenko, S.M. Dubna, “Algorithms for tuning of the coordinating automatic control systems,” *Radio electronics, computer science, control*, no. 1, pp. 190–199, 2020. doi: https://doi.org /10.15588/1607-3274 -2020-1-19
3. L.M. Boychuk, “Two Approaches to the Structural Synthesis of Control Systems: Dynamical Filtration and Automatic Compensation,” *Journal of Automation and Information Sciences*, vol. 30, issue 6, pp. 13–22, 1998. doi: 10.1615/ JAutomatInfScien.v30.i6.20
4. L.M. Boychuk, “Separation of motions and steady balance in the balance dynamic systems of the open type,” *Journal of Automation*, issue 6, pp. 55–64, 1991.
5. A. Sniders, T. Komass, “Invariant method of load independent pressure control in steam boiler,” *Electrical, Control and Communication Engineering*, vol. 1, issue 1, pp. 5–10, 2012. doi: https://doi.org /10.2478/v10314-012-0001-4

6. G. Liang, Y. Bai, G. Yang, W. Li, "A Boiler Combustion Control System with Combustion Ratio Optimisation and Soft Measurements," *Measurement and Control*, vol. 43, issue 4, pp. 112–115, 2010. doi: https://doi.org/10.1177/002029401004300403
7. W. Chatlatanagulchai, S. Rhienprayoon, K. Yaovaja, K. Wannatong, "Air/fuel ratio control in diesel-dual-fuel engine by varying throttle, EGR valve, and total fuel," *SAE Technical Paper*, no. 2010-01-2200, 16 p. 2010. doi: https://doi.org/10.4271/2010-01-2200
8. X. Jiao, J. Zhang, T. Shen, J. Kako, "Adaptive air–fuel ratio control scheme and its experimental validations for port-injected spark ignition engines," *International Journal of Adaptive Control and Signal processing*, vol. 29, issue 1, pp. 41–63, 2015. doi: 10.1002/acs.2456
9. L.M. Boychuk, "Sintez avtomaticheskih sistem koordiniruyushego upravleniya s relejnoj stabiliziruyushej korrekciej [Synthesis of automatic control systems with relay stabilizing correction]," *Automation*, no. 2, pp. 63–71, 1993.
10. T. Lee, E. Han, U.-C. Moon, K.Y. Lee, "Supplementary control of air–fuel ratio using dynamic matrix control for thermal power plant emission," *Energies*, 13(1), 226, 2020. doi: https://doi.org/10.3390/en13010226
11. J.R. Rodriguez Vasquez, R. Rivas Perez, J.J. Sotomayor Moriano, J.R. Peran González, "Advanced control system of the steam pressure in a fire-tube boiler," *IFAC Proceedings Volumes*, vol. 41, issue 2, pp. 11028–11033, 2008. doi: https://doi.org/10.3182/20080706-5-KR-1001.01868
12. S.G. Dukelow, B.G. Liptak, X. Cheng, R.H. Meeker Jr., "8.6 Boiler Control and Optimization," *Control and Optimization of Unit Operations*, pp. 1572–1631, 2006.

Received 20.02.2025

INFORMATION ON THE ARTICLE

Alexander A. Gurskiy, ORCID: 0000-0001-5158-2125, Odesa National University of Technology, Ukraine, e-mail: gurskiya2017@gmail.com

Andrey V. Denisenko, ORCID: 0000-0002-8610-0082, National University "Odessa Polytechnics", Ukraine, e-mail: denisenko.a.v@op.edu.ua

Sergey M. Dubna, ORCID: 0009-0002-1181-9110, Odesa National University of Technology, Ukraine, e-mail: dubna_s@ukr.net

ПРИНЦИПИ СИНТЕЗУ КООРДИНУВАЛЬНИХ СИСТЕМ АВТОМАТИЧНОГО УПРАВЛІННЯ / О.О. Гурський, А.В. Денисенко, С.М. Дубна

Анотація. Вирішено завдання, що пов'язане з розробленням моделей координувальних систем автоматичного управління. Це завдання актуальне, оскільки спрямоване на підвищення ефективності функціонування технологічних об'єктів завдяки покращенню якості регулювання співвідношення між регульованими змінними. Як приклад розглянуто паровий котел, у якому необхідно підтримувати відповідне співвідношення газ–повітря. Представлено етапи розроблення моделі координувальної системи автоматичного управління, яка забезпечує підтримання заданого співвідношення між регульованими змінними у перехідних режимах. Підтримання заданого співвідношення забезпечується через реалізацію підпорядкованості процесів регулювання. Проведені експерименти показали принципову придатність визначеного поетапного алгоритму синтезу, який можна покласти в основу автоматизації процесу налаштування систем на підпорядкованість процесів регулювання.

Ключові слова: координувальна система автоматичного управління, регулювання співвідношення, паровий котел, поетапне налаштування багаторівневих систем управління.

UDC 004:415.2
DOI: 10.20535/SRIT.2308-8893.2026.1.02

OPTIMIZING MICROSERVICES DESIGN PATTERN: MAXIMIZING COMMUNICATION SPEED AND PROLONGING APPLICATION LONGEVITY

Y.E. KOVALOV, Y.V. BOYKO

Abstract. Microservice oriented application design obtained popularity in the past years. Most researchers investigated some aspects in microservice design for implementing application functionality. Little research considered the core functionality of microservices. This research investigates how to construct a microservice communication system by yourself in detail. The results should assist developers and architects to construct their own microservice applications, use less amount of frameworks and therefore prolong overall microservice system life cycle. The standard TCP/IP connection and embedded libraries were used to construct the communication system without using any additional frameworks. As a practical application of this methodology a microservice core system was implemented with a minimum number of microservices to perform performance testing. The measured application layer communication speed turned out to exceed the speed in real application because of database operation limitations. The implemented microservice core system is intended to be used in financial commercial applications as well as in further scientific investigations.

Keywords: microservice, application design, communication speed, domain-driven, monolith application, application life cycle, event, message bus, osi model, delivery guarantee, application layer.

INTRODUCTION

One of the modern programming design patterns is microservice architecture. It was derived from both service-oriented architecture (SOA) and event-driven architecture (EDA) [1]. Many people have heard of it. Some developers used it. The microservice architecture has something in common with challenges in any other application: how quickly it works, how easy to develop and support, how to integrate different parts together etc.

Although some researchers have paid attention to different practical aspects of microservice programming design [2–12], much less research has investigated the core functionality of microservice applications. In this research a practical methodology for creating microservice applications from scratch is represented. First of all, it is a way microservices interact with each other to work as a whole.

Past research mostly used standard communication frameworks and technologies. Although they are good for most practical tasks, the lack of developer control may reduce their longevity. In this research the original event delivering system was created and tested in detail. This approach gives developers full control

for communications as the most important part of the microservice system thus drastically increasing the life cycle of application.

The research questions we are supposed to answer in this study are as follows:

- What are the main benefits and drawbacks of using applications with microservice design?
- What are the main parts of microservice applications?
- How to take advantage of microservice design and evade drawbacks?
- How to construct a microservices communication system by yourself?

This paper has four parts. First it reviews the literature relevant to microservices design. Then the research methodology is presented and outlines the main parts of microservice application design. In the next step practical results of research are summarized and discussed. The paper concludes with a summary of results and further research.

LITERATURE REVIEW

There have been many architectural paradigms developed over time in computer systems. Some of these approaches, such as the widely known object-oriented programming, have significantly influenced computer languages. Another, such as structured programming, changed how developers write programs by prioritizing clear structure, improving readability, and making development and maintenance easier. Despite their differences, these patterns shared a common goal: to facilitate clearer program structures for human comprehension, thereby streamlining the software development and maintenance.

On the contrary, machine code is straightforward and rudimentary, consisting of a sequence of instructions that direct the processor on how to execute tasks. Unlike humans, machines don't rely on structured systems since they possess the capacity to retain every detail. The creation of computer languages, paradigms, and architectures was solely a human endeavor. It reflects human limitations, yet superior comprehension of grammar, particularly in the context of code syntax.

One of the most contemporary design patterns is the microservice architecture. Growing number of programmers and system architects interact with this technology. Developers engaging with microservices often question why this approach, what drawbacks exist in other technologies, and where to start.

Monolith applications

A so-called monolith application is a program although it consists of different modules, but they are working in a tight-coupled way. One module could not operate without another or at least there’s a central part of the application on which other parts are intertwined. Often it had a single start point and single database to hold information for the entire system. The human brain is an example of such a system in terms the brain has different intertwined tightly coupled parts.

No wonder that most legacy applications and computer systems were architected as monoliths, as this architecture was the simplest way to develop. The monolith system has many advantages that should be considered when creating other systems. First of all, it offers a high speed of interaction between its parts because most interlinks are performed using fast memory operations. It also has

no issues with transaction clarity, as it usually has a single database with built-in transaction processing.

A well-constructed monolith system can perform better than purely designed microservices architecture. So, before making a decision, let's take into account all the strengths and weaknesses of these architectures.

Domain-driven design

Domain-driven design (DDD) was first introduced by Eric Evans in his book "Domain-Driven Design: Tackling Complexity in the Heart of Software" [13]. Over time, this approach became a well-known architecture, with many authors providing their perspectives on this design [14].

The point is that large computer systems are usually developed by multiple teams of developers, not just one developer or a single team. This approach enables the quick development of complex applications because each team works simultaneously and independently.

DDD can be used in both monolithic and microservice applications [3]. It represents an idea on how to divide large systems into parts. Each team focuses on one part of the system independently. Later the teams stitch it together using one of the integration mechanisms.

In this work, some ideas and interpretation of this design were implemented on our way in making the microservice system.

Taking advantage of microservice architecture

A microservice-driven application was a further development of ideas on how to break down a complex system. Once again, it had something to do with human thinking. Our society is an example of such a system. Each individual is responsible for themselves and has their own skills and abilities, acting just like a single microservice in an application. Similarly, society can function as a whole because people communicate with each other and coordinate their efforts to achieve their goals.

One of the main goals of the microservice approach is to prolong the longevity of an application. Although monolith systems are easier to create, they are much harder to support. People tend to migrate to new technologies, computer languages, frameworks etc. Consequently, the number of developers willing to support legacy applications inevitably decreases over time.

As usual, renewing a monolith application means it must be rewritten from scratch. This interrupts the life cycle of old (legacy) applications, even though they may still satisfy all needs. Society wastes time and money by doing this. This issue occurs not only in software development but also throughout our daily lives.

Consider a simple object as a fridge. Some people throw it away just because it broke down or has an outdated design. But what if somebody makes a fridge like a Lego set allowing you to repair, upgrade, and change its parts and design as needed? I think this approach will inspire you to keep your fridge indefinitely. With this mindset, society could use our limited resources much more efficiently. Additionally, this approach is crucial for mitigating climate change.

Interest in the term "microservice" has grown since 2014 [15]. Many authors have made an effort to clarify what it is and how to work with it [2–6]. Table 1 illustrates how they addressed the challenges of microservice design.

Table 1. Microservice design approaches

	A Design Study of Microservice Architecture on White Label Travel Platform [3]	Application of Microservice Architecture in Commodity ERP Financial System [2]	Enhancing Saga Pattern for Distributed Transactions within a Microservices Architecture [4]	A Microservice-based Software Architecture for Improving the Availability of Dental Health Records [5]	Introducing Cloud-Assisted Micro-Service-Based Software Development Framework for Healthcare Systems [6]
Messages format					Soap (Simple Object Access Protocol) or JSON
How microservices interact with each other	Asynchronous Message bus	RestAPI inter service communication	RestApi, Kafka		RestApi, AMQP (Asynchronous Message Queuing Protocol)
How microservices interact with frontend	Synchronous Api gateway through RestAPI and Api gateway on some microservices. So some microservices have two-way communication	RestApi communication with Api gateway. Api gateway makes a connection through api with microservices	RestApi	RestApi	Service access tier through RestApi
How it was proposed to divide into microservices	Domain-driven design	Business logic			
Microservice's internal structure	Domain-driven design		Clean architecture		
Working with a database	Several microservices share the same database	One microservice – one database	One microservice – one database	One microservice – one database	One microservice – one database
Microservices connection information and discovery		Services registry configuration center			
What happens if microservice restarts					
What happens on microservice's lost communication		Special guard "Sentinel"			
How to find errors		Special guard "Sentinel"	logback encoder		
Interaction with other systems (including monolith systems)					
Authentication		Through separate microservice named Service Management Center (use spring Security + auth2 0)		Through separate microservice named Dental Service Provider	
Authorization		Several microservices- Authorized authentication service and User micro service		Through separate microservice named Dental Service Provider	
Complex (multiple microservices) transaction implementation			Saga, quota cache, eventual commit sync service		

But how to not find yourself lost in this variety of approaches, in other words how to catch a fish in a microservice's cocktail? This research will find answers to this question.

RESEARCH METHODOLOGY

When one microservice sends a message to others it may not know recipients. It just emits some mark of its own activity with useful information. So, it is more reasonable to name such an activity as an event.

The objective of this research is to create microservice application cores in a step-by-step manner. It includes communication, event format, events routing, database, event delivery, event processing and API gateway.

Communication

Communication is the imperative part of the microservice application since it relates to all microservices. Most applications under investigation used standard communication technology and frameworks (see Table1) such as RestApi for synchronous communication. For asynchronous communication top known methods were Kafka [16] and RabbitMQ [17].

As already mentioned above, monolith applications had an advantage of huge communication speed between the parts of the system. So, it is very important to provide a quick connection between microservices, otherwise it becomes a bottleneck of our application. Thus, it must be something lightweight and quick. RestApi may be a good choice, but it isn't lightweight and isn't quick enough as was concluded during interconnection speed investigation.

Of course, it is possible to use one of the open source message systems. But both methods that were mentioned above didn't guarantee that messages or events would not be duplicated. Additional efforts must be implemented to avoid duplication. Moreover, what if something goes wrong with them? Microservice systems may need some additional functions and don't need others. Maybe future developed microservice would require exact computer language and this open source system would not support it. Moreover, open source communication systems may suffer from a security vulnerability that would affect the whole system. And because it is open source all around would know about that. And finally, the lifecycle of microservice systems and open source systems can be different. You would have a problem if the open source system team stops supporting it.

If you want to do something good — do it yourself, especially when it is comparably easy to perform. It's reasonable to rely on technologies and methods that are supposed to survive longer than our microservice system's lifetime. Let's try figuring it out using the Open System Interconnection model (OSI) [18]. This model was developed in 1984 by the International Organization for Standardization (ISO). Let's scrutinize this model and try to make the best of it.

So, there are seven layers in this model (Fig. 1). The lowest level is Physical, the highest is Application. The lower layer the quicker communication but the harder to write a managed program.

The lower three layers are media layers. They deal with hardware and low-level software such as drivers, on-board programs and operating system's services. They are way too low for our purposes. But there was something important

we must keep in mind about physical connections – there were duplex connections and usually there were input and output low-level data caches. It means that a communication system must be constructed with embedded support of simultaneous input and output data flow to maximize performance.

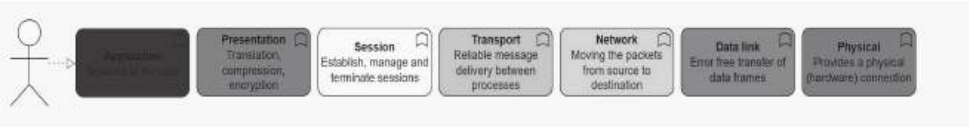


Fig. 1. OSI model

The upper four layers are host layers. And the lowest in this group (Transport) is good enough for our needs. But that's just a model. How about practical implementation?

TCP/IP model is a more practical implementation of communication suite compared to OSI.

It used four layers instead of seven layers in OSI model (Table 2) [19]. Now it is a communication standard supported almost by all devices, systems, operating systems, computer languages etc. It survived for a long time, and is supposed to survive even longer.

Table 2. OSI vs. TCP/IP model

OSI	TCP/IP
Application	Application
Presentation	
Session	
Transport	Transport
Network	Internet
Data link	Link
Physical	

There were two main protocols on a Transport layer – transport control protocol (TCP) and user datagram protocol (UDP). The UDP protocol is stateless while TCP establishes and holds connections. UDP is quicker but the main disadvantage, meaning that it was less applicable for our purposes, was that it didn't track the sequence of data. The messages should be accepted by our receiver microservice exactly in succession they were transmitted. Thus, TCP protocol is best for our application core.

Event format

It is reasonable to use the message format that most of the computer languages and operating systems are using now and is expected to use in the future. One of the well-known formats that fit this demand is text format.

Well-known standard for encoding object information and data interchange in a text form is JavaScript Object Notation (JSON) [20]. It is commonly used in web application programs. Many computer languages have libraries to parse it.

It is not very difficult to write such a library on your own for legacy systems. It has basic types for Number, String, Boolean, Array, Object. From our point of view this format is good enough to be used as an event format, but you can choose any other text format you prefer.

Events routing

When events are emitting it is not the microservice's responsibility to know recipients. Otherwise, they would be tightly coupled, which is considered as a bad smell for microservices architecture [21]. So, the most reasonable approach is to implement subscriptions and event types (let's name it EventId). Microservice's subscriptions should somehow be remembered. Events of specified EventId must be sent to its recipients.

As it was concluded the microservice would be implemented for this functionality. Let's name it EventBroker. It works like a delivery system (Fig. 2).

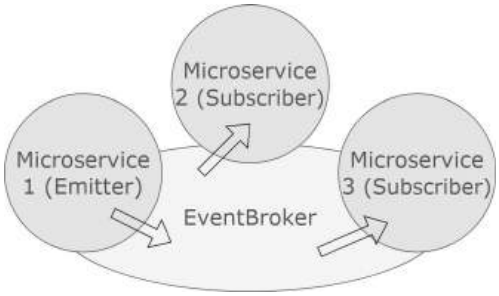


Fig. 2. Events routing

At the startup EventBroker subscribes itself to receive subscription events (startup record in a subscription table). Every microservice first emits a subscription event either on the startup or at any time. On receiving, EventBroker fixes this information in the database. From now on all events of that EventId will be forwarding to its subscribers. Fig. 2 portrayed stances in which microservice 1 is an event emitter and microservices 2 and 3 are subscribers for that EventId.

Sadly, EventBroker in this architecture is a single point of error, which is considered a bad smell in miscroservice's design. If EventBroker stops working the whole system will freeze. But let's keep in mind that there is already a single point of error in any microservice system – network. Everything goes wrong if the network stops working. And EventBroker is an extension of the network with responsibility to deliver events. If the network layout is super resilient and has several reserved lines it is possible to create several EvenBroker microservices for each line to avoid a single point of error. This would not change our methodology.

The EventBroker is a central coordinator of all events and simultaneously logs keeper. If something goes wrong in our microservice system – having logs is vital for debugging and moreover it is considered a bad smell if the microservice system doesn't have a central log keeper. Our goal is to create a microservice system, not just a bunch of microservices.

EventBroker establishes tcp/ip connection with every microservice we have. In order to maximize performance, it must actively use parallel execution tech-

nologies of your favorite computer language. Thus, there are two simultaneously running tasks for each microservice (input and output events processing).

Database

Every system needs data storage. The microservice system has a lot of microservices that have to be loose coupled. Shared databases in this approach are a way out of line. So, it is imperative for all microservices to have their own database.

It is up to developers to choose the database and framework to deal with. It can be Sql, NoSql, object-oriented and so on. Of course, there is no reason for one development team to use different database types in one project.

Event delivery guarantee – not less than once, not more than once

Communication is commonly used by many microservices, so it is reasonable to build a communication library for each computer language used. There are several reasons that may affect microservice system event delivery. First of all, it is a connection issue and microservice application problem.

Connection issues are caused by unstable network connection. If this happens one or several tcp/ip connections between EvenBroker and microservice may interrupt. The communication library must track this somehow and initialize connection renew. The simplest way to implement this functionality is periodic connection testing with small data amounts.

Microservice applications may be affected by following: be under the maintenance, in the restart process, hangs up etc. As a result, events suffer from not being delivered at all or being delivered several times. How to mitigate these artifact behaviors?

There is no problem at all sending events again and again until the subscriber accepts it. But to avoid duplication every event should be unique and easily identified. And keeping in mind that any microservice (including EventBroker) can be interrupted for reasons mentioned above, the only way to provide this is keeping the event state in our storage – database. Using database transaction technique, it is possible to keep atomicity of database changes even during restart.

So, every microservice must have a pool of outgoing and incoming events in the database with a unique id for every outgoing. When a microservice receives an event it first checks against the database if the event with that id already was registered. It registers events in the database if not. It replies to the sender with "Event with this id was received" either after registration or when it was already found in the database (Fig. 3).

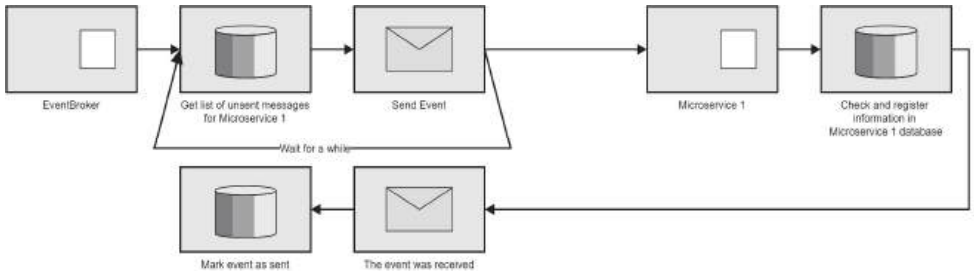


Fig. 3. Event delivery

Event processing

The event delivery approach proposed above is not only about the resilience of the microservice system but also gives the opportunity to perform event processing atomically.

Microservice is like a black box. It is not necessary for you to know how exactly it is working unless you send and receive events and understand its structure. But what if something happens in the middle between events being received and sent in other words when events are under processing?

In the proposed approach we can put the whole sequence into one transaction (Fig. 4).

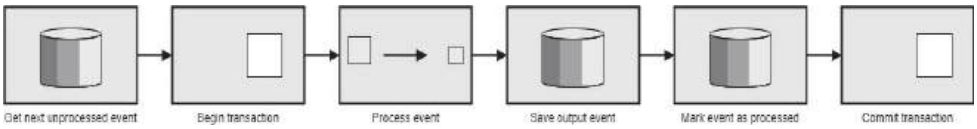


Fig. 4. Event Processing

Thus, either during processing the answer will be saved or the database state will be returned to the initial state just after the input event was received.

API gateway

As was mentioned earlier, RestApi today is the most widely used protocol for web applications. In our microservice architecture it can be used to connect other applications, frontend applications and so on. All we need is just another microservice (name it APIGateway), connected to EventBroker but with specific behavior.

APIGateway performs interlink between internal tcp/ip and Rest Api protocols adding caching capabilities. It is also a good place for user authentication since other parts of a system could be hidden from the outer world.

PRACTICAL RESULTS OF RESEARCH AND DISCUSSION

The microservice system was implemented based on the research methodology presented above. It was coded with c# for .net multiplatform program language. Entity framework, System.Net.Sockets library and MySql database were used as helper components.

In addition to EventBroker and APIGateway, a test microservice named “Ping” was implemented. With help of these microservices some vital microservice core parameters were measured.

Application layer speed

First of all, the application layer speed was measured in isolation without database operations. Using APIGateway we generated an event for Ping microservice. On event arrival Ping microservice generates 50000 messages of specified size addressed to EventBroker.

The software measured both times consumed to publish and receive messages. Two computers were used to measure performance with 1 Gigabit per

second network connection between them. First computer, with Ping microservice installed, is equipped with CPU "Intel Core I5-6400 2.7 Ghz" and 8 gigabytes of memory on Windows 10 system. Second, with ApiGateway, EventBroker microservices and MySql has CPU "Intel Core I5-6600K 3.5 Ghz" and 16 gigabytes of memory on Windows 10 system. One thread was used on both publish and receive sides.

It turned out the bottleneck in application layer performance was CPU on a publish computer with Ping microservice launched (one CPU kernel was fully engaged). Thus, consumed and published times were identical with few deviations, for that reason our graph shows only consumed time transformed to speed without consideration to publish time (Fig. 5).

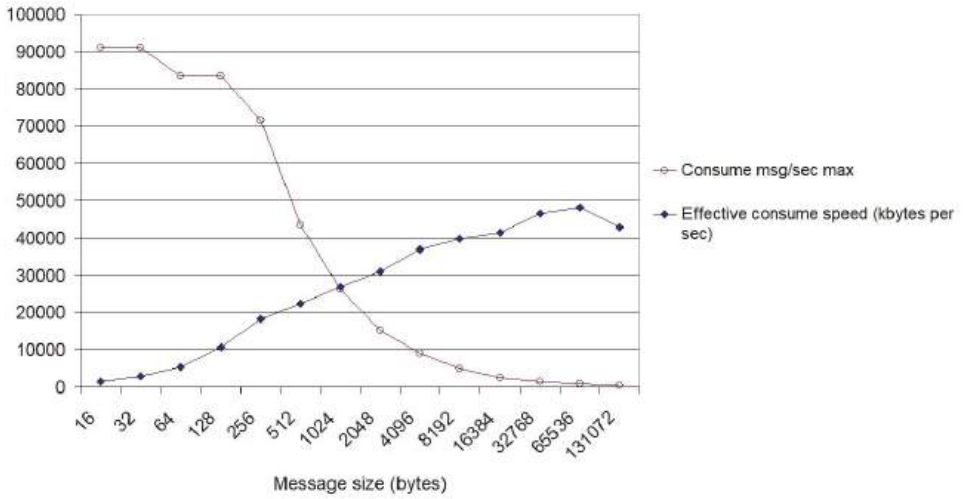


Fig. 5. Application layer speed

In addition, the effective consume speed shows that only a part of network connection bandwidth was engaged, as for 1 gigabit per second network theoretical bandwidth limit is about 122000 kilobytes per second (Fig. 5). But as already mentioned above this is because of a CPU bottleneck on a publishing computer.

For the most practical cases this speed is more than enough and exceeds performance of other universal message brokers already mentioned in paragraph "Communication".

Events processing speed

During the next phase let's add database operations to our experimental environment. The computer hardware configuration is the same as mentioned in a previous step. The results are on Fig. 6. Event size here is 398 bytes.

The events flood publisher here is microservice Ping. It generates 100 events and sends them to event receiver EventBroker microservice. How quickly EventBroker can save to the database was measured. EventBroker can engage multiple threads (c# Task library) to process events. With the number of tasks 4 the maximum saving rate was reached in this configuration. This happens when 100 % CPU usage was registered on EventBroker microservice (number of tasks 4 and 5).

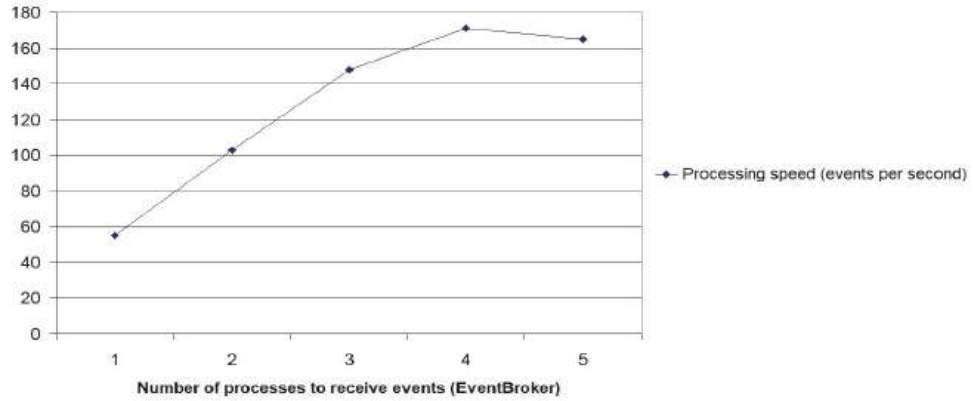


Fig. 6. Database was engaged in speed measurement

In the next phase the different event sizes were investigated with the same hardware configuration. Number of tasks engaged at EventBroker was fixed to 4 (Fig. 7).

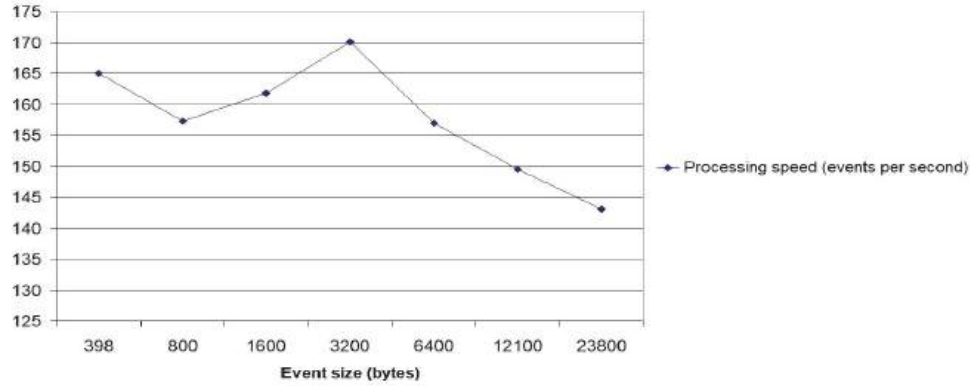


Fig. 7. Event size dependence in database engaged configuration

If we trace over the graph it is clear that processing speed is almost the same with few deviations until event size 6400 bytes is reached and then it slowly drops down. Thus, for this system architecture it is preferable to generate less amounts of bigger events to increase performance.

CONCLUSIONS AND FUTURE WORK

The microservice system with direct network connection application level was constructed. In this system special microservice EventBroker was implemented to be in charge of delivering events to other microservices. It also makes sure every microservice receives not less than one event, not more than one event. The event processing speed was measured in isolation without database operations and in real situations with database transactions.

The performance of application layer speed without database operations exceeds the universal message broker's speed even with one CPU engaged and reaches 90000 events per second. It is also possible to perform multithreading events processing for even better results.

The performance in real applications depends on how quickly the database operations are performed. For the experimental part a MySql database was involved. It shows a maximum productivity of about 170 events per second in the test environment. Multithreading environment boosts the performance. Thus, in a real environment with powerful servers the high system productivity is expected.

Self written microservices communication library gives developers control over application lifetime as the most important part of this design pattern.

The goal of the further research may include testing different database types in terms of productivity increasing. Also, the important parts of the microservice application should be investigated such as authentication, authorization.

REFERENCES

1. O.O. Petrenko, "A comparison of architecture types of services," *System Research and Information Technologies*, no. 4, pp. 48–62, 2015.
2. Wan Yan, Fu Shuai, "Application of Microservice Architecture in Commodity ERP Financial System," *International Journal of Computer Theory and Engineering*, vol. 14, no. 4, November 2022, pp. 168–173. doi: 10.7763/IJCTE.2022.V14.1324
3. I Gede Rahmat Wijaya, Ahmad Nurul Fajar, "A Design Study of Microservice Architecture on White Label Travel Platform," *Journal of System and Management Sciences*, vol.13, no. 4, pp. 249–264, 2023. doi: 10.33168/JSMS.2023.0415
4. Eman Daraghmi, Cheng-Pu Zhang, Shyan-Ming Yuan, "Enhancing Saga Pattern for Distributed Transactions within a Microservices Architecture," *Applied Sciences (Switzerland)*, vol. 12, issue 12, June-2 2022, Article number 6242, pp. 1–24. doi: 10.3390/app12126242
5. Juan Arcila-Diaz, Carlos Valdivia, "A Microservice-based Software Architecture for Improving the Availability of Dental Health Records," *International Journal of Computing*, vol. 21, issue 4, pp. 475–481, 2022. doi: 10.47839/ijc.21.4.2783
6. John Zaki, S.M. Riazul Islam, Norah Saleh Alghamdi, M. Abdullah-Al-Wadud, Kyung-Sup Kwak, "Introducing Cloud-Assisted Micro-Service-Based Software Development Framework for Healthcare Systems," *IEEE Access*, vol. 10, March 22, 2022, pp. 33332–33348. doi: 10.1109/ACCESS.2022.3161455
7. Zhongyi Lu, Declan T. Delaney, David Lillis, "A Survey on Microservices Trust Models for Open Systems," *IEEE Access*, vol. 11, March 23, 2023, pp. 28840–28855. doi: 10.1109/ACCESS.2023.3260147
8. Randa Ahmad Al-Wadi, Adi A. Maaita, "Authentication and Role-Based Microservice Architecture: A Generic Performance-Centric Design," *Journal of Advances in Information Technology*, vol. 14, no. 4, pp. 758–768, 2023. doi: 10.12720/jait.14.4.758-768
9. Ahmet Vedat Tokmak, Akhan Akbulut, Cagatay Catal, "Boosting the visibility of services in microservice architecture," *Cluster Computing*, vol. 27, pp. 3099–3111, September 18, 2023. doi: 10.1007/s10586-023-04132-5
10. Iury Araujo, Nuno Antunes, Marco Vieira, "Evaluation of Machine Learning for Intrusion Detection in Microservice Applications," *LADC '23: Proceedings of the 12th Latin-American Symposium on Dependable and Secure Computing*, pp. 126–135, October 17, 2023. doi: 10.1145/3615366.3615375
11. Wesley K.G. Assunção, Jacob Krüger, Sébastien Mosser, Sofiane Selaoui, "How do microservices evolve? An empirical analysis of changes in open-source microservice repositories," *The Journal of Systems & Software*, vol. 204, October 2023, 111788, pp. 1–14. doi: 10.1016/j.jss.2023.111788
12. Francisco Ponce, Jacopo Soldani, Hernán Astudillo, Antonio Brogi, "Smells and Refactorings for Microservices Security: A Multivocal Literature Review," *Journal*

of Systems and Software, vol. 192, October 2022, 111393, pp. 1–18. doi: 10.1016/j.jss.2022.111393
13. Eric Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software*. Boston: Addison-Wesley, 2004, 501 p.
14. Vaughn Vernon, *Domain-Driven Design Distilled*. Boston: Addison-Wesley, 2016, 136 p.
15. Victor Velepucha, Pamela Flores, "A Survey on Microservices Architecture Principles, Patterns and Migration Challenges," *IEEE Access*, vol. 11, 15 August 2023, pp. 88339–88358. doi: 10.1109/ACCESS.2023.3305687
16. *Kafka documentation*. Available: https://kafka.apache.org/documentation/
17. *RabbitMQ documentation.* Available: https://www.rabbitmq.com/docs/documentation
18. *OSI Model.* Available: https://www.javatpoint.com/osi-model
19. *What is TCP/IP?* Available: https://www.techtarget.com/searchnetworking/definition/TCP-IP
20. *Introducing JSON.* Available: https://www.json.org/json-en.html
21. Tomas Cerny, Amr S. Abdelfattah, Abdullah Al Maruf, Andrea Janes, Davide Taibi, "Catalog and detection techniques of microservice anti-patterns and bad smells: A tertiary study," *The Journal of Systems & Software*, 206 (5):111829, December 2023, pp. 2–43. doi: 10.1016/j.jss.2023.111829

Received 17.09.2024

INFORMATION ON THE ARTICLE

Yurii E. Kovalov, ORCID: 0009-0002-1649-751X, Taras Shevchenko National University of Kyiv, Ukraine, e-mail: yuk123@meta.ua

Yuriy V. Boyko, ORCID: 0000-0003-1417-7424, Taras Shevchenko National University of Kyiv, Ukraine, e-mail: yuriyboyko@knu.ua

ОПТИМІЗАЦІЯ ШАБЛОНУ СТВОРЕННЯ МІКРОСЕРВІСІВ: МАКСИМІЗАЦІЯ ШВИДКОСТІ ЗВ'ЯЗКУ ТА ПОДОВЖЕННЯ ЧАСУ ЖИТТЯ СИСТЕМИ / Ю.Е. Ковальов, Ю.В. Бойко

Анотація. Останнім часом набуло популярності створення застосунків із використанням технології мікросервісів. Більшість дослідників аналізують можливості цієї технології для реалізації функціонування застосунку. Небагато досліджень присвячено функціям ядра функціонування мікросервісної системи. Мета дослідження — детальний розбір можливості самостійної побудови системи зв'язку мікросервісної системи. Результатами дослідження можуть скористатися розробники й архітектори програмного забезпечення для побудови своїх мікросервісних систем таким чином, щоб у них задіювалася менша кількість програмних шаблонів, таким чином збільшуючи життєвий цикл системи. Комунікаційну систему побудовано на базі стандартного TCP/IP з'єднання та вбудованих бібліотек для роботи із ним без використання додаткових програмних шаблонів. Як приклад практичного використання цієї методології розроблено ядро мікросервісної системи із мінімальною кількістю мікросервісів, необхідних для перевірки швидкості роботи. Як виявилося, виміряна швидкість зв'язку рівня застосунку перевищує швидкість у реальній ситуації через обмеження швидкості роботи із базою даних. Заплановано використати реалізоване ядро мікросервісної системи для розроблення комерційних фінансових застосунків та у ході проведення подальших досліджень.

Ключові слова: мікросервіс, архітектура програми, швидкість з'єднання, домен-орієнтований, монолітний застосунок, життєвий цикл програми, подія, шина повідомлень, модель osi, гарантування доставки, рівень застосунку.

UDC 004.932.2
DOI: 10.20535/SRIT.2308-8893.2026.1.03

MULTIMODAL SYSTEM FOR SKIN CANCER DETECTION

V. SYDORSKYI, I. KRASHENYI, O. YAKUBENKO

Abstract. Melanoma detection is vital for early diagnosis and effective treatment. While deep learning models on dermoscopic images have shown promise, they require specialized equipment, limiting their use in broader clinical settings. This study introduces a multi-modal melanoma detection system using conventional photo images, making it more accessible and versatile. Our system integrates image data with tabular metadata, such as patient demographics and lesion characteristics, to improve detection accuracy. It employs a multi-modal neural network combining image and metadata processing and supports a two-step model for cases with or without metadata. A three-stage pipeline further refines predictions by boosting algorithms and enhancing performance. To address the challenges of a highly imbalanced dataset, specific techniques were implemented to ensure robust training. An ablation study evaluated recent vision architectures, boosting algorithms, and loss functions, achieving a peak Partial ROC AUC of 0.18068 (0.2 maximum) and top-15 retrieval sensitivity of 0.78371. Results demonstrate that integrating photo images with metadata in a structured, multi-stage pipeline yields significant performance improvements. This system advances melanoma detection by providing a scalable, equipment-independent solution suitable for diverse healthcare environments, bridging the gap between specialized and general clinical practices.

Keywords: medical image classification, computer vision, gradient boosting, deep neural networks, clinical decision support systems.

INTRODUCTION

Skin cancer is one of the most commonly diagnosed types of cancer, posing a significant public health concern due to its high incidence rates and the risk of severe complications if not detected early [1]. The most effective approach to managing skin cancer is through early detection and prevention [2]. Despite substantial progress in medical imaging and diagnostic technologies, reliably and efficiently detecting melanoma remains challenging. Traditional diagnostic methods depend heavily on dermatologists' expertise, which can be subjective and vary between practitioners. As a result, there is increasing interest in utilizing deep learning techniques to automate and improve the accuracy of skin cancer detection [3]. Progress in dataset curation and related classification challenges has demonstrated potential for fast and accurate skin cancer detection [4]. Deep learning methods have recently

gained popularity and have been shown to improve skin cancer detection performance [5–10]. The effectiveness of these models relies heavily on the quality of their training datasets and the limitations of deep learning methods. Additionally, early detection may not always be feasible due to lengthy manual diagnostic procedures [11], and many low-income individuals cannot afford these options. This highlights the need to develop approaches that surpass human diagnostics, providing faster and more accurate results.

Deep learning has become a gold standard for skin cancer classification. Classical approaches using CNN architectures such as ResNets [12], DenseNets [13], and convolutional networks enhanced with attention mechanisms [5] have been widely adopted. More complex pipelines combine segmentation, feature extraction, and attention-based classification neural networks [3]. Synthetic data generated by GANs has proven effective in enhancing model performance [14]. Multi-modal approaches incorporating diverse data modalities further improve classification [15, 16].

Several studies employ hybrid approaches, blending deep learning with traditional machine learning algorithms. For instance, [17] uses VGG16 for feature extraction, followed by XGBoost for final image classification. This study also leverages synthetic data as part of its data augmentation strategy. Another noteworthy approach involves skin cancer detection using genetic data. In [18], various machine learning algorithms, including KNN, SVM, and XGBoost, are applied to classify melanoma. Similarly, [19] explores the use of NIR spectroscopy as input data, utilizing XGBoost, LightGBM, 1D-CNNs, and other machine and deep learning methods for classification.

Despite progress, several research gaps remain:

- Development of complex multi-modal systems integrating different modalities (e.g., image and tabular data) in parallel or sequential architectures.
- Optimization of multi-modal neural networks for datasets with and without meta-features.
- Adaptation to challenging imaging conditions, such as images captured using mobile devices, addressing data imbalance and quality variations.

In this study, we propose a novel framework for melanoma detection that integrates visual, structural, and lesion metadata with patient information such as age and sex. Our solution combines multi-modal neural networks for processing visual and metadata inputs, and a gradient-boosting model for metadata analysis unified within a three-stage pipeline. Additionally, we introduce a two-step training methodology to accommodate datasets with varying metadata availability. Advanced training techniques and feature engineering are applied to address class imbalance, ensuring robust and efficient melanoma detection. Finally, we enhance system performance through two stages of feature engineering, enabling robust and efficient melanoma detection.

MATERIALS AND METHODS

Data. This research utilizes several data sources:

- Data from ISIC 2024 Kaggle Challenge [20] - Main Data.
- Data from ISIC Archive [21] - ISIC Archive Data.
- Artificially generated image data [22] - Generated Data.

The primary dataset from the ISIC 2024 Kaggle Challenge is the foundation for most of the training and validation processes. However, additional data sources play a crucial role in enhancing the proposed system, particularly in improving the neural network performance. However, diverse datasets introduce domain shifts and varying feature subsets, creating challenges in harmonizing and effectively integrating the information. The proper fusion of these heterogeneous data sources represents a key contribution to our work, enabling robust performance across different data modalities and domains.

Competition Data. The ISIC 2024 Kaggle Challenge dataset [20] includes images and metadata from single-lesion crops extracted from 3D total body photos (TBP) [23]. This dataset presents several challenges:

- Lower data quality compared to dermatoscopy images. The images resemble close-up smartphone photos, making them highly relevant for telehealth applications, where patients often submit similar-quality images (Fig. 1).
- Different labeling confidence. The dataset includes two categories of labels: "strongly-labeled tiles," verified through histopathology, and "weakly-labeled tiles," which were not biopsied and considered benign by a doctor's assessment.
- Severe class imbalance. The dataset contained 401059 tiles, where 400666 (99.902 %) are benign, and 393 (0.098 %) are malignant.

Each image represents 15x15 mm of skin area but may come with a slightly varying resolution centered around 133 pixels (Fig. 2). Besides image data, the dataset includes metadata about patients, tile location, image characteristics, and extracted features derived from and extracted features derived from [24] and [25].

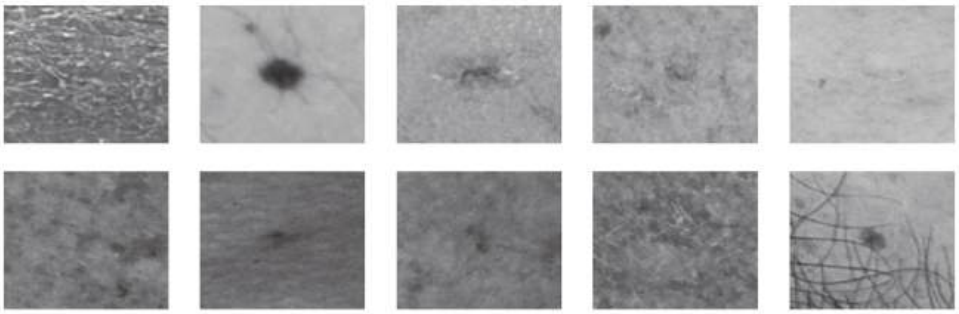


Fig. 1. 1st row – benign; 2nd row – malignant images from ISIC 2024 Kaggle Challenge

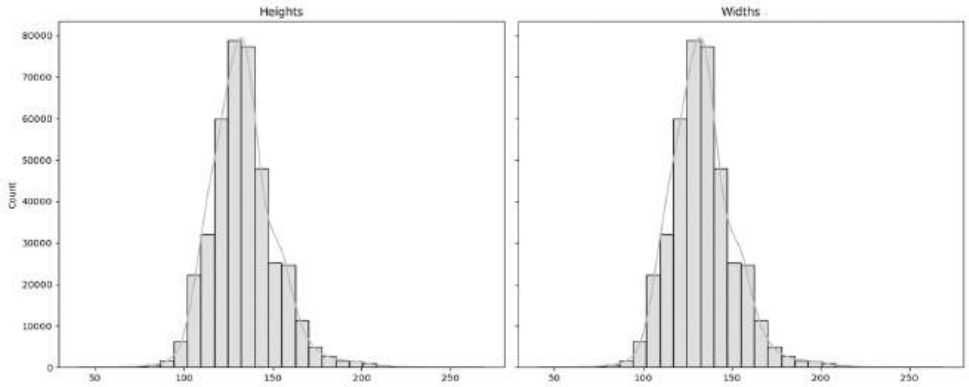


Fig. 2. Distribution of image shapes in Data from ISIC 2024 Kaggle Challenge

ISIC Archive Data. The ISIC Archive dataset [21] contains 81.722 images accompanied by metadata. However, the dataset is highly unstructured due to its compilation from various data sources and competitions. For this study, most of the available meta-features are disregarded, and only patient information, target labels, and images are utilized for system development.

To avoid potential data leakage, all patients included in the ISIC 2024 Kaggle Challenge dataset [20] are excluded from the ISIC Archive data. Additionally, images lacking explicit benign/malignant labels are removed. After such a filtration ISIC Archive contains 71080 images, where 61910 (87.099 %) are benign, and 9170 (12.901 %) are malignant. This dataset has approximately 5.6 times fewer total images than [20], but it contains 23.3 times more malignant images.

The primary challenge with the ISIC Archive dataset lies in its substantial data diversity. It aggregates skin lesion tiles from various sources, including dermatoscopy and standard photo images. The images differ significantly in size, aspect ratio, scale, and padding, influenced by the medical devices used to capture them (Fig. 3). Most of the images are dermatoscopy and come into higher resolution – median height resolution is 3024, and width is 2016 (Fig. 4). This diversity introduces a significant domain shift compared to the ISIC 2024 Kaggle Challenge dataset [20]. Despite these challenges, the ISIC Archive dataset is a valuable source of malignant images, addressing their severe undersampling in the primary dataset. Its inclusion enriches the data diversity, improving model robustness and generalization.

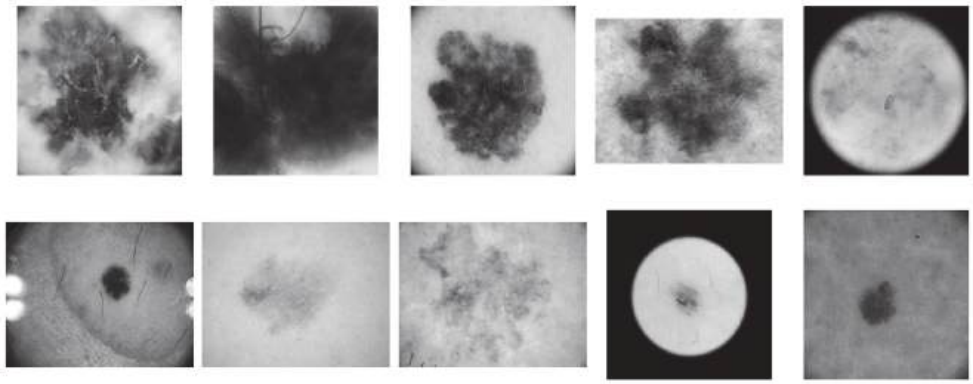


Fig. 3. 1st row – benign; 2nd row – malignant images from ISIC Archive

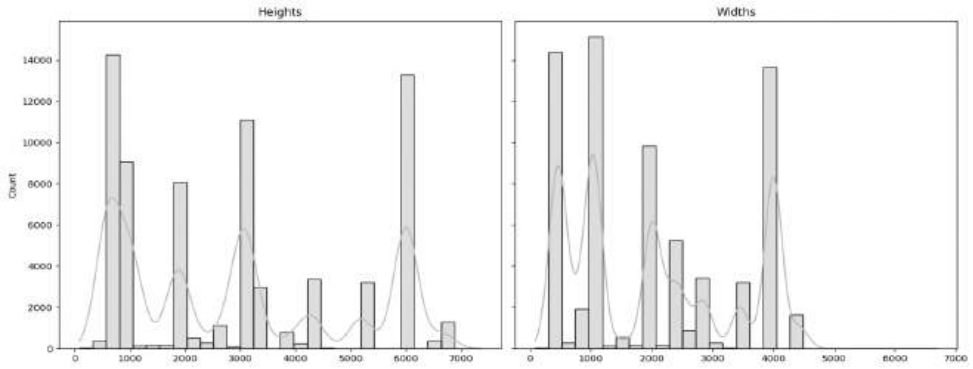


Fig. 4. Distribution of image shapes in ISIC Archive

Generated Data. The generated image dataset from [22] was created using the Stable Diffusion 2 model [26]. It consists of 6.012 images, with a nearly equal distribution of classes: 3.012 malignant and 3.000 benign images. All images are standardized to a resolution of 512×512 pixels. While the generated images can often be distinguished by artifacts and smooth textures characteristic of generative models (Fig. 5), the model performs well in preserving malignant lesions' shapes and visual characteristics. This fidelity provides valuable information that can enhance the training of deep learning models by supplementing the limited data of malignant cases in real-world datasets.

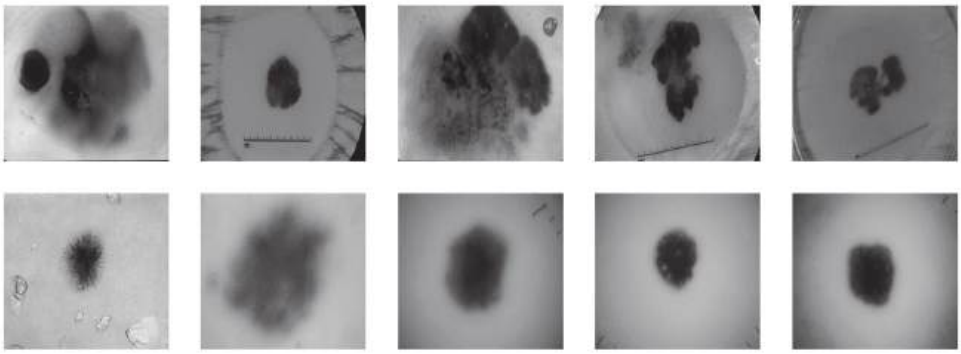


Fig. 5. 1st row – benign; 2nd row – malignant images from Generated Dataset

METHODS

This subsection describes the proposed system, including its components, models, optimization processes, evaluation metrics, and validation procedures.

Metrics. To evaluate the proposed system and models, the following metrics are utilized:

- ROC AUC. A standard metric to measure the overall performance of a binary classification model.
- Partial ROC AUC [27]. This metric calculates the area under the ROC curve only for True Positive Rates (TPR) above 80%. The score ranges from 0.0 to 0.2, emphasizing performance in high-sensitivity regions critical for clinical applications.
- Top-15 retrieval sensitivity [28]. This metric is the most appropriate to real clinic scenarios when a dermatologist has limited time for a patient and should pay attention to the most suspicious lesions [29].

Out of fold (OOF), Mean Fold metrics will be reported.

Validation. To evaluate the models, a classical 5-fold cross-validation approach is used. Folds are stratified based on the target label (benign/malignant), and no patient overlaps across folds are ensured.

In cases where a two-stage training approach is employed (first stage: ISIC Archive + Main + Generated Data; second stage: only Main Data), the datasets are split separately, and respective folds are merged afterward.

For models using only tabular data, validation is repeated five times with different random seeds, and average scores are reported. Hyperparameter tuning for tabular models is performed using the Optuna algorithm [30], with the tuning strategy discussed in the Tabular Approach.

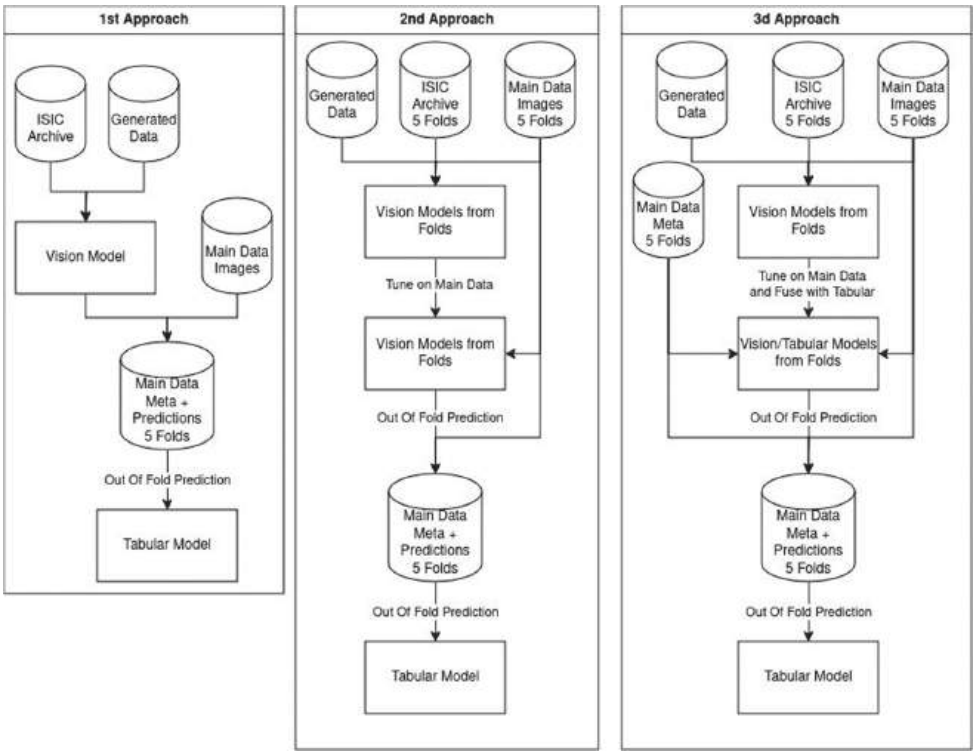


Fig. 6. Models pipelines and validation schemes

The two-stage system, which integrates Vision and Tabular models, can follow several aggregation approaches (Fig. 6):

1. Vision-Only Pretraining:
- Train the Vision model on ISIC Archive and Generated data.
- Generate predictions for the Main dataset.
- Train the Tabular model using Vision model predictions and tabular features.
2. Vision Model Pretraining and Fine-Tuning:
- Train the Vision model on ISIC Archive and Generated + Main datasets.
- Fine-tune the Vision model on the Main dataset only.
- Generate out-of-fold predictions on the Main dataset.
- Train the Tabular model using these predictions and tabular features.
3. Multi-Modal Pretraining and Fine-Tuning: Same as Approach 2, but tabular data is also incorporated during the Vision model fine-tuning.

In the final third stage of the system (Fig. 7), the Optuna algorithm is used for the final stage of coefficient optimization. However, it is crucial to recognize that the validation metrics obtained during the second stage, particularly for the second and third approaches, may be unreliable and could lead to overly optimistic outcomes. Similarly, the final stage lacks validation, which can further contribute to unfavorable results. To address these limitations and ensure a robust evaluation of the final system's performance, the Public and Private Leaderboards from the Kaggle Competition [31] are used as benchmarks. The Public test set contains approximately 140.000 tiles, while the Private test set includes around 360.000 tiles. These external benchmarks provide a more realistic and unbiased assessment of the system's capabilities.

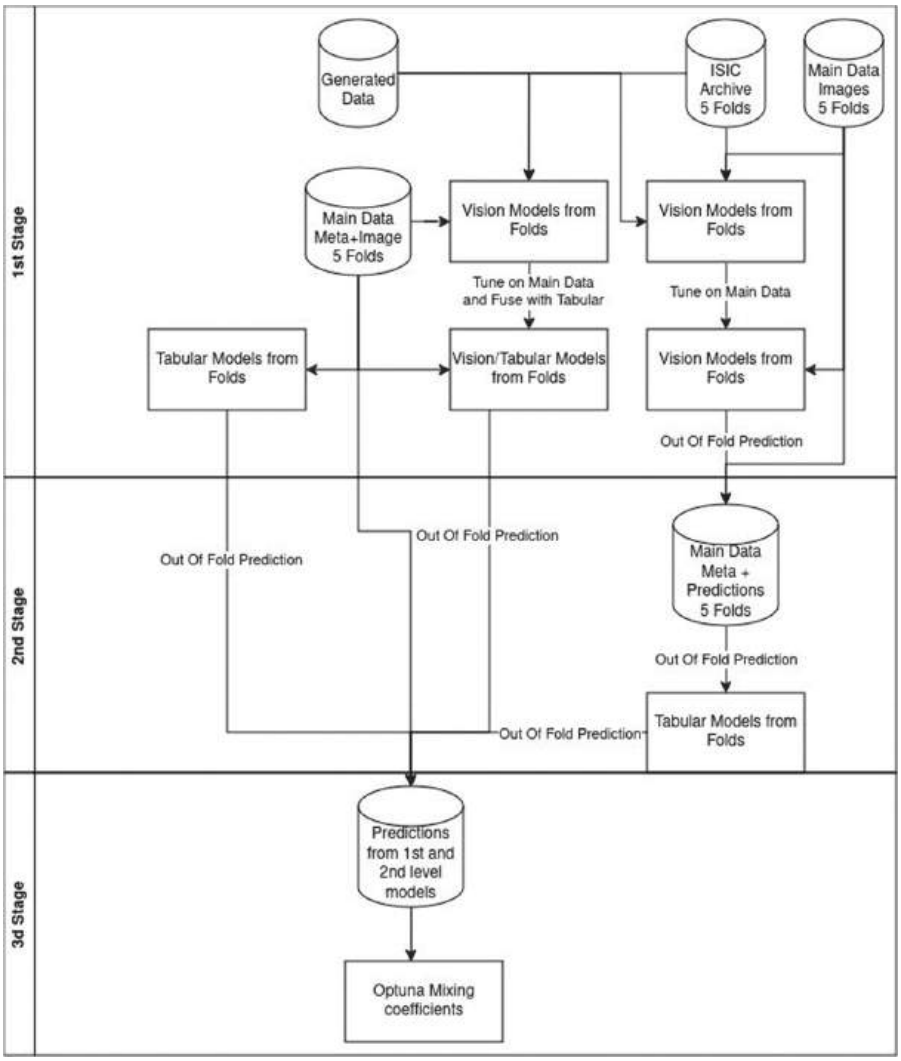


Fig. 7. Scheme of Multi-model fusion system

Feature Engineering. Basic preprocessing steps are applied to the tabular data, including handling missing values and removing redundant features. Specifically, missing numerical values are filled with the median, accompanied by adding a missing indicator feature, while missing categorical values are replaced with a new "nan" category. Columns that are static or present only in the training data are dropped to ensure consistency across datasets.

In addition to basic tabular features, several advanced features are manually engineered to capture spatial, color, and physical relationships inherent in the data. Initial proposals for these features are generated using ChatGPT [32] and refined through pruning. Key engineered features include:

- Lesion size ratio: The ratio of the minimum to the maximum diameter of the lesion.
- Hue contrast: The difference in hue between the lesion's center and periphery.
- Perimeter-to-area ratio: The ratio of the lesion's perimeter to its area.

A critical aspect involves the comparison of lesion characteristics within the same patient or body region, motivated by [33]. To address this, aggregation features are introduced:

1. Deviation within patients: StandardScaler [34] is applied within each patient-id group to capture deviations relative to other lesions of the same patient.

2. Deviation within body regions: StandardScaler is applied within combined patient-id and anatomic-site-general groups, reflecting deviations in specific body regions (e.g., arm, leg).

3. Extremes within patients: Maximum and minimum feature values per patient are calculated. Given the limited patient sample size, these features are discretized using QuantileTransformer [35] to mitigate overfitting risks.

Additionally, incorporating skin type as a feature inspired by [36] improved performance.

Finally, categorical features are one-hot encoded to prepare the data for model training.

Multi-Modal Neural Net: Image + Tabular data. Multi-Modal Vision + Tabular model is trained in 2 stages:

1. A CNN Encoder combined with a Multilayer Classifier is trained on the ISIC Archive, Generated, and Main datasets.

2. The pre-trained CNN Encoder is combined with a randomly initialized Feed-Forward Tabular Neural Net in a new Multilayer Classifier, and this combined model is fine-tuned only on the Main dataset.

All images are resized to a resolution of 128×128. Images from the ISIC Archive are also center-cropped before resizing. Continuous tabular features are normalized using the StandardScaler, while categorical features are one-hot encoded.

ConvNeXt V2 Pico [37], EdgeNeXt Base [38], and EfficientNetV2 B0 [39] CNN architectures are used. Pre-trained models from the timm repository [40] are starting points for first-stage training: convnextv2_pico.fcmae_ft_in1k, edgenext_base.in21k_ft_in1k, tf_efficientnetv2_b0.in1k. EdgeNeXt Base is used in one of the multi-modal architectures to leverage its attention mechanisms, prioritizing robustness over inference speed. EfficientNetV2 B0 is employed as a first-level model to generate predictions for the second-level pipeline, benefiting from its high inference speed. ConvNeXt V2 Pico balances inference speed and accuracy, making it suitable for prediction generation and multi-modal architectures.

Heavy augmentations are applied to enhance model robustness and mitigate overfitting in undersampled malignant cases. These include various spatial, color, blurring, distortion, and dropout augmentations, introducing variability and improving generalization (see Section Detailed Neural Net Architecture and Training Setup).

To address class imbalance during training:

- A balanced sampling strategy is employed in the first stage, ensuring equal representation of positive and negative classes.
- A “square” balancing strategy is applied in the second stage to refine class distribution further.

For generating predictions, the last and best (based on validation Partial ROC AUC) is used. Predictions are averaged across test-time augmentations (TTA), incorporating four flips to increase accuracy and robustness.

Detailed Neural Net Architecture and Training Setup. Images are first normalized to the [0, 1] range and then normalized to ImageNet statistics [41]. For image resizing, methods from the OpenCV library [42] are used, specifically INTER_AREA and INTER_LANCZOS4. The final choice is INTER LANCZOS4, though the overall difference between methods is marginal.

Both first and second-stage models are trained with the following series of augmentations:

- Transpose with a probability of 0.5.
- Vertical Flip with a probability of 0.5.
- Horizontal Flip with a probability of 0.5.
- Random Brightness and Contrast adjustment, with changes in brightness and contrast within the range [–0.2, 0.2] and a probability of 0.75.
- One of the following blurs: Motion, Median, Gaussian, or Gaussian Noise, with variation in the range [5, 30] and a blur kernel size limit of up to 5, applied with a probability of 0.7.
- One of the following distortions: Optical (limit up to 1.0), Grid (5 steps with a limit of up to 1.0), or Elastic Transform (alpha up to 3), applied with a probability of 0.7.
- CLAHE with a clip limit of up to 4, applied with a probability of 0.7.
- Random adjustment of hue, saturation, and value. Hue changes within [-10, 10], saturation within [–20, 20], and value shift within [–10, 10], with a probability of 0.5.
- Shifting, scaling, and rotation of the image. Shift within [–0.1, 0.1], scale within [–0.1, 0.1], and rotation within [–15, 15] degrees, applied with a probability of 0.85.
- Coarse Dropout with one hole of 48 width and height, applied with a probability of 0.7.

Models for both stages are trained with a batch size 64 and Binary cross-entropy loss. First-stage models are trained for 10 epochs, while second-stage models are trained for one epoch. The limited number of epochs is due to the large dataset size combined with a small number of positive samples. Balanced sampling is used, and overfitting tends to occur early.

For optimization, the Adam optimizer [43] is used in the first stage, while RAdam [44] is used in the second stage. The learning rate settings are as follows:

- For the first stage, the starting learning rate is set to. For the EdgeNeXt model, weight decay is set to.
- For the second stage:
- The CNN encoder (from the first stage) starts with a learning rate of 1e-4.
- The Feed-forward Tabular Neural Net and Multilayer Classifier start with a learning rate of 1e-3.

Such a choice of different starting learning rates is crucial to fit the entire model within one epoch while avoiding overfitting the training data or one of the modalities (image or tabular). Learning rates are reduced using cosine scheduling to 10 times smaller than their initial value. For second-stage training, the last checkpoint from the first stage is used. “Square” balancing weights for second-stage

training are illustrated in Equation below. For EfficientNetV2, the starting learning rate is set to and reduced to. This model is trained on all ISIC Archive and Generated data, resulting in good convergence.

$$class\ weight_k = \sqrt{\frac{\sum_{j=1}^{C} \sum_{i=1}^{N} class_{i,j}}{\sum_{i=1}^{N} class_{i,k}}}$$

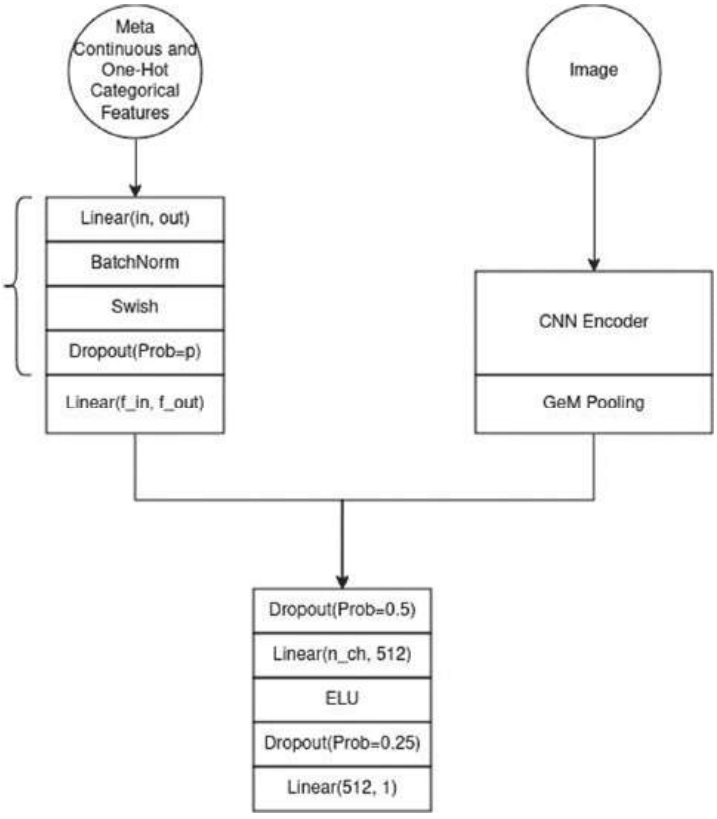


Fig. 8. Vision and Tabular general model architecture

The overall Image and Tabular model architecture is inspired by [16]. The resulting model architecture is shown in Fig. 8. Detailed values for the number of hidden channels and other hyperparameters can be found in Tables 1 and 2.

Table 1. CNN Encoders

CNN Encoder	Embedding Shape	Vision and Tabular Embedding Shape
ConvNeXt	512	576
EdgeNeXt	584	648
EfficientNetV2	192	–

Table 2. Feed Forward Net for Meta Features

Layer	In Channels	Out Channels	Dropout Probability
1	200	256	0.3
2	256	512	0.3
3	512	128	0.3
Final	128	64	–

An important feature of the multi-modal model is a substantial bottleneck in the final Feed-forward Net layer. This bottleneck is critical for preventing overfitting to the tabular branch. Using a more significant number of channels results in faster overfitting and poorer final results.

Tabular Approach. To address the class imbalance and enhance the efficiency of training and hyperparameter selection, RandomUnderSampler [45] is used as the initial step for all tabular models. The underlying model used is a boosting algorithm (LightGBM [46] or XGBoost [47]), with or without early stopping.

- For boosting without early stopping, a single model is trained on the entire training dataset, and the number of epochs is selected as one of the hyperparameters.
- In the case of boosting with early stopping, an ensemble of five models is trained. Each model is fitted on 4/5 of the training data, with early stopping performed on the remaining 1/5 (the data split follows the same principle described in the validation section) (Fig. 9).

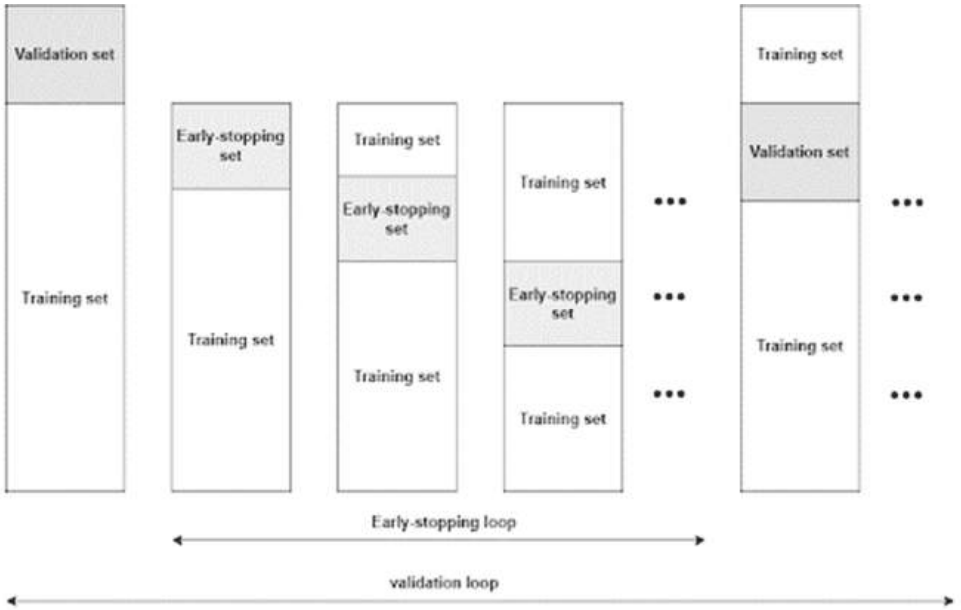


Fig. 9. Validation strategy for Tabular Models

Hyperparameter tuning, including the under-sampling ratio and the number of epochs for models without early stopping, is conducted using the Optuna optimization framework in the following steps:

1. Initial Optimization: 300 Optuna trials are performed without predefined starting parameters.

2. Parameter Refinement: The best parameters from the top five trials are combined to reduce overfitting and ensure a more robust solution. For numerical parameters, medians are calculated across the top-performing trials. For categorical parameters, the most frequent value or the value from the model with the highest Partial ROC AUC score is selected.

The primary fusion scheme is depicted in Fig. 7.

1. In the first stage, tabular and multi-modal tabular/image neural models are trained. An image-only neural model is also trained for use in the second stage.

2. In the second stage, a tabular model is trained using tabular features and the outputs of the image-only neural model.

3. In the third stage, the outputs of all models from the first stage (except the image-only neural model) and the outputs of the second stage models are ensembled using Optuna coefficient optimization. The optimization is performed on Partial ROC AUC directly on validation folds.

Predictions are generated from each fold model in the final system, increasing inference time but significantly improving robustness. This tradeoff is crucial for real-world applications such as skin cancer detection.

Due to the different nature of models and variations in the number of positive samples, the final probability distributions for each model can vary (Fig. 10). The distributions are standardized using the rank method to address this, where probabilities are converted to ranks before ensembling.

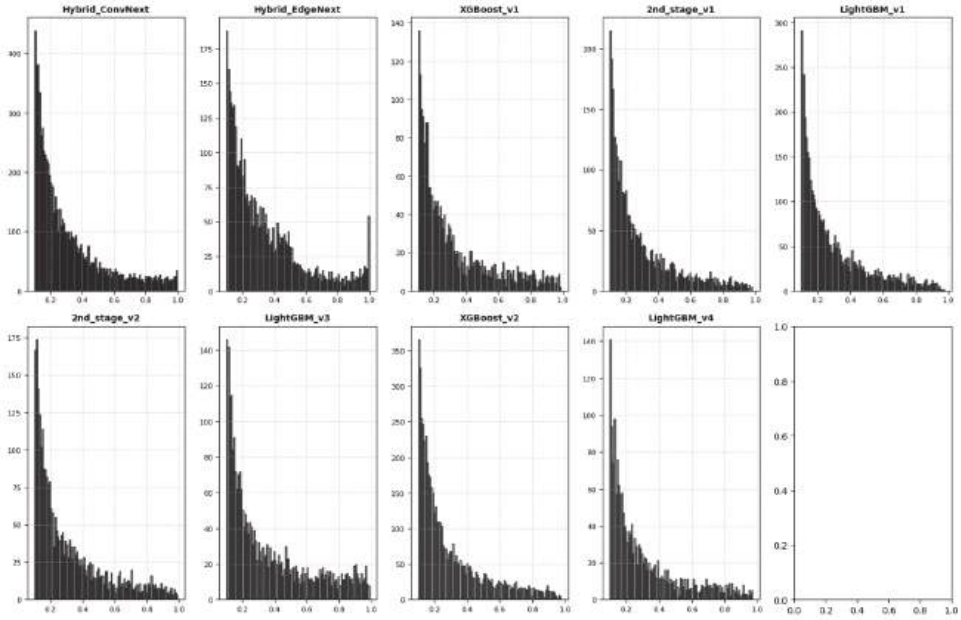


Fig. 10. Probability distribution of different models, trimmed by 0.1. Trimming is needed because most of the probabilities are lower than 0.1, while the most interesting part is above this point

For the final Optuna optimization stage, overfitting to the training (validation) set is a notable risk. The top 10 results from 5 optimization runs are collected and averaged to mitigate this. The results of the coefficient search are shown in Fig. 11. Final ensemble weights are adjusted manually to refine the system further.

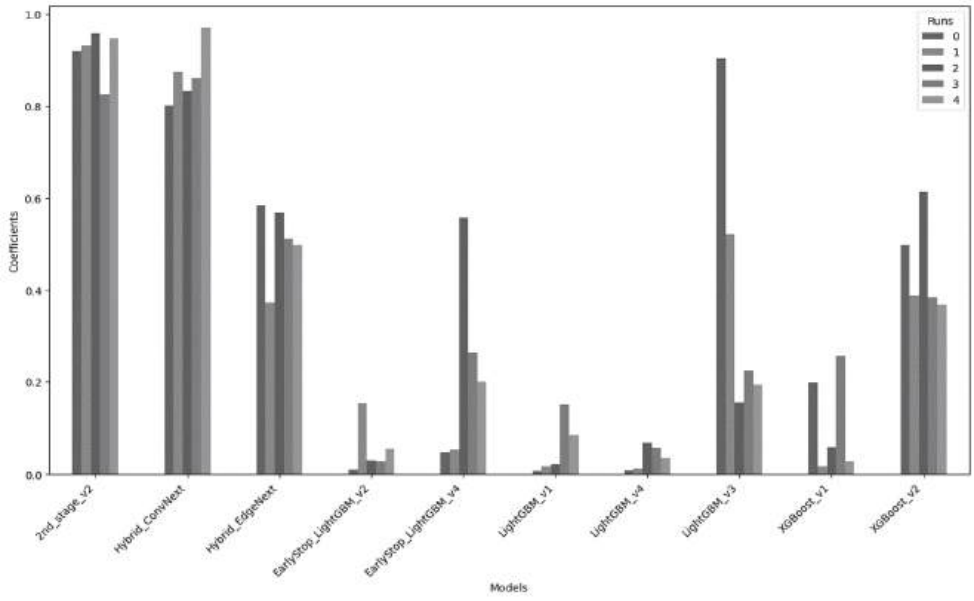


Fig. 11. Coefficients of 3rd Stage (Three-stage v2) obtained from Optuna across several runs

RESULTS

In the results, Tables 3, 4, 5 and the coefficient Table 12 different versions refer to the following:

- Different versions of Two-stage models. These correspond to the different approaches illustrated in Fig. 6.
- Different versions of XGBoost and LightGBM models. These reflect minor adjustments in the Optuna configurations or feature setups. For instance, version ≥ 2 incorporates the skin tone feature.

We evaluate the proposed methods using validation, private, and public datasets. The primary metrics include Partial ROC AUC, ROC AUC, and Top 15 Retrieval Sensitivity. For the validation dataset, we report both Out-of-Fold (OOF) and Mean metrics, with their relevance discussed in Section Metrics. Due to limitations, we only report the Top 15 Retrieval Sensitivity for the validation dataset.

Metrics of One-stage and Two-stage Models

The proposed two-stage v2 model demonstrates superior performance across all datasets in Partial and Full ROC AUC metrics, as detailed in Tables 3 and 4. The Multi-Modal ConvNext model achieves the highest Top 15 Retrieval Sensitivity (Table 5). While validation results for two-stage models may appear over-optimistic (see Section Validation), their consistent outperformance on public and private datasets supports such conclusions.

Table 3. Partial ROC AUC of One-Stage and Two-Stage Models

Model	OOF	Mean	Private	Public
Two-stage v2	0.17666	0.17862	0.16941	0.18608
Multi-Modal ConvNext	0.17497	0.17698	0.16090	0.17714
XGBoost v2	0.17348	0.17460	–	–
XGBoost v1	0.17252	0.17351	–	–
EarlyStop LightGBM v4	0.17225	0.17305	–	–
LightGBM v3	0.17173	0.17266	0.16107	0.18400
LightGBM v4	0.17105	0.17210	–	–
EarlyStop LightGBM v1	0.17029	0.17183	–	–
EarlyStop LightGBM v2	0.17024	0.16187	0.16187	0.18336
Two-stage v1	0.17010	0.17116	0.16173	0.18283
LightGBM v1	0.17005	0.17165	–	–
Multi-Modal EdgeNext	0.15892	0.17410	0.16082	0.17481

Table 4. ROC AUC of One-Stage and Two-Stage Models

Model	OOF	Mean
Two-stage v2	**0.97234**	**0.97395**
Multi-Modal ConvNext	0.97082	0.97244
XGBoost v2	0.96826	0.96916
XGBoost v1	0.96741	0.96817
EarlyStop LightGBM v4	0.96719	0.96765
LightGBM v3	0.96599	0.96682
LightGBM v4	0.96586	0.96666
Two-stage v1	0.96518	0.96600
EarlyStop LightGBM v2	0.96502	0.96605
EarlyStop LightGBM v1	0.96491	0.96629
LightGBM v1	0.96485	0.96632
Multi-Modal EdgeNext	0.95110	0.96701

Table 5. Top 15 Retrieval Sensitivity of One-Stage and Two-Stage Models

Model	OOF	Mean
Multi-Modal ConvNext	**0.76081**	**0.75995**
Two-stage v2	0.74809	0.75375
XGBoost v2	0.73791	0.73919
LightGBM v1	0.73537	0.73769
EarlyStop LightGBM v2	0.72774	0.73095
LightGBM v4	0.72774	0.72939
XGBoost v1	0.72519	0.72621
Two-stage v1	0.72265	0.72304
EarlyStop LightGBM v4	0.72265	0.72658
EarlyStop LightGBM v1	0.72010	0.72296
LightGBM v3	0.71501	0.71614
Multi-Modal EdgeNext	0.71247	0.71542

Metrics of Three-stage System

Three-stage systems outperform both standalone and two-stage models on the private dataset (Table 6). For this comparison, validation results may not be fully reliable; however, we can evaluate performance based on results from the Private and Public datasets. Finally, compared to the top solutions from the competition (Table 7), our proposed solutions underperform approximately by 2 %. Taking into account a small number of malignant cases, we may consider such a difference to be a marginal one.

Compared to prior works on melanoma detection (Table 8), our solutions outperform all previous approaches. While this comparison is not entirely equitable due to differences in training and validation datasets, most prior studies rely on higher-quality dermoscopic images. Given the lower quality of photo images in our dataset, poorer results might have been expected. However, the findings strongly indicate that our proposed solution performs comparably, if not better, on lower-quality photo images, demonstrating its robustness and applicability in real-world scenarios.

Table 6. Metrics of Three-stage Systems

System	Part ROC AUC	Private	Public	Sensitivity
Two-stage v2	0.17666	0.16941	**0.18608**	0.74809
Two-stage v1	0.17010	0.16173	0.18283	0.72265
Three-stage v2	**0.18068**	**0.17042**	0.18528	**0.78371**
Three-stage v1	0.18014	0.16982	0.18449	0.77608
Three-stage v1 mc	0.17939	0.17039	0.18527	0.78117

Table 7. Comparison with Best Competition Results

Solution	Private	Public
1st Private Place	**0.17264**	0.18611
1st Public Place	0.17051	**0.188**
Three-stage v1 mc	0.17039	0.18527
Three-stage v2	0.17042	0.18528

Table 8. Comparison with Other SOTA Researches

Experiment	Dataset	ROC AUC
2020 Best Solution [16]	2020 ISIC Competition [48]	0.9490
Saranya N et al. [14]	PH2 [49]	0.87
Saranya N et al. [14]	Derm7pt [15]	0.76
Jojoa Acosta et al. [50]	ISIC 2017 Challenge [21]	0.91
Ours (Multi-Modal ConvNext)	ISIC 2024 Kaggle Challenge	0.97244
Ours (XGBoost v2)	ISIC 2024 Kaggle Challenge	0.96916
Ours (Two-stage v2)	ISIC 2024 Kaggle Challenge	**0.97395**

Image and Multi-Modal Models Ablation Study

As a Baseline model, EfficientNet B1 [51] is used. It uses image resolution 128, severe data augmentations, and a balanced data sampler (described in Section Detailed Neural Net Architecture and Training Setup). As shown in Table 9, the model benefits from adding ISIC Archive data, even considering the modality shift. Regarding increasing image resolution, we observe reasonable score improvements on the Validation set but controversial results on Public and Private sets. Higher resolution noticeably slows down model training and inference, leading us to drop this feature. Discussing different loss functions, such as Focal [52] and ASL [53], we see improvements across all scores, which is expected given the high label imbalance. However, Balanced MixUp [54] does not prove effective for this task. Regarding backbone architecture search across EfficientNet B1, EfficientNet V2 B0, EdgeNeXt Base, and ConvNext V2 Pico, Table 9 shows that the EdgeNeXt family performs the worst, while EfficientNet V2 achieves the best results. The ConvNext family performs in the middle, marginally underperforming EfficientNet V2.

Table 9. Ablation Study of Image-Only Model

Setup	Mean Partial ROC AUC	Public	Private
Baseline	0.15252	0.14769	0.13588
Add ISIC Archive data	0.15478	0.15500	0.14390
Add ISIC Archive data + Resolution 256	0.15761	0.15208	0.14633
Add ISIC Archive data + Focal loss	0.15889	0.15361	**0.14645**
EfficientNet V2 B2 + Add ISIC Archive data	**0.162993**	0.15020	0.13812
ConvNext V2 Pico + Add ISIC Archive and Generated data	0.15723	0.15490	0.14388
ConvNext V2 Pico + Add ISIC Archive and Generated data + Balanced MixUp	0.154894	0.14902	0.14315
ConvNext V2 Pico + Add ISIC Archive and Generated data + ASL loss	0.159298	0.15581	0.14474
EdgeNeXt Base + Add ISIC Archive and Generated data	0.160584	0.14729	0.13514
EfficientNet V2 B2 + 2 stage with tune on main data	0.159025	**0.15917**	0.14224

All multi-modal setups are first trained in image-only mode on Main, ISIC Archive, and Generated data and then tuned on Main data with tabular features (Sections Multi-Modal Neural Net: Image + Tabular data and Detailed Neural Net Architecture and Training Setup). As shown in Table 10, ConvNext outperforms EfficientNet v2. ASL loss shows the best performance on Validation and Private datasets. Finally, all multi-modal models outperform images only by a significant margin. This is evident because their input information is enhanced with tabular features, which are much less noisy than images.

Table 10. Ablation Study of Multi-Modal Image/Tabular Models

Setup	Mean Partial ROC AUC	Public	Private
ConvNext V2 Pico	0.17698	**0.17740**	0.16409
EfficientNet V2 B0	0.16836	0.16698	0.15547
EfficientNet V2 B2	0.16600	0.16322	0.15908
EdgeNeXt Base	0.17410	0.17481	0.16082
ConvNext V2 Pico + Balanced MixUp	0.17582	0.16851	0.15337
ConvNext V2 Pico + ASL loss	**0.17740**	0.17403	**0.16458**

Tabular Models Ablation Study

As a baseline model, LightGBM is utilized, incorporating 85 initial features. As shown in Table 11, the model is further enhanced by adding 42 features based on lesions' spatial, color, and physical properties. Performance significantly improves by including 193 features that aggregate and compare lesion characteristics within the same patient or body region.

Table 11. Ablation Study of Tabular Model

Setup	Mean Partial ROC AUC.	Public	Private
LightGBM with basic features	0.1586	0.1693	0.1494
LightGBM with additional features	0.1604	0.1707	0.1518
LightGBM with aggregated features	**0.1728**	**0.1837**	**0.1643**

DISCUSSION

Our proposed method is based on the fusion of image and metadata information in different ways, resulting in a three-stage system of different models.

Three-stage systems effectively leverage the strengths of multi-modal data integration, allowing them to correct errors from earlier stages. All three-stage systems outperform their two-stage counterparts, even when the first version of the two-stage system (Two-stage v1) is included as part of the three-stage system. This improvement likely arises from the ability of other models in the system to compensate for and correct errors introduced by Two-stage v1.

An important observation is that the first version of the first-level model is not trained on the Main data. As a result, its predictions for the Main data are affected by domain shift, potentially introducing errors and even confusing the second-stage model. We hypothesize that increasing the number of positive samples available to the second-stage model could mitigate this issue, enabling it to correct better and utilize the first-level model's predictions.

Table 12 provides coefficients for individual models and systems used to construct the three-stage system. The automated coefficient search using Optuna assigns higher coefficients to models demonstrating better performance on the validation dataset. However, there are exceptions; for instance, the Multi-Modal Vision Transformer (Multi-Modal EdgeNext) receives a notable coefficient despite not being the top performer. This is likely due to its contribution to system diversity, as it relies on an attention mechanism for decision-making.

Table 12. Coefficients for 3-Stage System

Model	3-Stage v1	3-Stage v1 mc	3-Stage v2
Multi-Modal ConvNext	**0.30939**	0.2	0.23926
Two-stage v2	–	–	**0.25295**
XGBoost v2	0.18841	0.15	0.12442
LightGBM v1	0.01507	–	0.01548
EarlyStop LightGBM v2	0.01673	–	0.01518
LightGBM v4	0.02799	–	0.00984
XGBoost v1	0.03195	0.05	0.03067
Two-stage v1	0.03730	**0.25**	–
EarlyStop LightGBM v4	0.10580	–	0.06188
LightGBM v3	0.09203	0.15	0.11032
Multi-Modal EdgeNext	0.17531	0.15	0.14000

Another noteworthy observation is the comparable performance of three-stage v1 and three-stage v1 mc. This occurs because the two-stage v1 model is re-weighted in the second three-stage system. This re-weighting validates the hypothesis that other models within the system effectively correct the errors introduced by two-stage v1, resulting in comparable or even slightly better performance for three-stage v1 mc. This highlights the value of leveraging diverse models in multi-stage systems to enhance overall robustness and accuracy.

When discussing tabular models and the ablation study, the proposed two-stage feature engineering process shows a clear performance improvement. Another notable observation is that XGBoost outperforms LightGBM, supporting the hypothesis that while LightGBM is more suitable for rapid prototyping, other boosting approaches, such as XGBoost, should be utilized to achieve the best performance.

Regarding the comparison of Tabular and Multi-Modal Vision approaches, interesting trends emerge:

- Multi-Modal ConvNext outperforms Tabular approaches on the validation dataset.
- Tabular approaches outperform Multi-Modal ConvNext on the Public dataset.
- Both approaches exhibit comparable performance on the Private dataset, with a slight preference for Tabular approaches.

These differences may stem from two key factors:

- The small number of malignant tiles in the evaluation datasets introduces noise in the evaluation procedure.
- Variations in data quality between Tabular and Image data originate from different clinics and institutions. For instance, validation folds may contain higher-quality images than the Public dataset. Additionally, the automated feature extraction algorithm likely performs differently depending on the quality of the input images.

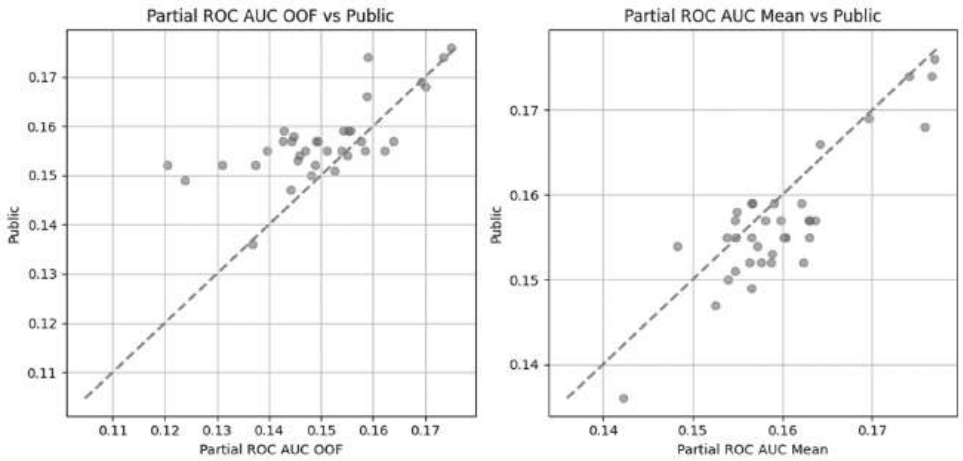


Fig. 12. Plot of scores on the validation set and Public set across 39 different Image-only and Multi-Modal experiments

The multi-modal design demonstrates robustness, but the scarcity of datasets combining image and metadata remains a significant limitation. Additionally, variations in data quality across datasets introduce evaluation noise, as reflected in discrepancies between validation and public metrics. In Figure 12, we explore the correlation between scores obtained from two evaluation datasets, using OOF and Mean score for the validation dataset. We can conclude that for both computation approaches (Mean and OOF), the score difference is reasonable for Public and validation sets. However, the Mean score correlates better.

For future work, we identify the following key directions:

– Conduct additional benchmarks on new skin cancer datasets that include metadata to evaluate system generalizability.

– Develop hybrid models capable of aggregating information across multiple nearby lesions to improve classification accuracy.

– Address the domain shift problem between dermoscopic and photo images to bridge the gap between clinical and real-world applications.

CONCLUSIONS

This paper proposes a three-stage system that leverages multi-modal data, including images and metadata, to classify skin cancer. Unlike previous works, our approach incorporates metadata directly related to the characteristics of individual lesions. We achieve enhanced system performance by employing multiple datasets (both with and without metadata), implementing a multi-step feature engineering pipeline, and using advanced techniques for optimizing performance on highly imbalanced datasets. The experiments are conducted on a novel skin cancer classification dataset composed of photo images, demonstrating the potential applicability of our approach in real-world scenarios, benefiting many patients.

ACKNOWLEDGEMENTS

First and foremost, we express our deepest gratitude to the Armed Forces of Ukraine, the Security Service of Ukraine, the Defence Intelligence of Ukraine, and the State Emergency Service of Ukraine for ensuring the safety and security that made it possible to complete this work. We also sincerely thank the Kaggle team, Canfield Scientific, The Shore Family Foundation, and all contributing institutions for providing the essential data and materials that enabled us to build models, test hypotheses, and complete this research. The authors acknowledge the use of OpenAI's ChatGPT for text refinement during the preparation of this manuscript. This tool enhanced the text's clarity and flow while ensuring the technical content's accuracy remained intact.

REFERENCES

1. P. Gruber, P.M. Zito, *Skin cancer*. Treasure Island (FL): StatPearls Publishing, 2024.
2. Andrew J. Wagner, Nancy Berliner, Edward J. Benz Jr., "Anatomy and physiology of the gene," in *Hematology*, pp. 3–16. Elsevier, 2018.
3. M. Mateen, S. Hayat, F. Arshad, Y.-H. Gu, M.A. Al-antari, "Hybrid Deep Learning Framework for Melanoma Diagnosis Using Dermoscopic Medical Images," *Diagnostics*, 14(19), 2242, 2024. doi: https://doi.org/10.3390/diagnostics14192242

4. V. Rotemberg et al., “A patient-centric dataset of images and metadata for identifying melanomas using clinical context,” *Sci. Data*, 8(1):34, 2021. doi: 10.1038/ s41597-021-00815-z
5. A.A. Adegun, S. Viriri, “Deep learning techniques for skin lesion analysis and melanoma cancer detection: a survey of state-of-the-art,” *Artificial Intelligence Review*, vol. 54, pp. 811–841, 2020. doi: 10.1007/s10462-020-09865-y
6. M. Naqvi, S.Q. Gilani, T. Syed, O. Marques, H.C. Kim, “Skin Cancer Detection Using Deep Learning—A Review,” *Diagnostics*, 13(11), 1911, 2023. doi: https://doi.org/ 10.3390/diagnostics13111911
7. W. Gouda, N.U. Sama, G. Al-Waakid, M. Humayun, N.Z. Jhanjhi, “Detection of Skin Cancer Based on Skin Lesion Images Using Deep Learning,” *Healthcare*, 10(7), 1183, 2022. doi: https://doi.org/10.3390/healthcare10071183
8. J.R.H. Lee, M. Pavlova, M. Famouri, A. Wong, “Cancer-Net SCa: tailored deep neural network designs for detection of skin cancer from dermoscopy images,” *BMC Medical Imaging*, vol. 22, article no. 143, 2022. doi: https://doi.org/ 10.1186/s12880-022-00871-w
9. B. Cassidy, C.Kendrick, A. Brodzicki, J. Jaworek-Korjakowska, M.H. Yap, “Analysis of the ISIC image datasets: Usage, benchmarks and recommendations,” *Medical Image Analysis*, vol. 75, 102305, 2022. doi: https://doi.org/10.1016/ j.media.2021.102305
10. D. Wen, A. Soltan, E. Trucco, R.N. Matin, “From data to diagnosis: skin cancer image datasets for artificial intelligence,” *Clinical and Experimental Dermatology*, vol. 49, issue 7, pp. 675–685, 2024. doi: https://doi.org/10.1093/ced/llae112
11. K.M. Hosny, M.A. Kassem, M.M. Foaud, “Classification of skin lesions using transfer learning and augmentation with Alex-net,” *PLOS One*, 14(5), e0217293, 2019. doi: https://doi.org/10.1371/journal.pone.0217293
12. K. He, X. Zhang, S. Ren, J. Sun, *Deep Residual Learning for Image Recognition*, 2015. doi: https://doi.org/10.48550/arXiv.1512.03385
13. G. Huang, Z. Liu, L. van der Maaten, K.Q. Weinberger, *Densely Connected Convolutional Networks*, 2018. doi: https://doi.org/10.48550/arXiv.1608.06993
14. N. Saranya, Alfred C. Jowin, R.R. Rishikesh, Idayan I. Gilbert, “Analysis of GAN for Melanoma Skin CancerClassification with Dermatologist Recommendation,” in *Proceedings of the 2024 International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), Chennai, India, 2024*, pp. 1–8. doi: https://doi.org/10.1109/ RAEEUCCI61380.2024.10547727
15. J. Kawahara, S. Daneshvar, G. Argenziano, G. Hamarneh, “Seven-point checklist and skin lesion classification using multitask multimodal neural nets,” *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, pp. 538–546, 2019. doi: https://doi.org/10.1109/JBHI.2018.2824327
16. *boliu61, SIIM-ISIC Melanoma Classification - Discussion on Kaggle*. 2020. Accessed on: October 27, 2024. Available: https://www.kaggle.com/c/siim-isic-melanoma-classification
17. M.R. Thanka et al., “A hybrid approach for melanoma classification using ensemble machine learning techniques with deep transfer learning,” *Computer Methods and Programs in Biomedicine Update*, 3(11):100103, 2023. doi: 10.1016/ j.cmpbup.2023.100103
18. A. Ju, J. Tang, S. Chen, Y. Fu, Y. Luo, “Pyroptosis-related gene signatures can robustly diagnose skin cutaneous melanoma and predict the prognosis,” *Frontiers in Oncology*, vol. 11, 709077, 2021. doi: https://doi.org/10.3389/fonc.2021.709077
19. F.P. Loss et al., *Skin cancer diagnosis using NIR spectroscopy data of skin lesions in vivo using machine learning algorithms*, 2024. doi: https://doi.org/10.48550/ arXiv.2401.01200
20. *International Skin Imaging Collaboration. SLICE-3D 2024 Challenge Dataset, 2024. Creative Commons Attribution-Non Commercial 4.0 International License*. doi: https://doi.org/10.34970/2024-slice-3d

21. N. Kurtansky et al., "The SLICE-3D dataset: 400,000 skin lesion image crops extracted from 3D TBP for skin cancer detection," *Scientific Data*, 11, article no. 884, 2024. doi: https://doi.org/10.1038/s41597-024-03743-w
22. *MAli-Farooq.Derm-T2IM-Dataset*, 2024. Accessed on: October 20, 2024. Available: https://huggingface.co/datasets/MAli-Farooq/Derm-T2IM-Dataset
23. *Canfield Scientific, I. VECTRA® WB360 Imaging System*, 2024. Accessed on: October 21, 2024.
24. B. D'Alessandro, "Methods and Apparatus for Identifying Skin Features of Interest," US11164670B2, Nov. 2021. Filed: March 18, 2016; Issued: November 2, 2021.
25. B. Betz-Stablein et al., "Reproducible Naevus Counts Using 3D Total Body Photography and Convolutional Neural Networks," *Dermatology*, 238(1), pp. 4–11, 2022. doi: https://doi.org/10.1159/000517218
26. R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, *High-Resolution Image Synthesis with Latent Diffusion Models*, 2021. doi: https://doi.org/10.48550/arXiv.2112.10752
27. "Partial Area Under the ROC Curve," *Wikipedia*, 2023. Accessed on: October 26, 2024. Available: https://en.wikipedia.org/wiki/ Partial_Area_Under_the_ROC_Curve
28. *ISIC Research. Challenge 2024 Metrics*, 2024. Accessed on: October 26, 2024. Available: https://github.com/ISIC-Research/Challenge-2024-Metrics/tree/main
29. *Kaggle. ISIC 2024 Challenge - Secondary Prize Metrics*, 2024. Accessed on: October 26, 2024. Available: https://www.kaggle.com/competitions/isic-2024-challenge/overview/ secondary-prize-metrics
30. T. Akiba, S. Sano, T. Yanase, T. Ohta, M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623–2631, 2019. doi: https://doi.org/10.1145/3292500.3330701
31. N. Kurtansky, V. Rotemberg, M. Gillis, K. Kose, W. Reade, A. Chow, "ISIC 2024 - Skin Cancer Detection with 3D-TBP," *Kaggle*, 2024. Available: https://kaggle.com/competitions/isic-2024-challenge
32. *OpenAI. ChatGPT*. Accessed on: November 22, 2024. Available: https://chatgpt.com/
33. A. Scope et al., "The "ugly duckling" sign: agreement between observers," *Archives of dermatology*, 144(1), pp. 58–64, 2008. doi: 10.1001/archdermatol.2007.15
34. "StandardScaler," *scikit learn*, 2024. Accessed on: November 22, 2024. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
35. "Quantile Transformer," *scikit learn*, 2024. Accessed on: November 22, 2024. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html
36. S. Du, B. Hers, N. Bayasi, G. Hamarneh, R. Garbi, "FairDisCo: Fairer ai in dermatology via disentanglement contrastive learning," *in Proceedings of the Computer Vision–ECCV 2022 Workshops: Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*. Springer, 2023, pp. 185–202.
37. S. Woo et al., "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023*, pp. 16133–16142. doi: 10.1109/CVPR52729.2023.01548
38. M. Maaz et al., "EdgeNeXt: Efficiently Amalgamated CNN-Transformer Architecture for Mobile Vision Applications," in *Proceedings of the European Conference on Computer Vision*. Springer, 2022, pp. 3–20.
39. M. Tan, Q. Le, "Efficientnetv2: Smaller models and faster training," in *Proceedings of the International Conference on Machine Learning, PMLR, 2021*, pp. 10096–10106.
40. R. Wightman et al., *PyTorch Image Models*, 2019. doi: https://doi.org/10.5281/zenodo.4414861
41. J. Deng, W. Dong, R. Socher, L.J. Li, K. Li, L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255. doi: https://doi.org/10.1109/CVPR.2009.5206848
42. *Itseez. Open Source Computer Vision Library*, 2015. Available: https://github.com/itseez/opencv

43. D.P. Kingma, J. Ba, *Adam: A Method for Stochastic Optimization*, 2017. doi: https://doi.org/10.48550/arXiv.1412.6980
44. L. Liu et al., *On the Variance of the Adaptive Learning Rate and Beyond*, 2021. doi: https://doi.org/10.48550/arXiv.1908.03265
45. G. Lemaître, F. Nogueira, C.K. Aridas, "Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning," *Journal of Machine Learning Research*, 18, pp. 1–5, 2017.
46. G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, 30, 2017.
47. T. Chen, C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794. doi: https://doi.org/10.1145/2939672.2939785
48. A. Zawacki et al., "SIIM-ISIC Melanoma Classification," *Kaggle*, 2020. Available: https://kaggle.com/competitions/siim-isic-melanoma-classification
49. T. Mendonça, P.M. Ferreira, J.S. Marques, A.R.S. Marcal, J. Rozeira, "PH2 - A dermoscopic image database for research and benchmarking," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Osaka, Japan, 2013*, pp. 5437–5440. doi: https://doi.org/10.1109/EMBC.2013.6610779
50. M.F.J. Acosta, L.Y.C. Tovar, M.B. Garcia-Zapirain, W.S. Percybrooks, "Melanoma diagnosis using deep learning techniques on dermatoscopic images," *BMC Medical Imaging*, vol. 21, article no. 6, 2021. doi: https://doi.org/10.1186/s12880-020-00534-8
51. M. Tan, Q.V. Le, *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*, 2020. doi: https://doi.org/10.48550/arXiv.1905.11946
52. T.Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, *Focal Loss for Dense Object Detection*, 2018. doi: https://doi.org/10.48550/arXiv.1708.02002
53. E. Ben-Baruch et al., *Asymmetric Loss for Multi-Label Classification*, 2021. doi: https://doi.org/10.48550/arXiv.2009.14119
54. A. Galdran, G. Carneiro, M.A. González Ballester, "Balanced-MixUp for Highly Imbalanced Medical Image Classification," in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, vol. 12905, pp. 323–333. Springer International Publishing, 2021. doi: https://doi.org/10.1007/978-3-030-87240-3_31

Received 16.01.2025

INFORMATION ON THE ARTICLE

Volodymyr S. Sydorskyi, ORCID: 0000-0001-9697-7403, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: volodymyr.sydorskyi@gmail.com

Igor E. Krashenyi, ORCID: 0000-0003-0424-147X, Ukrainian Catholic University, Ukraine, e-mail: igor.krashenyi@ucu.edu.ua

Oleksii P. Yakubenko, ORCID: 0009-0009-5752-4546, "Pleso Therapy", Ukraine, e-mail: yakubenko.oleksii@gmail.com

МУЛЬТИМОДАЛЬНА СИСТЕМА ДЛЯ ВИЯВЛЕННЯ МЕЛАНОМИ ШКІРИ / В.С. Сидорський, І.Е. Крашений, О.П. Якубенко

Анотація. Виявлення меланоми є надзвичайно важливим для ранньої діагностики та ефективного лікування. Хоча глибокі нейронні мережі на основі дерматоскопічних зображень показали багатообіцяльні результати, їх використання потребує спеціалізованого обладнання, що обмежує їх застосування в ширших клінічних умовах. Представлено мультимодальну систему виявлення меланоми, яка використовує звичайні фотозображення, що робить її більш доступною й

універсальною. Ця система інтегрує зображення із табличними метаданими, такими як демографічні дані пацієнтів і характеристики утворів, для підвищення точності виявлення. Вона використовує мультимодальну нейронну мережу, яка поєднує оброблення зображень і метаданих, і підтримує двохетапну модель для випадків із метаданими або без них. Трирівнева система додатково вдосконалює прогнози за допомогою алгоритмів градієнтного бустингу, покращуючи результати. Для вирішення проблем, пов'язаних із дуже незбалансованим набором даних, реалізовано спеціальні техніки, які забезпечують надійне навчання. У дослідженні впливу компонентів оцінено сучасні архітектури комп'ютерного зору, алгоритми бустингу і функції втрат із досягненням пікового значення часткової AUC ROC 0.18068 (максимум 0.2) та чутливості в топ-15 пошуку 0.78371. Результати демонструють, що інтеграція фотозображень із метаданими у багаторівневу систему забезпечує суттєве покращення продуктивності. Ця система просуває виявлення меланоми, пропонуючи масштабоване рішення, яке не залежить від спеціалізованого обладнання і підходить для різноманітних умов надання медичної допомоги, об'єднуючи спеціалізовану та загальноклінічну практики.

Ключові слова: класифікація медичних зображень, комп'ютерний зір, градієнтний бустинг, глибокі нейронні мережі, клінічні системи підтримання прийняття рішень.

UDC 536.531
DOI: 10.20535/SRIT.2308-8893.2026.1.04

POLYNOMIAL-BASED METHOD FOR LINEARIZING THE TEMPERATURE RESPONSE OF NTC THERMISTORS

S.N. MATVIENKO, G.S. TYMCHYK

Abstract. The objective of this research is to develop a circuit-based method for linearizing the temperature characteristics of thermistors using a polynomial digital technique for NTC-thermistor temperature characteristics, together with the characteristics of the designed measurement channel, by means of an original MATLAB Simulink model. A model of a temperature-measurement device employing a thermistor-based sensor is proposed. The polynomial digital method for linearizing the temperature characteristics of NTC thermistors in the developed device has been simulated. To improve measurement accuracy, the components of the measurement channel were selected such that they enable simulation with minimal error. Methods for introducing correction-coefficient values into the model have been determined to compensate for error-inducing factors, including the thermistor's self-heating effect. The proposed data-processing algorithm offers advantages for implementation in a low-cost, low-power microcontroller.

Keywords: temperature measurement, NTC thermistor, MATLAB Simulink, linearization.

INTRODUCTION

Temperature measurement devices play a critical role across various industrial and consumer domains, particularly within Internet of Things (IoT) networks that leverage Smart technologies, including smart cities, smart homes/digital houses, smart grids, and smart sensors. A fundamental requirement for temperature sensors in such applications is high measurement accuracy over a wide temperature range, which is primarily determined by the type and characteristics of the sensor.

Negative Temperature Coefficient (NTC) thermistors are widely employed as temperature sensors due to their broad operational temperature range, capability for remote monitoring, resistance to strong magnetic fields, and compact physical dimensions. However, one of the primary limitations of thermistors, shared by many sensor types — is the nonlinear nature of their resistance–temperature characteristic $R(T)$, which adversely affects measurement accuracy.

Additionally, the transfer characteristics of the thermistor interface circuitry and the voltage signal amplifier are also nonlinear. Since nonlinearities introduce measurement errors, improving the linearity of the transfer function of the entire measurement channel is essential for achieving high accuracy. The task of linearizing the temperature dependence of a measurement system can be addressed

through analog, digital (software-based), or mixed hardware-software digital techniques.

Analog methods can reduce measurement error to some extent but typically only within a narrow temperature range. In contrast, digital linearization techniques not only improve measurement accuracy but also significantly extend the effective temperature measurement range. These techniques are implemented via software or through the use of analog-to-digital interface circuits that generate a linear signal response.

Among analog approaches, passive compensation networks are the most common; they enable the creation of quasi-linear segments on the $R(T)$ curve within specific temperature intervals. Analog methods are generally more cost-effective compared to digital techniques, which require microcontrollers, field-programmable gate arrays (FPGAs), digital signal processors (DSPs), or personal computers. Nevertheless, digital methods provide a substantial improvement in accuracy over a wider temperature range.

With advancements in digital signal processing and the availability of increasingly capable digital integrated circuits, the implementation of such methods has become significantly more practical and accessible.

RELEVANCE OF THE WORK

Increasing the accuracy of temperature measurement is achieved by optimizing the design of the measuring probe and selecting the appropriate characteristics of the thermistor. Accuracy can also be increased by setting the optimal parameters for the measuring channel and implementing effective digital data processing algorithms. During the design phase of a temperature measurement device, it is essential to conduct mathematical modeling to determine the optimal linearization method, select component parameters, and define the appropriate data processing algorithm.

The development of mathematical models enables the identification of suitable characteristics for the components of the measurement channel and the data processing algorithms, tailored to the specific requirements of the application. In this research, a custom-designed model implemented in MATLAB Simulink is used to investigate a digital method for linearizing the device characteristics by applying polynomial functions to the measured data. The model facilitates the estimation of measurement errors and the development of calibration recommendations aimed at improving overall measurement accuracy.

ANALYSIS OF RECENT RESEARCH AND PUBLICATIONS

Linearized sensor characteristics significantly simplify the design and calibration processes and improve measurement accuracy. To compensate for the inherent nonlinearity of thermistor characteristics and the measurement channel, both analog and digital linearization methods have been widely reported in the literature. Researches such as [1–3] provide an in-depth analysis of these techniques and offer a comprehensive overview of various approaches used for sensor characteristic linearization.

It is noted that digital methods, particularly when combined with software-based processing, yield superior results in terms of flexibility and performance. Software

algorithms implemented in digital systems have proven to be more effective, practical, and adaptable than traditional analog methods. Common software-based linearization techniques include spline fitting, polynomial curve approximation, and advanced intelligent methods such as artificial neural networks (ANNs) [4].

Among these, the use of polynomial functions remains the most widespread technique for correcting measured data [1]. A single high-order polynomial can be used to model the full sensor response. However, to reduce computational complexity, the entire temperature range is often divided into smaller sub-ranges, each approximated with a lower-order polynomial. This segmentation improves processing efficiency without sacrificing accuracy.

Additionally, alternative enhanced techniques based on lookup tables are also employed, including piecewise linear interpolation (PWLI), piecewise linear equations (PWLE), and programmable gain amplifiers (PGA) [1]. In PWLE, the processor selects the appropriate linear equation based on the input value and retrieves stored coefficients from memory. In some implementations, the processor dynamically determines the applicable linear equation for each measurement based on pre-calibrated data points.

Depending on the complexity of the data processing algorithm, various hardware platforms can be used, such as microcontrollers, FPGAs, DSPs, or PCs. More complex algorithms typically require higher-cost solutions and result in increased power consumption. A comparison of selected algorithms can be found in [5]. Therefore, it is critical to choose the most suitable linearization approach and its hardware implementation based on the specific performance requirements of the temperature measurement device.

Mathematical modeling of the device and its linearization method can significantly streamline the design process, enabling efficient selection of component parameters and processing algorithms to meet the required accuracy specifications. Manufacturer-provided models [6–8] are also beneficial in selecting appropriate thermistor types for specific applications and offer detailed characteristic data that can be used to address $R(T)$ nonlinearity.

Researchers and engineers have proposed a wide range of algorithms to address the nonlinearity problem of the $R(T)$ characteristic [1; 3–5; 9–11]. Given the diversity of available digital linearization techniques, selecting the optimal algorithm and its hardware implementation can be challenging and depends on the specific application requirements.

During the modeling stage, it is essential to evaluate the achievable measurement accuracy, the impact of hardware element characteristics, and the feasibility of implementing calibration procedures. In this research, a classical linearization approach is explored, using polynomial correction of measured data with segmentation of the full measurement range into smaller sub-ranges. This approach reduces the required polynomial order, thereby lowering the computational load on the processing hardware. Consequently, cost-effective microcontrollers can be used, and device calibration functionality can be feasibly integrated into the system.

Therefore, **the purpose of our work** is to develop a mathematical model of the measurement channel of a temperature sensing device based on an NTC thermistor within the MATLAB/Simulink environment. Using the developed model of digital linearization methods for NTC thermistor temperature characteristics — based on polynomial approximations — it is intended to determine the required

specifications of the components within the measurement channel. Additionally, an analysis of the sources of measurement error will be conducted to improve temperature measurement accuracy across a wide temperature range.

The aim of this work is to investigate digital methods for linearizing the temperature characteristics of NTC thermistors and the associated measurement channel components using the developed simulation models.

RESEARCH METHODOLOGY AND RESULTS

Measuring temperature using an NTC thermistor involves using the dependence of the thermistor's resistance on its temperature, that is, on the temperature of the environment surrounding the thermistor.

The dependence of the electrical resistance of an NTC thermistor on temperature has the form [12]:

$$R(T) = R_N \exp\left[\mathrm{B}\left(\frac{1}{T} - \frac{1}{T_N}\right)\right], \qquad (1)$$

where $R(T)$ — *is* the resistance of the NTC thermistor at temperature T in K; R_N — NTC thermistor resistance at nominal temperature T_N in K; T — the current temperature (in K) value at which the thermistor resistance R_N is calculated; T_N — nominal temperature (in K), i.e. the reference or standard temperature at which the nominal resistance of the thermistor is known; B — is a constant coefficient that depends on the thermistor material in K.

This dependence is nonlinear, which creates certain difficulties in creating temperature measurement devices and introduces additional error. The value of the error will depend on the linearization method used.

Fig. 1 shows the developed structural and mathematical model of a device for measuring temperature using an NTC thermistor in the MATLAB Simulink environment.

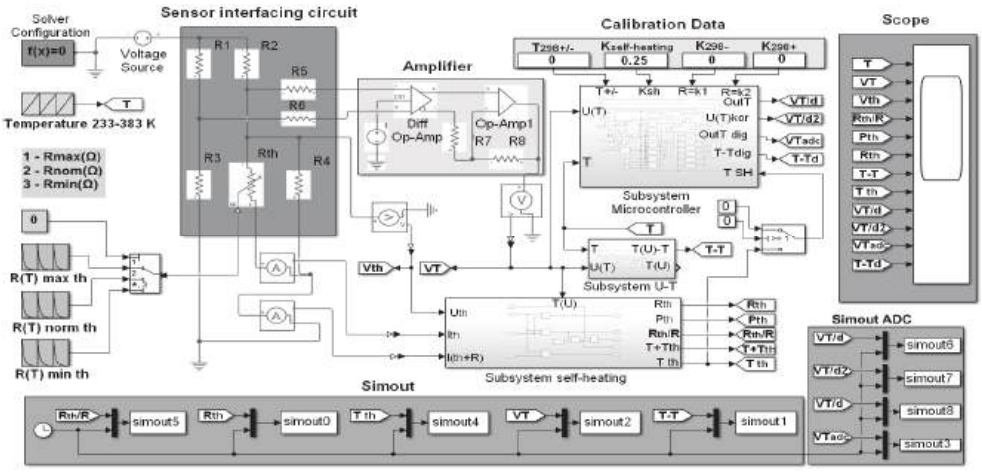


Fig. 1. Functional diagram of a temperature measurement device in the MATLAB Simulink environment

The model consists of 2 groups of blocks for modeling the electrical circuit of the measuring channel, these are the sensor interface group "*Sensor interfacing circuit*" and the "*Amplifier*" group, 3 software modules "*Subsystem*" — "*Subsystem Microcontroller*", "*Subsystem U-T*" and "*Subsystem self-heating*".

The group "*Sensor interfacing circuit*" is a diagram of connecting the thermistor model — R_{th} to the differential amplifier model "*Fully Differential Op-Amp*" of the group "*Amplifier*" using a correspondingly configured diagram of connecting the resistor models "*Resistor*" *R1,.. R6*.

R_{th} can be connected to the Wheatstone bridge arm *R1*, *R2*, *R3*, R_{th} to the "*Fully Differential Op-Amp*". The Wheatstone bridge arms *R1-R3* and *R2-* R_{th} are connected to the "*Voltage Source*" model.

In the MATLAB Simulink environment there is a model "*Thermistor*" in the library that simulates the operation of a thermistor according to the set parameters, but in this model the resistance of the thermistor is calculated by formula (1) depending on the temperature. In a real thermistor, the coefficient *B* is not a constant value, but varies depending on the temperature. Fig. 2 shows the difference between the $R(T)$ characteristic of the thermistor of the "*Thermistor*" MATLAB Simulink model and the characteristic built from tabular data for a thermistor type RH18 6Y103 Mitsubishi Materials.

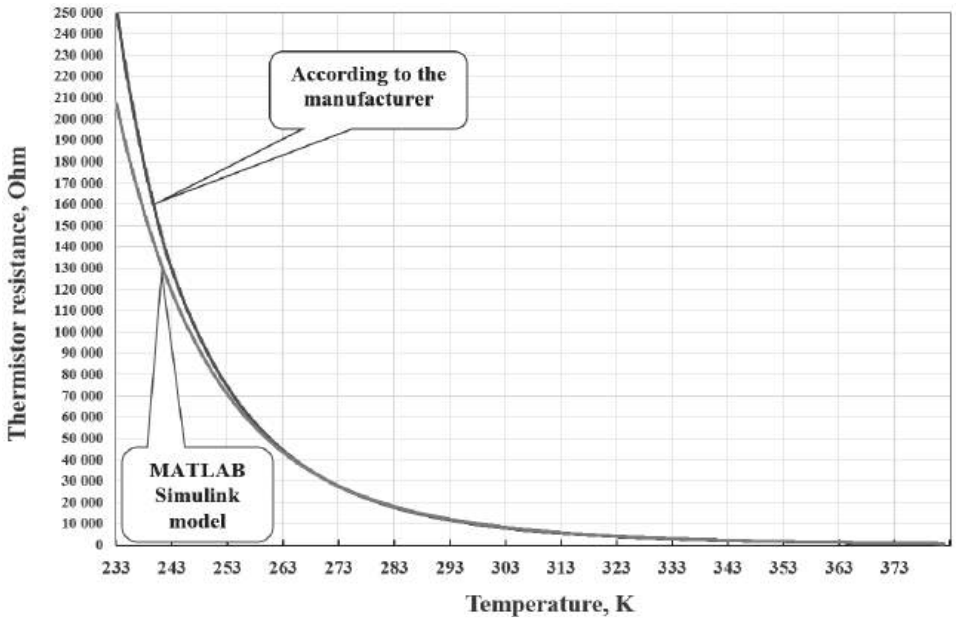


Fig. 2. Graph of the difference in $R(T)$ characteristics for the thermistor model "Thermistor" and tabulated data for the thermistor type RH18 6Y103 Mitsubishi Materials in the range from 233 *K* to 383 *K*

If the thermistor is used within a narrow temperature range and moderate measurement accuracy is acceptable, the built-in "*Thermistor*" block in MATLAB Simulink may be employed for modeling purposes. However, when the temperature range is wide and high measurement accuracy is required, the simulation should rely on tabulated data provided by the manufacturer for the specific thermistor type.

In the presented model, the thermistor is implemented using the "*Variable Resistor*", whose resistance is defined based on the manufacturer's tabular data via three "*Repeating Sequence*" blocks. These sequences correspond to the nominal, minimum, and maximum resistance values of the thermistor. The resistance value used in the simulation "R_{max}", "R_{nom}" and "R_{min}" is selected using switch S1.

To convert the temperature value measured by the thermistor into a corresponding voltage output, a widely adopted namely "*Sensor interfacing configuration*", the Wheatstone bridge (Fig. 1) — is utilized. To reduce the nonlinearity of the thermistor's resistance-temperature characteristic $R(T)$, a circuit-level linearization technique is applied by connecting a resistor *R4* in parallel with the thermistor (Fig. 1). This approach minimizes the deviation between the linearized and actual nonlinear characteristics of the "*Sensor interfacing circuit*" output, thereby enhancing the accuracy of polynomial correction methods.

The resistance of the parallel resistor used to linearize the $R(T)$ characteristic is determined according to the following formula [13]:

$$R_4 = R_{thN} \times \frac{B - 2T_N}{B + 2T_N},$$

where R_4 — resistance of a parallel-connected resistor, Ohm; R_{thN} — NTC thermistor resistance at nominal temperature T_N in *K*; *T*, T_N — temperature in *K*; *B* — is a constant coefficient that depends on the thermistor material in *K*.

The total resistance of a thermistor with a resistor connected in parallel is determined by the formula:

$$R_p = \frac{R_{thN} \times R_4}{R_4 + R_{thN}}.$$

For thermistor RH18 6Y103F Mitsubishi Materials with $B_{25/85}$ = 3435 *K* and resistance 10 kOhm±1% at temperature T_N = 298 *K* (25 °*C*) R_4 = 7043 Ohm a R_p =4142 Ohm.

The "*Amplifier"* group consists of the "*Fully Differential Op-Amp*" and the ideal operational amplifier model "*Op-Amp*", which together provide the required voltage gain *G* for the Wheatstone bridge imbalance signal.

The software module "*Subsystem U-T*" is used to obtain the uncorrected analog signal, whose voltage is proportional to the measured temperature. This output is necessary for evaluating the effectiveness of the digital linearization method. The block diagram of the "*Subsystem U-T*" module is shown in Fig. 3, *a*.

The instantaneous voltage at the output of the "*Amplifier*" block, which is proportional to the set temperature, is converted by the "*PS-Simulink Converter*" into a signal compatible with the Simulink environment. This signal is then scaled using the "*Divide*" block by an appropriate gain coefficient and converted into the corresponding temperature value in *K*. The computed thermistor self-heating temperature, determined by the "*Subsystem self-heating*", is then added to this value.

The "*Subsystem self-heating*" module is employed to calculate the self-heating temperature of the thermistor as a function of its operating temperature.

The internal structure of this module is depicted in Fig. 3, *b*.

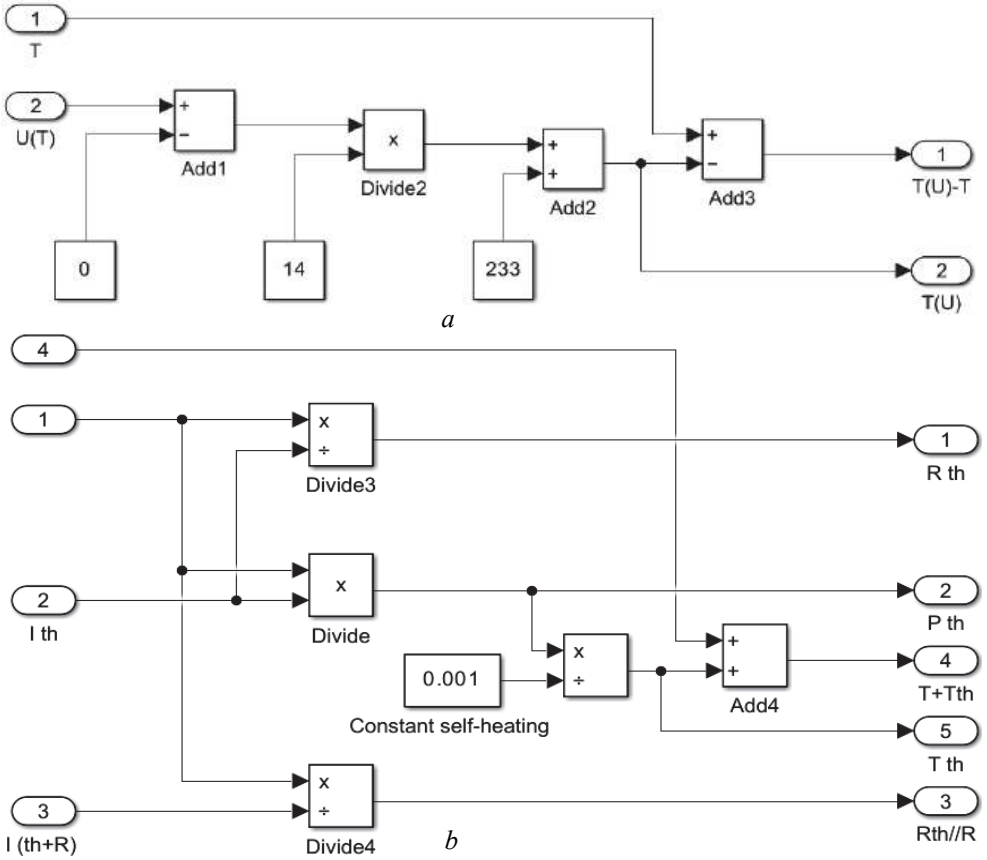


Fig. 3. Functional diagram "*Subsystem UT*" of the device module — *a*; functional diagram of the "*Subsystem self-heating*" device module — *b*

The thermistor's self-heating temperature is influenced by the magnitude of the current flowing through it, the materials and structural design of the sensor, as well as the thermal conductivity of the surrounding medium [12; 14]. This phenomenon is utilized in devices for measuring thermophysical properties of materials [15–17], as well as in systems for determining fluid flow velocity [18].

Self-heating in NTC thermistors causes an additional decrease in their resistance, which leads to distortion in the measurement result. Therefore, in temperature measurement systems, a correction must be applied to the measured value. This correction is computed using the following formula:

$$T_A = T - \frac{U^2}{\delta_{th} + R(\mathrm{T})} = T - \frac{I^2 \times R(T)}{\delta_{th}}, \qquad (2)$$

where T_N — actual value of the controlled temperature; T — Measured temperature value; U — instantaneous value of the voltage on the thermistor, I — instantaneous value of the current flowing through the thermistor; $R(T)$ — the value of the resistance of the thermistor corresponding to the temperature T; δ_{th} — heat dissipation coefficient in the measuring medium.

Using the voltage sensor model "*Voltage Sensor*" the voltage values on the "*Variable Resistor*" at the current time are recorded, and using the current sensor

models "*Current Sensor*" the current value passing through the "*Variable Resistor*" is recorded. From these data, the self-heating temperature of the thermistor for air is determined by formula (2). This data is used to correct the measurement data in the "*Subsystem Microcontroller*" module.

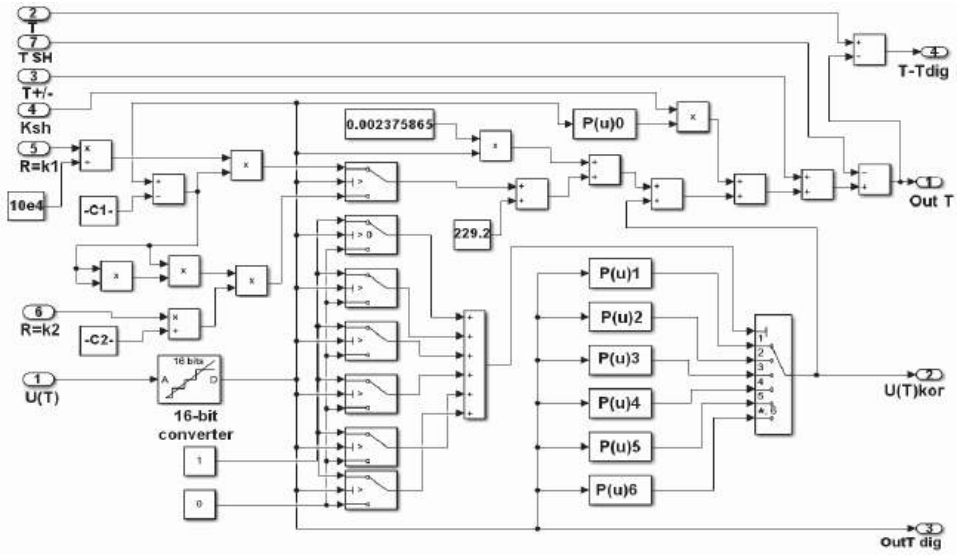


Fig. 4. Functional diagram of the "Subsystem Microcontroller" device module

For correction of measured data in the "*Subsystem Microcontroller*" module "*The model implements*" a digital linearization algorithm using polynomials with the division of the full measurement range into small subranges. Fig. 4 shows a model of the software module "*Subsystem Microcontroller*".

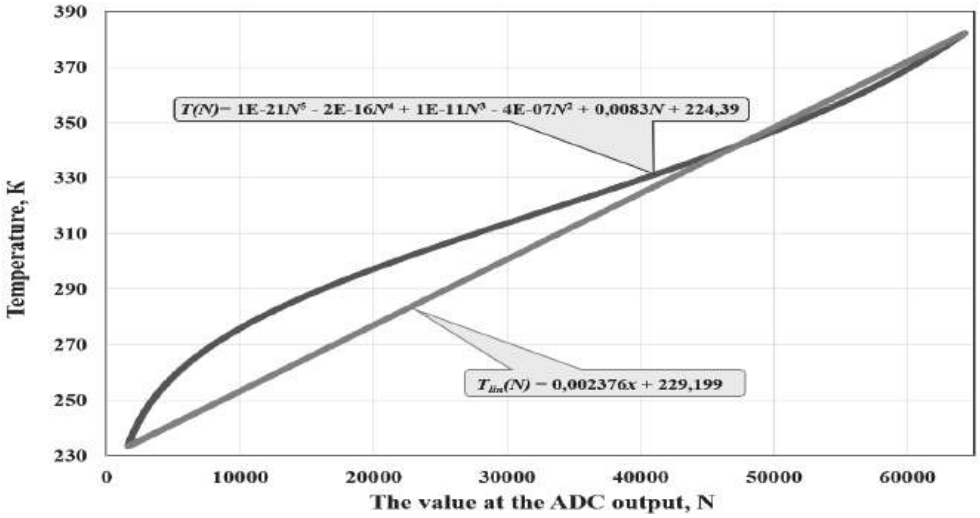


Fig. 5. Graphs of the linear characteristic $T_{lin}(N)$ compared to the nonlinear characteristic $T(N)$

N by means of a 16-bit ADC "*Idealized ADC quantizer*". Due to the nonlinearity $R(T)$ of the thermistor characteristic, the nonlinear transfer characteristic of the Wheatstone bridge circuit, the differential amplifier and the ADC, the charac-

teristic of the source code at the ADC output is $N(T)$ will also be nonlinear. The task of the "Subsystem Microcontroller" module is to form a linear characteristic $T_{lin}(N)$.

Fig. 5 shows the generated linear characteristic $T_{lin}(N)$ compared to the nonlinear characteristic $T(N)$.

Formation of linear characteristic $T_{lin}(N)$ so that it connects the starting point of the nonlinear characteristic $T(N)$, when the value at the ADC output $N_{\min}$ corresponds to the temperature $T_{\min} = 233\ K$, and the end point of the range, when the value at the ADC output $N_{\max}$ corresponds to the temperature $T_{\max} = 383\ K$, i.e., it corresponds to the expression:

$$T_{lin}(N) = a \times N + T_{\min},$$

where a — characteristic slope coefficient, which is determined by the formula:

$$a = \frac{T_{\max} - T_{\min}}{N_{\max} - N_{\min}}.$$

Then, for the correcting function should be equal to:

$$\Delta T(N) = T(N) - T_{lin}(N).$$

That is, to obtain the measured temperature value at the output of the "*Microcontroller Subsystem*" module, it is necessary to add to the calculated temperature value by the linear function $T_{lin}(N)$ the correction value at point N, which corresponds to ΔT at point N. This value can be calculated using the polynomial function:

$$P(N) = c_n N^n + c_{n-1} N^{n-1} + c_{n-2} N^{n-2} + \cdots + c_1 N^1 + c_0$$

where $c_n, c_{n-1}, c_{n-2}, \ldots c_1, c_0$ — polynomial coefficients, which are determined by the polynomial trend line of the function $\Delta T(N)$. In this model, this is done using Excel using saved time series data or an array in the basic MATLAB workspace by the "*Simout ADC*" module group.

Therefore, to calculate the measured temperature value at the output of the "*Microcontroller Subsystem*" module, the value of the polynomial $P(N)$ at point N is added to the calculated temperature value according to the linear function $T_{lin}(N) = a \times N + T_{\min}$, which corresponds to the value of ΔT at point N, and therefore the measured temperature value is equal to:

$$T(N) = a \times N + T_{\min} + P(N).$$

To do this, the value N from the ADC output is fed to 6 polynomial evaluation modules, which calculate the value of the polynomial $P(N)$ with a given polynomial array of coefficients c depending on the value of N. The polynomial array of coefficients for each of the subranges is determined separately. Depending on the current value of N, the corresponding module $P(u)$ is connected, where the value of the polynomial $P(u)$ is cleared with the corresponding subrange given polynomial array of coefficients. These values

are added to the formed linear characteristic $T_{lin}(N)$. Also, the "*Microcontroller Subsystem*" module implements the possibility of correcting the output characteristic according to the calibration results, and introducing correction coefficients for the self-heating correction of the thermistor depending on the thermal conductivity of the environment.

The "*Solver Configuration*" module defines general simulation parameters. The obtained data of each of the obtained values is recorded by the oscilloscope "*Scope*" (Fig. 6) and entered using the "*Simout*" module group into the specified time series or array in the basic MATLAB workspace for further data processing.

"*Calibration Data*" block consists of "*Constant*" modules, in which the values of the coefficients K_{298-}, K_{298+} and $T_{298+/-}$ and $K_{self-heating}$, which allows the calibration data to be used to adjust the original measurement data accordingly to compensate for the influence of possible causes of error.

To generate the current temperature value over a time series, the "*Repeating Sequence*" model is used, where a sequence of numbers time-temperature value is entered. Next, the temperature values are converted from Simulink format to physical signal data-temperature in *K* using the "*Simulink-PS Converter*". This data is used to compare the measured temperature value with the set value corresponding to the current thermistor resistance value.

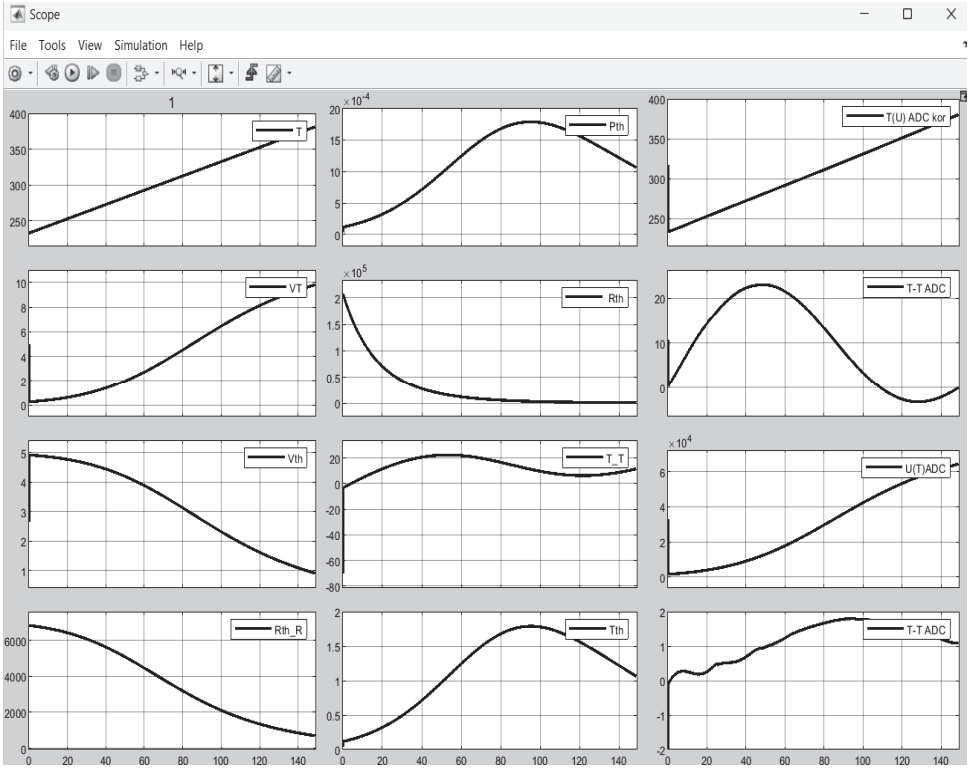


Fig. 6. Graphs of changes in parameter values during the simulation of temperature measurement in MATLAB Simulink

RESULTS OF THE RESEARCH

The research was conducted over the full temperature range of the thermistor, from 233 *K* to 383 *K* (i.e., from –40°*C* to +110°*C*). The operational temperature range and parameters of the thermistor corresponded to those of an RH18-type thermistor manufactured by Mitsubishi, with a nominal resistance of $(R) = 10\,\text{k}\Omega$ at 25°*C* and a $B_{25/85}$ constant of 3435 *K*. This thermistor is encapsulated in epoxy resin and features compact dimensions (1.8 mm in diameter and 7 mm in length), making it particularly sensitive to self-heating effects (thermal dissipation constant $\delta_{th}) = 1\,\text{mW}/^\circ\text{C}$ in air). This characteristic enables a comprehensive analysis of the impact of linearization methods on measurement error across the entire temperature range, as well as the determination of correction values accounting for the self-heating effect of the thermistor.

During simulation, temperature measurement error was evaluated at the analog output of the amplifier, both without correction and with the application of digital signal processing using a high-order polynomial function for data correction over the full temperature range. Additionally, the measurement range was divided into three and six smaller subranges, where lower-order polynomial functions were applied, respectively, to evaluate the influence of segmentation on measurement accuracy.

Fig. 7 illustrates the dependence of measurement error on the current temperature value. The corresponding measurement data are presented in tabular form.

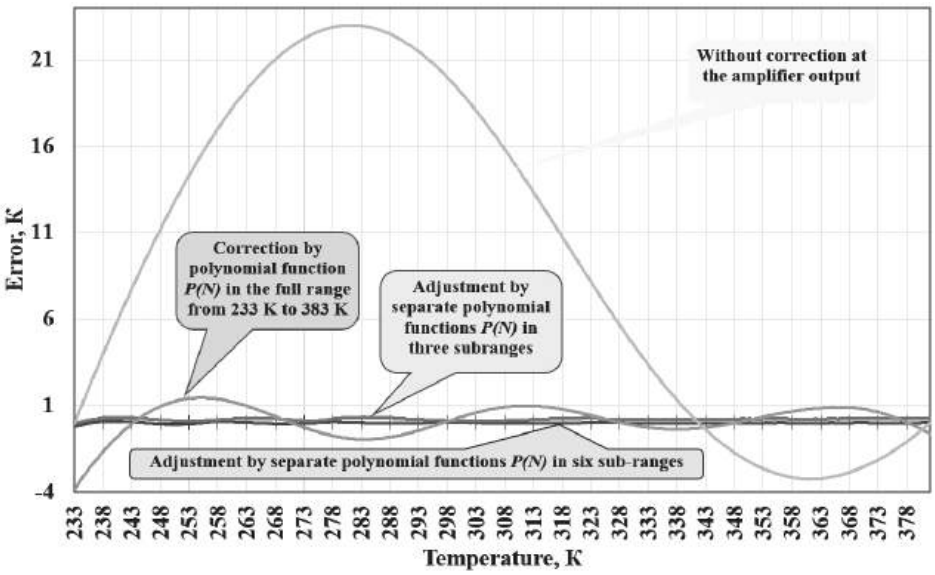


Fig. 7. Dependence of measurement error on the current temperature value

As shown in Fig. 7 and the data table, increasing the number of subranges into which the full measurement range is divided results in a reduction of measurement error and allows for the use of lower-degree polynomial correction functions.

Table 1 presents the simulation results for the digital signal processing algorithm used to correct the measured data, based on polynomial functions of varying orders applied over the full temperature range and over its division into three and six smaller subranges.

Table 1. Research results data

Number of subbands	Adjustment	Error	
		Average value ($\overline{\Delta T}$), K	Root mean square value σ, K
1 (Full)	No adjustment	9.589	9.518
1(Full)	6th order polynomial	0.145	0.859
3 subbands	4th order polynomial	0.111	0.055
6 subbands	3rd order polynomial	-0.002	0.036

Fig. 8 presents the actual temperature measurement error values at resistance points $R_{\min}$, R_{nom} and $R_{\max}$, obtained using third-order polynomial correction functions within each of the six subranges. The values of $R_{\min}$ R_{nom}, and $R_{\max}$, over the temperature range from 233 K to 383 K (– 40°C to +110°C) were taken from the datasheet provided by the manufacturer of the RH18 6Y103 thermistor (Mitsubishi Materials).

As observed from the plots, the minimum measurement error occurs at the nominal temperature of 298 K (+25°C), while the error increases as the temperature approaches the lower $T_{\min}$ and $T_{\max}$ bounds of the range. Calibration of the device allows for the determination of additional correction coefficients, denoted as K_{298-}, K_{298+} and $T_{298+/-}$ aimed at reducing the measurement error.

The coefficient K_{298-} — compensates for the slope deviation of the linear characteristic within the range from 233 K to 298 K (–40°C to +25°C), while K_{298+} serves the same purpose in the range from 298 K to 383 K (+25°C to +110°C). The coefficient $T_{298+/-}$ represents a temperature offset at the nominal point T_N =298 K (+25 °C), as determined during calibration. The developed model provides the capability to input these coefficients to enhance measurement accuracy.

Fig. 8 demonstrates the potential for reducing measurement error through the inclusion of the correction coefficients K_{298-}, K_{298+} and $T_{298+/-}$.

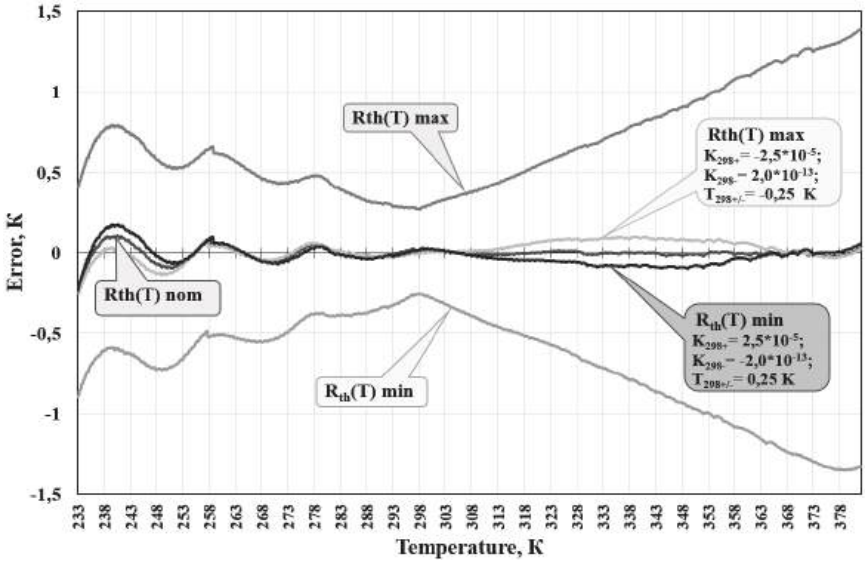


Fig. 8. Actual value of temperature measurement error at $R_{\min}$, R_{nom} and $R_{\max}$ when correcting measured data using 3rd order polynomial functions in each of the 6 subbands

Fig. 9 shows the actual temperature measurement error under conditions of a reference voltage shift in the Wheatstone bridge circuit, U_{ref} on ΔU_{ref} = ±0.05V (±0.5 %), using third-order polynomial correction functions in each of the six subranges. The results are shown for both uncalibrated conditions and after applying the additional calibration coefficients.

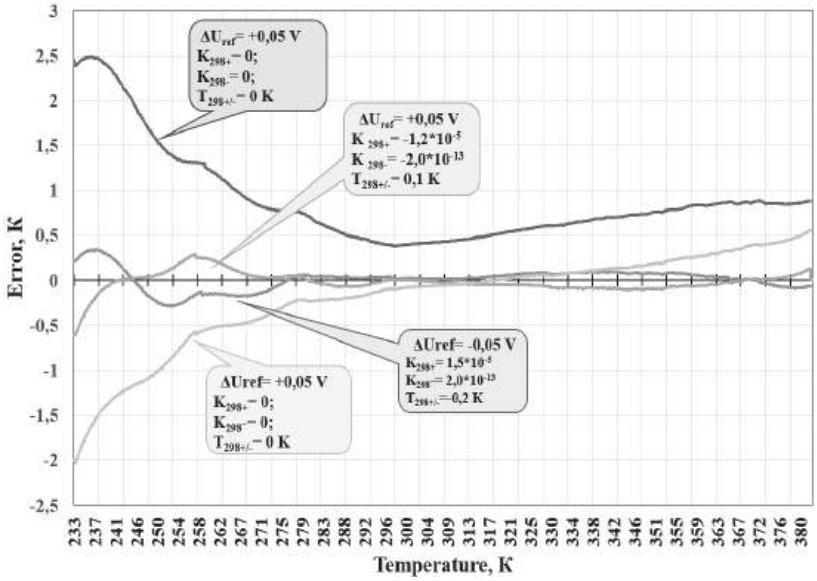


Fig. 9. Actual value of temperature measurement error when shifting the reference voltage of the Wheatstone bridge U_{ref} at ΔU_{ref} = ±0.05 V

Fig. 10 presents the actual temperature measurement error resulting from a deviation of the amplifier gain coefficient from its ideal value by ΔG = ±0.01, using third-order polynomial correction functions within each of the six subranges. The results are shown both without and with the application of additional calibration coefficients.

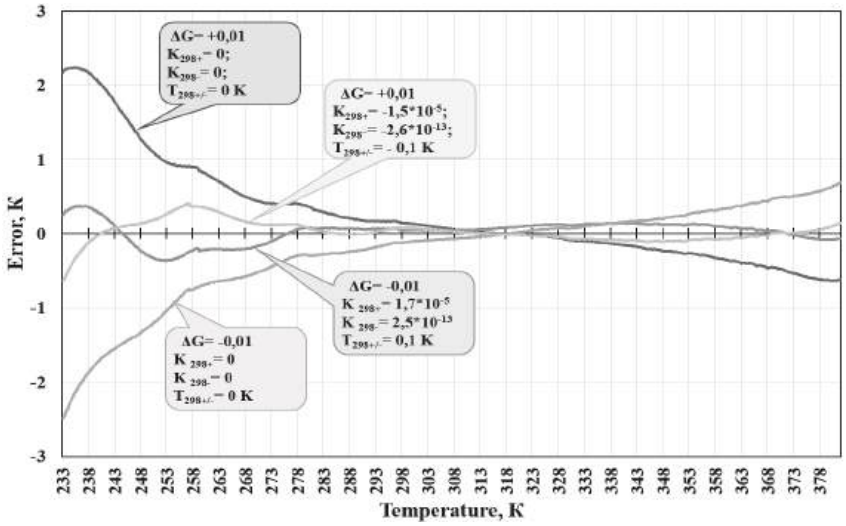


Fig. 10. The actual value of the temperature measurement error when the actual value of the amplifier gain deviates from its ideal value by ΔG = ±0.01

Fig. 11 illustrates the actual temperature measurement error due to thermistor self-heating, corrected using third-order polynomial functions within each of the six subranges, with the inclusion of various values of the $K_{self-heating}$ coefficient: 0, 0.25, 0.5, 0.75, and 1.0.

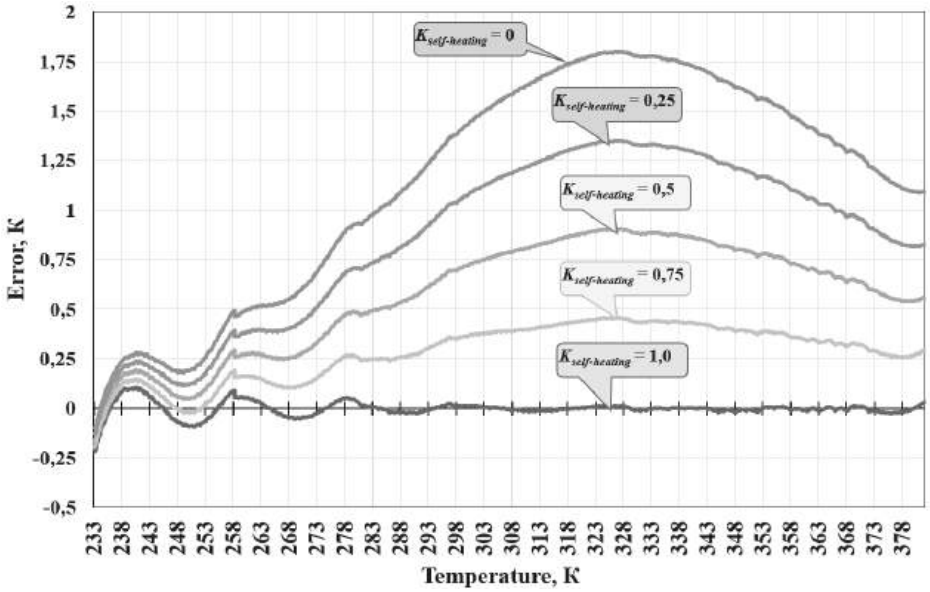


Fig. 11. Actual value of temperature measurement error taking into account thermistor self-heating when correcting measured data using 3rd order polynomial functions in each of 6 subranges with the input of $K_{self-heating}$ = 0; 0.25; 0.5; 0.75; 1.0

The summarized data of the research results are presented in Table 2.

To investigate the measurement error caused by the self-heating effect of the thermistor and the potential for compensating this error, the model includes a dedicated module, "*Subsystem self-heating*". This module determines the self-heating temperature of the thermistor at a given ambient temperature under still air conditions, which is used as the baseline reference. This approach is justified by the fact that thermistor manufacturers typically specify the thermal dissipation constant δ_{th} only for still air.

When the thermistor operates under different environmental conditions (e.g., in liquids or moving air), the actual δ_{th} value must be determined experimentally. Subsequently, the self-heating correction coefficient $K_{self-heating}$ should be adjusted proportionally relative to the value of $K_{self-heating}$ determined under still air conditions.

In the case of stationary water, which has a significantly higher thermal conductivity compared to air, the thermal dissipation constant δ_{th} increases by a factor of 2 to 5, depending on the thermistor's design, the material of its protective coating, and other structural factors. Consequently, the self-heating temperature of the thermistor is reduced, and a correction must be introduced by applying a corresponding self-heating compensation coefficient $K_{self-heating}$ typically ranging from 0.2 to 0.5.

Table 2. Summary of research results

Parameter	Calibration factors				Error	
	K_{298+}	K_{298-}	$T_{298+/-}$	$K_{self-heating}$	$\overline{\Delta T}$, K	σ, K
R_{nom}	-				-0.002	0.036
$R_{\min}$	-				0.678	0.305
	$2.5*10^{-5}$	$-2.0*10^{-13}$	0.25 K	-	-0.018	0.057
$R_{\max}$	-				-0.689	0.309
	$-2.5*10^{-5}$	$2.0*10^{-13}$	-0.25 K	-	0.016	0.057
U_{ref} = 10.00 V	-				-0.002	0.036
ΔU_{ref} =+0.05 V (+0.5%)	-				0.896	0.525
	$-1.2*10^{-5}$	$-2.0*10^{-13}$	0.1 K	-	0.011	0.119
ΔU_{ref} = –0.05 V (–0.5%)	-				-0.198	0.529
	$1.5*10^{-5}$	$2.5*10^{-13}$	-0.2 K	-	-0.018	0.108
G = 1.6	-				-0.002	0.036
ΔG = +0.01 (+0.63%)	-				0.258	0.695
	$-1.5*10^{-5}$	$-2.6*10^{-13}$	-0.1 K	-	0.026	0.150
ΔG = –0.01 (–0.63%)	-				-0.246	0.659
	$1.7*10^{-5}$	$2.5*10^{-13}$	0.1 K	-	0.026	0.140
$\Delta T_{self-heating}$	-			0	1.141	0.551
				0.25	0.855	0.413
				0.5	0.569	0.276
				0.75	0.282	0.141
				1.0	-0.002	0.036

CONCLUSIONS

The issue of sensor characteristic linearization is critically important for real-time applications across various fields, as most sensors exhibit nonlinear behavior. Digital linearization methods offer greater flexibility and improved measurement accuracy over a wide range of measured quantities. These methods can be implemented using software on a personal computer or via dedicated hardware platforms such as microcontrollers, FPGAs, or DSP processors.

This work proposes a method for linearizing the temperature characteristics of NTC thermistors using digital correction based on polynomial functions. The proposed approach can be implemented on low-power, cost-effective microcontrollers through sequential execution of programmed arithmetic operations.

The method was validated using a newly developed simulation model of a temperature measurement device based on an NTC thermistor, created in the MATLAB Simulink environment. This model includes a data processing algorithm capable of generating all necessary polynomial functions, incorporating correction coefficients obtained through calibration.

Simulation results demonstrated that the highest measurement accuracy over a wide temperature range can be achieved using a polynomial correction algorithm that divides the full measurement range into multiple subranges. Increasing the number of subranges enables the use of lower-order polynomial functions within each subrange, which in turn enhances overall measurement accuracy. Specifically, when the full temperature range from 233 K to 383 K was divided into six subranges, third-order polynomial functions were used for data processing. Under these conditions, the measurement error did not exceed 0.1 K (RMS).

The research identified several factors that influence measurement accuracy: deviations of the actual thermistor resistance from its nominal value due to the temperature dependence of the B-parameter; instability of the Wheatstone bridge reference voltage ΔU_{ref}; deviation of the amplifier gain from its ideal value ΔG; and errors introduced by self-heating of the thermistor. To achieve a measurement error within 0.1 K (RMS), the accuracy of the B-parameter, ΔU_{ref}, and ΔG must be within 0.5%, and the thermistor current I_{th} must be limited to no more than 100 μA. At this current level, self-heating of the thermistor does not exceed 0.1 K and therefore does not significantly affect measurement results.

To further enhance the effectiveness of temperature characteristic linearization of NTC thermistors based on the proposed method, future research will focus on improving the mathematical model. This will support the design and analysis of various temperature measurement circuits aimed at achieving the desired measurement accuracy using optimally selected sensors and components of the measurement channel.

REFERENCES

1. T. Islam, S.C. Mukhopadhyay, “Linearization of the sensors characteristics: a review,” *Int. J. Smart Sens. Intell. Syst.*, vol. 12, no. 1, pp. 1–21, 2019. doi: https://doi.org/10.21307/ijssis-2019-007
2. A.J. Lopez-Martin, A. Carlosena, “Sensor signal linearization techniques: A comparative analysis,” in *2013 IEEE 4th Latin Amer. Symp. Circuits Syst. (LASCAS)*, *Cusco, Feb. 27–Mar. 1, 2013*, pp. 1–4. doi: https://doi.org/10.1109/lascas.2013.6519013
3. J.M. Dias Pereira, P.M.B. Silva Girao, O. Postolache, “Fitting transducer characteristics to measured data,” *IEEE Instrum. & Meas. Mag.*, vol. 4, no. 4, pp. 26–39, 2001. doi: https://doi.org/10.1109/5289.975463
4. T. Nenov, S. Ivanov, “Linearization of characteristics of relative humidity sensor and compensation of temperature impact,” *Sensors Mater.*, vol. 19, no. 2, pp. 095–106, 2007.
5. H. Erdem, “Implementation of software-based sensor linearization algorithms on low-cost microcontrollers,” *ISA Trans.*, vol. 49, no. 4, pp. 552–558, 2010. doi: https://doi.org/10.1016/j.isatra.2010.04.004
6. “Chip NTC Thermistor Simulation | Sensors and Sensor Systems - Temperature Sensors (NTC) - Chip NTC Thermistors (Sensor),” *TDK Product Center*. Accessed on: May. 5, 2025. [Online]. Available: https://product.tdk.com/en/search/sensor/ntc/chip-ntc-thermistor/simulation
7. “NTC Thermistor Performance Simulator,” *Murata Manufacturing Co*. Accessed on: May 5, 2025. [Online]. Available: https://ds.murata.co.jp/simsurfing/ntcthermistor.ht

ml?rgear=suaykx&rgearinfo=com&md5=67c837df0f254f67edb244383dec4b71
8. "Chip thermistor resistance simulator | SOLUTIONS / KNOWLEDGE | Electronic materials and components | Mitsubishi Materials," *Mitsubishi Materials*. Accessed on: May 6, 2025. [Online]. Available: https://www.mmc.co.jp/adv/en/solution/simulator.html
9. J.G. Webster, *Measurement, Instrumentation and Sensors Handbook*. Taylor Francis Group, 1998, 2608 p. doi: https://doi.org/10.1201/9781003040019
10. L.E. Bengtsson, "Lookup table optimization for sensor linearization in small embedded systems," *J. Sensor Technol.*, vol. 02, no. 04, pp. 177–184, 2012. doi: https://doi.org/10.4236/jst.2012.24025
11. S.B. Stankovic, P.A. Kyriacou, "Comparison of thermistor linearization techniques for accurate temperature measurement in phase change materials," *J. Phys.: Conf. Ser.*, vol. 307, Art. no. 012009, 2011. doi: https://doi.org/10.1088/1742-6596/307/1/012009
12. "NTC Thermistors, General technical information," *TDK Electronics - TDK Europe*. Accessed on: May 6, 2025. [Online]. Available: https://www.tdk-electronics.tdk.com/download/531116/19643b7ea798d7c4670141a88cd993f9/pdf-general-technical-information.pdf
13. J. Jovanović, D. Denić, "NTC thermistor nonlinearity compensation using wheatstone bridge and novel dual-stage single-flash piecewise-linear ADC," *Metrol. Meas. Syst.*, vol. 28, no. 3, pp. 523–537, 2021. doi: https://doi.org/10.24425/mms.2021.136616
14. H. Ebrahimi-Darkhaneh, "Measurement error caused by self-heating in NTC and PTC thermistors," *Analog Des. J.*, pp. 1–8, 2019. Available: https://www.ti.com/lit/an/slyt774/slyt774.pdf?ts=1736163042225
15. S. Matvienko, S. Vysloukh, A. Matvienko, A. Martynchyk, "Determination thermal and physical characteristics of liquids using pulse heating thermistor method," *Int. J. Eng. Res. & Sci.*, vol. 2, no. 5, pp. 250–258, 2016.
16. G. Tymchik, S. Matvienko, I. Sikorsky, P. Kisała, K. Nurseitova, A. Iskakova, "Improving the way of determination substances thermal physical characteristics by direct heating thermistor method," *Przeglad Elektrotechniczny*, vol. 1, no. 4, pp. 123–128, 2019. doi: https://doi.org/10.15199/48.2019.04.21
17. S. Matvienko, S. Vysloukh, O. Martynchyk, "Increasing accuracy of measuring thermal conductivity of liquids by using the direct heating thermistor method," *Eastern-Eur. J. Enterprise Technol.*, vol. 4, no. 5(82), pp. 20–30, 2016. doi: https://doi.org/10.15587/1729-4061.2016.75459
18. M.V. Nikolic, B.M. Radojcic, O.S. Aleksic, M.D. Lukovic, P.M. Nikolic, "A Thermal Sensor for Water Using Self-Heated NTC Thick-Film Segmented Thermistors," *IEEE Sensors J.*, vol. 11, no. 8, pp. 1640–1645, 2011. doi: https://doi.org/10.1109/jsen.2010.2103309

Received 11.06.2025

INFORMATION ON THE ARTICLE

Sergey N. Matvienko, ORCID: 0000-0002-7547-4601, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: s.matvienko@kpi.ua

Grygoriy S. Tymchyk, ORCID: 0000-0003-1079-998X, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: deanpb@kpi.ua

СПОСІБ ЛІНЕАРИЗАЦІЇ ТЕМПЕРАТУРНИХ ХАРАКТЕРИСТИК NTC-ТЕРМІСТОРІВ ЗА ДОПОМОГОЮ ПОЛІНОМІНАЛЬНОГО МЕТОДУ / С.М. Матвієнко, Г.С. Тимчик

Анотація. Мета роботи – розроблення схемотехнічного способу лінеаризації температурних характеристик терміcторів із використанням поліномінального цифрового методу температурних характеристик NTC-терміcторів та характеристик розробленого вимірювального каналу за допомогою розробленої оригінальної моделі в MATLAB Simulink. Запропоновано модель пристрою для вимірювання температури із сенсором на базі термістора. Виконано моделювання поліномінального цифрового методу лінеаризації температурних характеристик NTC-терміcторів розробленого пристрою. Для підвищення точності вимірювань вибрано елементи вимірювального каналу з параметрами, що дають змогу здійснювати моделювання вимірювань із мінімальною похибкою. Визначено методи внесення значення корекційних коефіцієнтів у модель для компенсації факторів впливу на похибку включно з ефектом саморозігріву термістора. Запропонований алгоритм оброблення даних має переваги за реалізації в мікроконтролері невисокої вартості та низької потужності.

Ключові слова: вимірювання температури, NTC-термістор, MATLAB Simulink, лінеаризація.

UDC 629.5.017.3:621.391
DOI: 10.20535/SRIT.2308-8893.2026.1.05

PRACTICAL ASPECTS OF CREATING A DATA TRANSMISSION SYSTEM FOR CONTROLLING UNMANNED SURFACE VEHICLES IN UNSTABLE COMMUNICATION CHANNELS

S.V. KURDIUK, O.M. MELNYK, O.A. ONISHCHENKO, S.M. VOLIANSKYY, V.A. SHEVCHENKO, B.M. ALIEKSIEICHUK

Abstract. The study presents the development and verification of an adaptive data transmission system for controlling unmanned surface vehicles (USVs) in unstable communication channels. The work aims to overcome the limitations of existing technologies, which include LTE networks and satellite systems that fail to deliver stable service quality for USV remote control operations. The proposed adaptive routing algorithm evaluates communication channel status through three vital indicators, which include delay, packet loss, and availability. The algorithm selects the best channels according to changing weight parameters. Experimental results confirmed a significant reduction in data transmission delays, stable real-time video streaming with a delay of 1–4 seconds, and a reduction in packet loss to below 2 %. In addition, the system implements the use of modern video coding standards (e.g., H.265) and secure VPN channels, which increase bandwidth efficiency and the level of cybersecurity. The results confirm the practical suitability of the proposed system for USV operation in real marine conditions, as well as its potential for use in critical scenarios that require stable, low-latency communication.

Keywords: adaptive data transfer, unmanned vehicles, handling, maneuvering, navigation safety, communication channels, course control, routing algorithm, delay optimization, loss reduction, data packets, operational efficiency, status monitoring, 5G integration, predictive machine learning models.

INTRODUCTION

Unmanned Surface Vehicles (USVs) have become widely used in various industries and defense applications. Their versatility is due to a combination of high autonomy, navigation accuracy, and the ability to perform tasks in difficult and potentially dangerous conditions where the use of manned vessels is economically or safety unjustified. One of the major aspects of their efficiency is the reliance on communication mechanisms, which provide real-time data exchange and facilitate the distant or self-governing control functions.

Despite progress in satellite technology and wireless networks, data channels for USVs remain vulnerable to bad weather, congested networks, and interference in areas of heavy ship traffic, which results in delays, packet losses, and communication disconnections, significantly lowering the reliability of important operations, including video streaming, remote control, and autonomous navigation. Thus, the development of very reliable and adaptive communication solutions is the only way to assure the uninterrupted operation of USVs in the highly dynamic marine locations.

With the development of data transmission technologies and the growth of mobile operators and satellite communications infrastructure, the need for a reliable communication system to control unmanned surface vehicles in unstable data links is increasing. Such systems are essential for efficient and secure remote control, especially when streaming live video and maintaining stable communication with mobile objects. However, current IP networks, which were not initially designed for online video transmission, face challenges such as packet loss and significant latency. These factors critically impact broadcast quality of experience (QoE), especially for streams compressed to modern standards such as H.265, which depend on link stability and bandwidth.

Moreover, LTE mobile networks and satellite systems like Starlink might run into impediments such as intermittent sporadic disconnectivity and exacerbated-latencies bursts from packet transmissions. In the backdrop of a situation of uncertainty, whereby perhaps signals from mobile companies would be erratic, and where amplitude variations on the dish would disallow constant quality in the provision of digital services; this situation would necessitate ad-hoc channel hopping solutions to ensure video transmission is successfully spread across channels with the least possible latency.

Current research principally focuses more on improving network variables or increasing the stability of individual links in the com-mon mesh scenario. Not much of the work has been directed at the design on a routing-system based on multiple adaptive channels for UAVs which consider their environmental dynamic nature. In the sense of technical overhand, this article attempts to fill in that space by introducing a new algorithm offers an absolute optimization for ad-hoc transmission of data.

RELATED WORK

The current approaches to maintaining quality of communication with unmanned surface vehicles (USVs) on unstable data links include a wide range of techniques to minimize latency, packet loss, and link resilience. Previous research in USV communication can be categorized into three main areas: signal and interference robustness, adaptive algorithms for dynamic environments, and multi-channel approaches for fault tolerance. This categorization allows us to identify the gaps the proposed research aims to address.

In 2023, 3GPP presented an overview of support for the NR (New Radio) standard for USVs, emphasizing the adaptation of modern networks for seamless data transmission and control in a multi-channel network environment [1]. However, current technologies are under-researched in terms of adaptive channel selection under dynamic network variability, which requires further development of algorithms that can take into account the parameters of current network conditions to improve quality of service (QoS) and quality of experience (QoE) perception.

Research in video streaming emphasizes the critical impact of packet loss and delay variations on real-time quality. In [2], they considered how delay and packet loss variations reduce the QoE perception, which is especially relevant for USV streaming, where delays can become critical for the operator. Other works, such as [3], have proposed a QoE prediction model for multimedia services but have not provided an adaptive solution capable of dynamically adjusting to changes in network conditions. In contrast, [4] proposed an approach to improve QoE in wireless

network conditions using multiple coding, improving data transmission reliability in a variable bandwidth network.

The predictive and adaptive QoE control techniques for video streaming proposed by [5] allow for estimating the current network parameters and adjusting the routing parameters based on changes in transmission conditions. However, their adaptation framework does not cover specialized solutions for multi-channel routing in drone environments, where latency and packet loss during control are critical. At the level of specific network technologies, [6, 7] discusses the importance of minimizing control channel delay for remote control systems, showing that LTE and 4G networks pose significant limitations regarding stability and latency. However, their research focuses more on theoretical estimates of delay parameters rather than the practical use of adaptive routing algorithms for continuous drone control on unstable communication channels [8].

Routing optimization and traffic management techniques, such as the one proposed in [9], offer solutions that provide fault tolerance and security to networks under multi-criteria routing conditions. However, applying these solutions to unmanned aerial vehicle (UAV) scenarios is limited because they do not provide dynamic adaptation to real-time channel changes, which is critical for link stability in drone control systems [10, 11]. The authors in [12] also proposed complex optimization methods for self-organizing networks, which can theoretically improve adaptive routing control, but requires refinement for practical application under high load conditions and frequent link switching.

Based on the analysis of existing research, it can be seen that although many approaches have been developed to improve QoE and reduce data transmission latency, a significant gap remains in adaptive routing for highly loaded, unstable links used for drone data transmission [13–16]. This study seeks to address this gap by proposing an algorithm capable of dynamically accounting for changes in network parameters and adjusting data routes to ensure high stability and quality of real-time drone communications [17, 18]. Additional research offers unique approaches to improve security and resilience in complex environments. In [19], radar-based methods for object detection and recognition on water were investigated, highlighting the importance of reliability and accuracy of data transmission in unstable environments similar to those observed in control. [20, 21] developed polarization-based approaches to improve object identification and safety under challenging data transmission conditions applicable to tasks. In [22] investigated the energy efficiency of motors, which can further contribute to the sustained operation of drones, especially in environments where channel quality and reliability are critical to maintain control.

Reliable two-way communication with UAVs is critical for efficient control and data exchange. Several studies have focused on energy efficiency and system optimization. For example, the energy efficiency improvement of electric motors in autonomous vehicles was investigated in [23], and energy-efficient positioning systems for multi-purpose ships were proposed in [26]. In [24], a simulation-based method for predicting the seaworthiness of vessels applicable to UAV performance modeling was developed, and operational efficiency in transportation projects was evaluated [25].

Communication protocols have also been a key area of research. Shi et al. reviewed protocols for UAV inspections [27] and studied optimal power allocation methods [28]. In [29], UAV swarm architecture for efficient data routing. Sources [30–35] are devoted to current research in the field of unmanned aerial vehicles and

marine technologies. The research focuses on designing two-way communication systems for UAVs and optimizing secure communication through full-duplex systems and RIS technology and implementing solar power solutions on commercial vessels for emergency fire protection. The combination of these research studies shows progress in multiple fields which include communication protocols and cyber security and energy sustainability and intelligent control systems.

Intelligent control systems and methods for enhancing maneuverability and optimizing energy consumption in transportation [36–39] are crucial for developing practical solutions in UAV control. Data security and threat countermeasures [40, 41] are vital for ensuring sustainable communication and defense in unmanned systems. The extension of mathematical tools and intelligent approaches for dynamic object control [42, 43] applies to UAV navigation. Threat analysis, obstacle avoidance techniques, safety of cargo carriage and operator interaction [44, 45] form a basis for creating reliable and safe UAV control systems. Sources [46–49] highlight key aspects of unmanned aerial vehicle development: inertial navigation accuracy, control system improvements, and spatial route optimization, demonstrating that current research is focused on improving the reliability and efficiency of drones in challenging operating conditions.

An analysis of existing research shows that, despite the significant number of approaches proposed to improve quality of service (QoE) and reduce data transmission delays, there remains a significant gap in adaptive routing for highly loaded and unstable channels used in unmanned aerial vehicle data transmission systems.

This paper attempts to address this shortcoming by developing an algorithm that can dynamically respond to changes in network parameters and adjust information transmission routes, ensuring high stability and quality of communication in real time.

The proposed approach covers the transmission of control signals, telemetry, and streaming video with minimal delays, and also implements the protection of information flows using VPN tunneling. As part of the study, experiments were conducted on the selection of routers and the optimization of their parameters in order to improve the efficiency of data transmission under conditions of unstable communication channels.

MATERIALS AND METHODS

Problem statement and system description

Modern data transmission systems for unmanned surface vehicle (USV) control require a stable connection that delivers control signals, telemetry, and streaming video. These data types have different quality of service requirements: minimum latency and high accuracy are critical for control signals while streaming video requires high bandwidth and minimal packet loss to preserve image quality. The basis of the proposed system is the use of two communication channels: LTE (4G) mobile networks and Starlink satellite system. LTE provides a wide coverage area and affordable data rates but is prone to instability in conditions of congestion or weak signal. Starlink, on the other hand, provides a more stable connection through low-orbit satellites, but is subject to signal fluctuations due to antenna movement and view limitations. In real-world conditions, none of these technologies can guarantee constant quality of service (quality of service), which requires the development of solutions that adapt to the cur-

rent state of the network. In real-world conditions, none of the existing wireless technologies can guarantee consistent Quality of Service (QoS), which necessitates the development of adaptive solutions capable of responding to the current state of the network. In particular, instability is observed in LTE mobile communications, especially in areas with heavy traffic or insufficient signal coverage. According to research results, typical problems include connection drops and increased data transmission delays. Fig. 1 shows that the time it takes to switch between channels can be tens of seconds, which significantly complicates the stable control of unmanned surface vehicles (USVs) and real-time data transmission.

Ping was selected as a universal tool for latency assessment due to its availability across all network environments. Although ICMP traffic can be deprioritized, our parallel measurements using UDP-based tools confirmed that ICMP delays closely mirrored the latency experienced by video and telemetry data (Fig. 1).

```
Reachable Hosts:
Reply from 192.168.89.1: bytes=32 time=74ms TTL=63
Reply from 192.168.89.1: bytes=32 time=73ms TTL=63
Reply from 192.168.89.1: bytes=32 time=75ms TTL=63
Reply from 192.168.89.1: bytes=32 time=76ms TTL=63
Reply from 192.168.89.1: bytes=32 time=80ms TTL=63
Reply from 192.168.89.1: bytes=32 time=75ms TTL=63

Unreachable Hosts:
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
Reply from 10.10.89.1: Destination host unreachable.
```

Fig. 1. Example of Ping command passing when switching between WAN links on Teltonika RUT series routers

Similar problems occur with the frequency of connection checking at routers, as shown in Fig. 2. Here, we can see that the minimum connection check time in the standard configuration of Teltonika routers is limited to 30 seconds, which is insufficient for rapid adaptation to changing channel conditions.

Fig. 2. Connection state polling interval parameter for RUT routers

The challenge is to design a data transmission system that minimizes latency, maintains stable transmission quality, and provides data protection. To achieve this, the following approaches are used:

– video stream tuning using modern compression algorithms (e.g., H.265) to reduce data volume without significant quality loss;

– VPN tunnels (e.g., based on OpenVPN or WireGuard) to encrypt transmitted data and improve its security;

– adaptive data routing using algorithms for assessing the current channel state (latency, packet loss, availability) to select the optimal channel in real-time.

Optimize equipment settings, including IP camera and router parameters, to coordinate bandwidth allocation, optimize encoding settings, and ensure stable transmission.

Thus, the proposed communication system focuses on complex signal processing to balance minimum latency, connection stability, and high data rate even in unstable channel conditions.

VPN encryption introduced an additional latency of 20–50 ms. We optimized MTU size, used hardware-accelerated encryption, and implemented fast packet re-routing to mitigate this. WireGuard was selected over OpenVPN due to its lower latency, simpler codebase, and faster handshake processes, making it more suitable for UAV real-time communication.

Data transfer methods and technologies

Distributed multi-channel data transmission methods and adaptive routing algorithms ensure stable communication with unmanned platforms even under unstable communication channel conditions. Such approaches minimize the impact of delays, packet loss, and communication failures by distributing information flows across multiple channels.

The developed communication system implements parallel data transmission through various media, including mobile networks (LTE) and the Starlink satellite system. For example, control commands can be transmitted via LTE, while video streams can be transmitted via Starlink and this distribution increases communication reliability and reduces the risk of complete loss of communication.

In addition, dynamic traffic distribution is applied, taking into account the current characteristics of the channels, such as bandwidth and latency. Switching between channels is based on specific criteria – latency, packet loss percentage, and channel availability.

Thus, if the delay in the LTE network exceeds the acceptable threshold (for example, 400 ms), the system automatically redirects traffic via Starlink. To improve the efficiency of the switching process, adaptive algorithms are used that take into account the weighting coefficients of each channel, which change dynamically depending on its current state.

A comparative analysis of various data transmission technologies has shown that the use of outdated communication standards, in particular 2G and GPRS, is accompanied by significant signal delays – from 124 to 2819 ms, which effectively makes it impossible to use them for tasks that require real-time data processing (Fig. 3).

```
C:\Users\Dr.Vad>ping 192.168.94.30 -t

Pinging 192.168.94.30 with 32 bytes of data:
Request timed out.
Reply from 192.168.94.30: bytes=32 time=825ms TTL=62
Reply from 192.168.94.30: bytes=32 time=124ms TTL=62
Reply from 192.168.94.30: bytes=32 time=154ms TTL=62
Reply from 192.168.94.30: bytes=32 time=108ms TTL=62
Reply from 192.168.94.30: bytes=32 time=100ms TTL=62
Reply from 192.168.94.30: bytes=32 time=551ms TTL=62
Reply from 192.168.94.30: bytes=32 time=2819ms TTL=62
Request timed out.
Request timed out.
Reply from 192.168.94.30: bytes=32 time=1841ms TTL=62
Request timed out.
Request timed out.
```

Fig. 3. Ping delays when modems operate in 2G and GPRS modes

In contrast, modern LTE modems that support channel aggregation significantly increase bandwidth. Fig. 4 shows that using an LTE Cat.6 modem allows you to combine 15 MHz and 10 MHz bands for a total channel width of up to 25 MHz. This improves link reliability and data transfer rates, especially for video streaming.

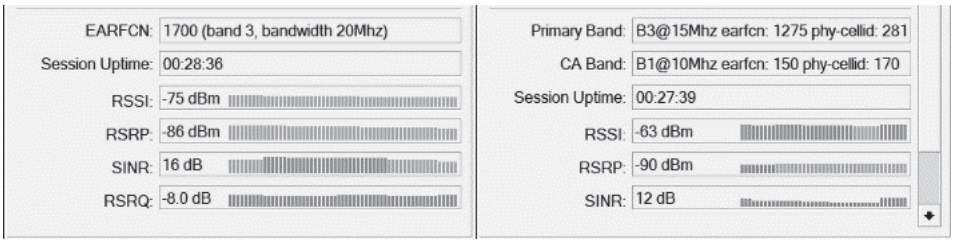


Fig. 4. Example of using channel aggregation (CA Band) on LTE Cat.6

Distributed data transmission and adaptive routing techniques allow optimal utilization of available communication channels, reducing latency and improving data transmission stability. This is especially important for drone control, where communication quality is critical to mission performance. The illustrations illustrate the advantages of modern data transmission technologies and the need to abandon outdated standards.

Adaptive channel selection algorithm

An adaptive routing algorithm ensures stable communication under unstable channel conditions by dynamically analyzing channel state and adjusting their priorities based on current parameters. The algorithm takes into account the following key indicators (Table 1).

Table 1. Key metrics used in the adaptive channel selection algorithm and their impact on prioritization

Metric	Description	Impact on Priority
Latency (Ping)	Measures the average response time of packets for each channel. If latency exceeds 400 ms, the channel receives a lower priority	Higher latency decreases the channel's priority
Packet Loss	Tracks the percentage of lost packets on each channel. Channels with high packet loss are excluded from routing or assigned penalty coefficients	High packet loss results in reduced priority or exclusion from routing
Channel Availability	Assesses the availability of the connection. If the channel is temporarily unavailable, its priority is automatically decreased	Unavailable channels are deprioritized or excluded
Weight Coefficients	Dynamically adjusted based on the channel's performance. Channels with low latency and minimal packet loss receive the highest priority	Optimized channels are prioritized based on their performance metrics

The logic of the algorithm is as follows:

1. Initialization. The initial channel weights are set (e.g., W1 = 3, W2 = 4, W3 = 5).

2. Checking packet losses. If the losses exceed the acceptable threshold, a penalty factor (e.g., +20) is added to the channel weight.

3. Checking the delay. If the delay is within the threshold, the channel is checked further; if the delay is above the threshold, a penalty factor (+10) is added to the weight.

4. Channel selection. The channel with the lowest penalty (highest priority) is selected from the available ones.

Fig. 5 shows the block diagram of the adaptive channel selection algorithm. It includes the following steps:

- initial initialization of parameters;
- packet loss analysis on the channel;
- ping delay verification;
- dynamic change of weighting coefficients;
- selection of the optimal channel for routing.

The advantage of this approach is that using this algorithm allows for reducing the switching time between channels. Minimize data loss in unstable conditions and Provide reliable traffic routing for drone control tasks.

The algorithm, in turn, can be implemented as a software script to run on routers that support dynamic routing, such as MikroTik devices. Its flexibility allows it to adapt to real-time network changes, making it a versatile solution for data communication systems.

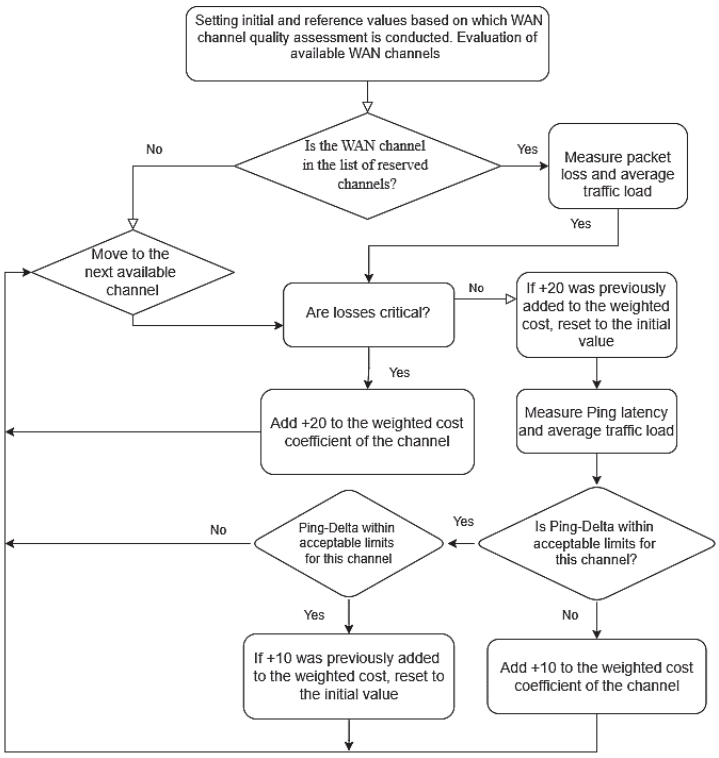


Fig. 5. Block diagram of the data link state quality assessment algorithm for traffic routing

Experimental validation and results

The experimental results confirm the effectiveness of the proposed methods for adaptive control of data links. The main objectives of the tests included reducing latency, improving link stability, and improving link capacity. Prior to system optimization, significant data transmission quality problems were observed.

Fig. 6 shows an example of unoptimized traffic where high latency and irregular link failures resulted in unstable network performance. Under high load conditions, packet losses reached up to 15–20%, making it impossible to control the drone in real time steadily.

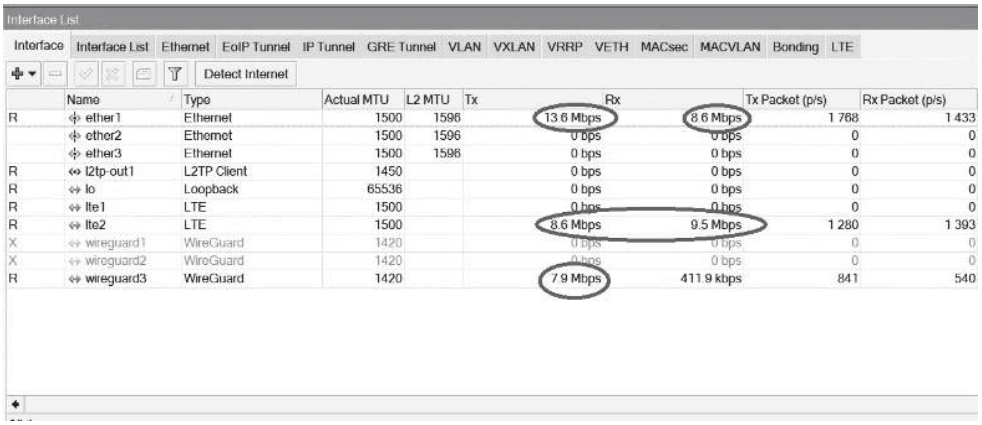


Fig. 6. Example of data traffic values with non-optimized IP camera settings

Significant improvements were observed after the implementation of adaptive channel selection algorithms. Fig. 7 demonstrates the optimized Ping latency values, which have been reduced to 50–150 ms. This meets real-time requirements and ensures reliable control of the USV. Packet loss was minimized to less than 2%, confirming the effectiveness of adaptive routing.

```
C:\Users\Dr.Vad>ping 192.168.92.30 -l 1380 -f -t

Pinging 192.168.92.30 with 1380 bytes of data:
Reply from 192.168.92.30: bytes=1380 time=107ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=98ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=95ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=114ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=115ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=129ms TTL=61
Request timed out.
Reply from 192.168.92.30: bytes=1380 time=118ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=103ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=102ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=117ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=114ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=91ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=125ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=94ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=114ms TTL=61
Request timed out.
Reply from 192.168.92.30: bytes=1380 time=91ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=115ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=119ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=110ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=91ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=94ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=125ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=133ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=92ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=208ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=117ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=91ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=115ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=118ms TTL=61
Reply from 192.168.92.30: bytes=1380 time=125ms TTL=61

Timestamp: 17:41:49
```

Fig. 7. Ping delay values in installed communication channels after implementation of recommendations and optimization of equipment settings

The results of the experiments showed the following:

1. Delay reduction on all channels was reduced by 60–70% on average, which ensured stable connection;

2. Stability improvement as the use of adaptive algorithms allowed to maintain stable data transmission even in conditions of high network load;

3. Optimization of throughput capacity. Due to dynamic routing, the load was evenly distributed between the channels, reducing the congestion of individual network segments.

These results confirm that implementing the proposed methodology of adaptive channel selection and equipment optimization can provide high-quality communication in drone control systems. These improvements prepare the system for real-world applications where link stability and minimum latency are critical.

RESULTS AND DISCUSSION

The study's results confirmed the effectiveness of the proposed adaptive routing system for data transmission in USV's control systems. The main achievements are significantly reducing data transmission delays, ensuring stable communication even under unstable network conditions, and minimizing packet loss. During

the experiments, it was possible to achieve video transmission with a delay of 1 to 4 seconds, corresponding to real-time requirements. The adaptive routing system proved its ability to dynamically select the optimal channel based on current network parameters such as latency, packet loss, and link availability. The fast switching between channels maintained high reliability and quality of service, making the system suitable for real-world applications. Baseline latency without our algorithm ranged from 8 to 12 seconds due to frequent retransmissions and unstable routing. Achieving 1–4 seconds with our adaptive routing is optimal for UAV operations in challenging network conditions, ensuring timely control and acceptable video quality.

Nevertheless, the work identified areas for further improvement. For example, it is possible to integrate channel state prediction techniques using machine learning algorithms, allowing for advance determination of optimal routes based on historical data and current trends. In addition, the use of modern video encoding technologies, in particular H.266/VVC, allows for a significant reduction in the amount of data transmitted without any noticeable loss of image quality. An additional area of optimization is the use of hardware with increased computing power, in particular routers with faster processors, which ensures more efficient traffic processing and routing.

Analysis of video recordings showed that in the absence of a developed algorithm, frame freezes, image pixelation, and increased latency due to packet loss are observed. After implementing the algorithm, the number of missed frames is significantly reduced, artifacts almost disappear, and the video stream remains stable even with changing network characteristics.

The developed system can be adapted to new communication standards, in particular 5G, which will provide higher bandwidth and minimal delays. This makes the proposed solution flexible and suitable for integration into scalable data transmission networks.

CONCLUSIONS

This paper presents a study aimed at the development and experimental verification of a data transmission system for controlling unmanned surface vehicles under conditions of unstable communication channels. The main limitations of modern communication technologies (LTE, Starlink) in their use for USV control are investigated. Special attention is paid to the impact of latency, packet loss, and link instability on quality of service (QoS). An algorithm that dynamically evaluates the state of available communication channels based on parameters such as delay, packet loss, and availability is proposed and described. A system of penalty coefficients is realized, allowing to correct channels' priority operatively.

The proposed system was tested on actual data, which showed that the use of adaptive routing allows for significantly reduced delays (up to 1–4 seconds) and minimized packet losses (up to less than 2%). Modern compression algorithms (H.265) and VPN tunnels were used to improve the security of transmitted data. This ensured more efficient utilization of channel bandwidth. A comparative analysis has shown that the enhanced data transmission system has evidently outclassed the non-enhanced one especially in the main indexes of connection stability, speed, and immunity against noise. Transmission delays, packet loss, and signal fluctuations, crucial variables for a reliable command of unmanned surface

vehicle (USV) activities, were minimized due to fine-tuning of the transmission parameters. The proposed architecture has shown resilience against environmental stress and channel instability, which shows its practical potential. The future direction could be integration of intelligent adaptation mechanisms specifically involving real-time channel status assessment in predictive machine learning models and linking next-generation communication technologies (5G) for elevating USV communication to the next level.

REFERENCES

1. "NR Support for UAVs," *3GPP*. 2023. Available: https://www.3gpp.org/ technologies/nr-uav
2. J. Frnda, M. Voznak, L. Sevcik, "Impact of packet loss and delay variation on the quality of real-time video streaming," *Telecommunication Systems*, 62(2), pp. 265–275, 2016. doi: https://doi.org/10.1007/s11235-015-0037-2
3. L. Sevcik, M. Voznak and J. Frnda, "QoE prediction model for multimedia services in IP network applying queuing policy," in *International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2014), Monterey, CA, USA, 2014*, pp. 593–598, 2014. doi: https://doi.org/ 10.1109/ SPECTS.2014.6879998
4. Farouk Boumehrez, Radhia Brai, Noureddine Doghmane, Khaled Mansouri, "Quality of experience enhancement of high efficiency video coding video streaming in wireless packet networks using multiple description coding," *Journal of Electronic Imaging*, 27(1), 013028, 2018. doi: https://doi.org/10.1117/1.JEI.27.1.013028
5. M. Taha, A. Canovas, J. Lloret, A. Ali, (2020). A QoE adaptive management system for high definition video streaming over wireless networks. *Telecommunication Systems*, vol. 77, pp. 63–81, 2021. doi: https://doi.org/10.1007/s11235-020-00741-2
6. A. Kutins, D. Brodnevs, "Determination of delay parameters in 4G LTE cellular mobile networks," *2022 Workshop on Microwave Theory and Techniques in Wireless Communications (MTTW), Riga, Latvia, 2022*, pp. 62–67. doi: https://doi.org/10.1109/ MTTW56973.2022.9942617
7. D. Brodnevs, A. Kutins, "Requirements of end-to-end delays in remote control channel for remotely piloted aerial systems," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 2, pp. 18–27, 2021. doi: https://doi.org/ 10.1109/MAES.2020.3039853
8. S.L. Volkov, V.V. Skachkov, V.I. Pavlovich, V.V. Chepkiy, "Information-entropy indicator of state quality for parametric systems in multi-criteria evaluation tasks," in *Proceedings of the Conference on Development Trends of Convergent Networks: Post-NGN, 4G and 5G Solutions*, pp. 23–27. Kyiv: State University of Telecommunications, 2016.
9. O. Lemeshko, O. Eremenko, O. Nevzorova, *Flow models and routing methods in infocommunication networks: Fault tolerance, security, scalability*. Kharkiv: Kharkiv National University of Radioelectronics, 2020. doi: https://doi.org/ 10.30837/978-966-659-282-1
10. "H.264 vs. H.265: Which Should You Use?" *Accsoon*, 2024. Available: https://accsoon.com/explore/h264-vs-h265-which-should-you-use/
11. "IP camera streaming guide: How to setup an IP camera," *Ant Media*, 2024. Available: https://antmedia.io/ip-camera-streaming-guide-how-to-setup-an-ip-camera/
12. A.V. Markovsky, G.N. Vlasenko, "Ensuring global internet access: Realities, prospects, and challenges," in *Proceedings of the Conference on Development Trends of Convergent Networks: Post-NGN, 4G and 5G Solutions*, pp. 30–34. Kyiv: State University of Telecommunications, 2016.
13. "Legal Terms," *Starlink*, 2024. Available: https://www.starlink.com/legal/documents/ DOC-1400-28829-70

14. M. Uhrina, J. Frnda, L. Sevcik, M. Vaculik, "Impact of H.264/AVC and H.265/HEVC compression standards on the video quality for 4K resolution," *Advances in Electrical and Electronic Engineering*, 12(4), pp. 421–428, 2016. doi: https://doi.org/10.15598/aeee.v12i4.1216
15. L. Nguyen, H.T. Nguyen, "Mobility based network lifetime in wireless sensor networks: A review," *Computer Networks*, vol. 174, 107236, 2020.
16. S. Kurdiuk et al., "Development of a high-reliability hybrid data transmission system for unmanned surface vehicles under interference conditions," *Drones*, 9(3), 174, 2025. doi: https://doi.org/10.3390/drones9030174
17. Y.V. Klymash, O.M. Shpur, M.V. Kaidan, "Complex optimization method for routing information flows in self-organized networks," *Bulletin of Lviv Polytechnic National University*, pp. 76–87, 2018. Retrieved from: https:// science.lpnu.ua/sites/ default/files/journal-paper/2018/jun/13512/12.pdf
18. "Bitrate and its place in video surveillance," *World Vision*, 2024. Available: http://worldvision.com.ua/articles/bitreyd-i-ego-mesto-v-videonablyudenii
19. "IP camera bandwidth calculator: Formula, example & tips," *Reolink*, 2024. Available: https://reolink.com/blog/ip-camera-bandwidth-calculation/
20. D. Korban, O. Melnyk, O. Onishchenko, S. Kurdiuk, V. Shevchenko, T. Obniavko, "Radar-based detection and recognition methodology of autonomous surface vehicles in challenging marine environment," *Scientific Journal of Silesian University of Technology. Series Transport*, vol. 122, pp. 111–127, 2024. doi: https://doi.org/10.20858/sjsutst.2024.122.7
21. M.S. Stetsenko et al., "Polarization-based target detection approach to enhance small surface object identification ensuring navigation safety," *System Research and Information Technologies*, no. 2, pp. 35–51, 2024. doi: https://doi.org/10.20535/SRIT.2308-8893.2024.2.03
22. O. Melnyk et al., "Full overlap ship security model: An integrative approach to shipboard equipment information security," *E3S Web of Conferences*, 501, Article 02002, 2024. doi: https://doi.org/10.1051/e3sconf/202450102002
23. Y. Volyanskaya, S. Volyanskiy, O. Onishchenko, S. Nykul, "Analysis of possibilities for improving energy indicators of induction electric motors for propulsion complexes of autonomous floating vehicles," *Eastern-European Journal of Enterprise Technologies*, vol. 2, no. 8 (92), pp. 25–32, 2018. doi: https://doi.org/10.15587/1729-4061.2018.126144
24. O. Melnyk, S. Onyshchenko, O. Onishchenko, O. Shcherbina, N. Vasalatii, "Simulation-based method for predicting changes in the ship's seaworthy condition under impact of various factors," *Studies in Systems, Decision and Control*, vol. 481, pp. 653–664, 2023. doi: https://doi.org/ 10.1007/978-3-031-35088-7_37
25. O. Melnyk, M. Malaksiano, "Effectiveness assessment of non-specialized vessel acquisition and operation projects, considering their suitability for oversized cargo transportation," *Transactions on Maritime Science*, vol. 9, no.1, pp. 23–34, 2020. doi: https://doi.org/10.7225/toms.v09.n01.002
26. O. Melnyk et al., "Fundamental concepts of deck cargo handling and transportation safety," *European Transport - Trasporti Europei*, issue 98, article 1, 2024. doi: https://doi.org/10.48295/ET.2024.98.1
27. L. Shi, N.J. Hernández Marcano, R.H. Jacobsen, "A review on communication protocols for autonomous unmanned aerial vehicles for inspection application," *arXiv preprint*, 2021. doi: https://doi.org/10.48550/arXiv.2111.06714
28. S. Nasrollahi, S.M. Mirrezaei, "Toward UAV-based communication: Improving throughput by optimum trajectory and power allocation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2022, article no. 9, 2022. doi: https://doi.org/10.1186/s13638-022-02087-6
29. X. Chen, J. Tang, S. Lao, "Review of unmanned aerial vehicle swarm communication

architectures and routing protocols," *Applied Sciences*, 10(10), 3661, 2020. doi: https://doi.org/10.3390/app10103661
30. T. Ulutaş, O. Avcı, E.C. Akar, B. Köksal, Y. Kalkan, "Simple design and implementation of two-way communication system through UAV," *Balkan Journal of Electrical and Computer Engineering*, vol. 11, issue 1, pp. 61–70, 2023. doi: https://doi.org/10.17694/bajece.1115408
31. H. Lai et al., "Optimization of full duplex UAV secure communication with the aid of RIS," *Drones*, 7(9), 591, 2023. doi: https://doi.org/10.3390/ drones7090591
32. L. García Rodríguez, L. Castro-Santos, M.I. Lamas Galdo, "Feasibility and limitations of solar energy integration in merchant ships: A case study on fire detection systems," *Journal of Marine Science and Engineering*, 13(5), 991, 2025. doi: https://doi.org/10.3390/jmse13050991
33. S.K. Khorasani, B.S. Ghahfarokhi, N. Movahhedinia, "UAV-assisted small base station ON-OFF switching in 6G cellular networks considering backhaul energy consumption," *Computer Communications*, 241, 108253, 2025. doi: https://doi.org/10.1016/j.comcom.2025.108253
34. G. Kou et al., "Intelligent UAV swarm key agreement survey: Systematic taxonomy, cryptographic automaton and quantum resistance," *Internet of Things*, vol. 34, 101720, 2025. doi: https://doi.org/10.1016/j.iot.2025.101720
35. S. Aggarwal, I. Budhiraja, S. Garg, G. Kaddoum, B.J. Choi, M.S. Hossain, "A blockchain-based secure path planning in UAVs communication network," *Alexandria Engineering Journal*, vol. 113, pp. 451–460, 2025. doi: https://doi.org/10.1016/j.aej.2024.10.078
36. S. Zinchenko, V. Kobets, O. Tovstokoryi, P. Nosov, I. Popovych, "Intelligent System Control of the Vessel Executive Devices Redundant Structure," in *CEUR Workshop Proceedings*, vol. 3403, paper 44, 2023. Available: https://ceur-ws.org/Vol-3403/paper44.pdf
37. V. Kobets, I. Popovych, S. Zinchenko, P. Nosov, O. Tovstokoryi, K. Kyrychenko, "Control of the Pivot Point Position of a Conventional Single-Screw Vessel," in *CEUR Workshop Proceedings*, vol. 3513, paper 11, 2023. Available: https://ceur-ws.org/Vol-3513/paper11.pdf
38. O. Fomin, A. Sulym, I. Kulbovskyi, P. Khozia, V. Ishchenko, "Determining rational parameters of the capacitive energy storage system for the underground railway rolling stock," *Eastern-European Journal of Enterprise Technologies*, vol. 2, no. 1 (92), pp. 63–71, 2018. doi: https://doi.org/10.15587/1729-4061.2018.126080
39. A.O. Sulym, O.V. Fomin, P.O. Khozia, A.G. Mastepan, "Theoretical and practical determination of parameters of on-board capacitive energy storage of the rolling stock," *Naukovyi Visnyk Natsionalnoho Hirnychoho Universytetu*, no. 5, pp. 79–87, 2018. doi: https://doi.org/10.29202/nvngu/2018-5/8
40. O. Melnyk, S. Kuznichenko, O. Onishchenko, "Impact of AIS manipulation on shipping safety and strategic countermeasures," *Lex Portus*, 10(4), pp. 31–39, 2024. doi: https://doi.org/10.62821/lp10403
41. V.V. Yakovenko, N.I. Furmanova, I.M. Flys, O.Y. Malyi, O.Y. Farafonov, H.V. Moroz, "Determination of the generalized optimality criteria for selecting civilian shelter facilities from attacks by ballistic (cruise) missiles and kamikaze drones in urbanized areas," *System Research and Information Technologies*, no. 3, pp. 25–43, 2024. doi: https://doi.org/10.20535/SRIT.2308-8893.2024.3.02
42. A.A. Gurskiy, A.V. Denisenko, A.E. Goncharenko, "Expansion of the mathematical apparatus of discrete-continuous networks for the automation of their synthesis procedures," *System Research and Information Technologies*, no. 2, pp. 93–99. doi: https://doi.org/10.20535/SRIT.2308-8893.2024.2.07
43. S.V. Melnykov, P.M. Malezhyk, A.S. Gasanov, P.I. Bidyuk, "Methodological aspects of operative control system intellectualization for dynamic objects,"

System Research and Information Technologies, no. 4, pp. 44–57, 2022. doi: https://doi.org/10.20535/SRIT.2308-8893.2022.4.04

44. Y. Koskina, S. Onyshenko, O. Drozhzhyn, O. Melnyk, "Efficiency of tramp fleet operating under the contracts of affreightment," *Scientific Journal of Silesian University of Technology. Series Transport*, vol. 120, pp. 137–149, 2023. doi: https://doi.org/10.20858/sjsutst.2023.120.9
45. I. Burmaka, I. Vorokhobin, O. Melnyk, O. Burmaka, S. Sagin, "Method of prompt evasive maneuver selection to alter ship's course or speed," *Transactions on Maritime Science*, vol. 11, no. 1, pp. 1–9, 2022. doi: https://doi.org/10.7225/toms.v11.n01.w01
46. A. Sotnikov, A. Tanciyra, O. Lavrov, "Calculating method of error calculations of the object coordination by means of conducting platform free inertial navigation systems of an unmanned aerial vehicle," *Advanced Information Systems*, vol. 2, no. 1, pp. 105–110, 2018. doi: https://doi.org/10.20998/2522-9052.2018.1.20
47. A. Podorozhniak, Y. Volotskov, O. Shevtsova, "Drone's Control System Research," *Advanced Information Systems*, vol. 2, no. 3, pp. 97–101, 2018. doi: https://doi.org/10.20998/2522-9052.2018.3.16
48. D. Voloshyn, V. Brechko, S. Semenov, "Method of an unmanned aerial vehicle composition route in space," *Advanced Information Systems*, vol. 5, no. 4, pp. 26–33, 2021. doi: https://doi.org/10.20998/2522-9052.2021.4.04
49. O.M. Melnyk et al., "Enhancing shipboard technical facility performance through the utilization of low-sulfur marine fuel grades," *Journal of Chemistry and Technologies*, vol. 32, no. 1, pp. 233–245, 2024. doi: https://doi.org/10.15421/jchemtech.v32i1.297916

Received 07.12.2024

INFORMATION ON THE ARTICLE

Sergiy V. Kurdiuk, ORCID: 0000-0002-3165-4571, National University "Odesa Maritime Academy", Ukraine, e-mail: s.kurd@ukr.net

Oleksiy M. Melnyk, ORCID: 0000-0001-9228-8459, Odesa National Maritime University, Ukraine, e-mail: m.onmu@ukr.net

Oleg A. Onishchenko, ORCID: 0000-0002-3766-3188, National University "Odesa Maritime Academy", Ukraine, e-mail: oleganaton@gmail.com

Sergiy M. Volianskiy, ORCID: 0000-0001-7922-0441, Odesa National Maritime University, Ukraine, e-mail: vffogres@gmail.com

Valerii A. Shevchenko, ORCID: 0000-0003-3229-1909, National University "Odesa Maritime Academy", Ukraine, e-mail: shevchenko@onma.edu.ua

Bogdan M. Alieksieichuk, ORCID: 0000-0003-1043-5174, National University "Odesa Maritime Academy", Ukraine, e-mail: b.alieksieichuk@gmail.com

ПРАКТИЧНІ АСПЕКТИ СТВОРЕННЯ СИСТЕМИ ПЕРЕДАВАННЯ ДАНИХ ДЛЯ КЕРУВАННЯ БЕЗПІЛОТНИМИ НАДВОДНИМИ АПАРАТАМИ В УМОВАХ НЕСТАБІЛЬНИХ КАНАЛІВ ЗВ'ЯЗКУ/ С.В. Курдюк, О.М. Мельник, О.А. Онищенко, С.М. Волянський, В.А. Шевченко, Б.М. Алєксєйчук

Анотація. Представлено розроблення й верифікацію адаптивної системи передавання даних для керування безекіпажними надводними апаратами (USV) в умовах нестабільних каналів зв'язку. Працю спрямовано на подолання обмежень наявних технологій, зокрема мереж LTE та супутникових систем, які не завжди забезпечують стабільну якість сервісу під час дистанційного керування USV. Запропоновано адаптивний алгоритм маршрутизації, що здійснює дина-

мічне оцінювання стану каналів зв'язку за ключовими показниками: затримкою, втратою пакетів та доступністю — і визначає оптимальні канали з урахуванням змінних вагових коефіцієнтів. Експериментальні результати підтвердили суттєве скорочення затримок передавання даних, стабільну трансляцію відео в реальному часі із затримкою 1–4 секунди та зниження втрат пакетів до рівня нижче 2%. Крім того, у системі реалізовано використання сучасних стандартів відеокодування (наприклад, H.265) та захищених VPN-каналів, що підвищує ефективність використання пропускної здатності та рівень кіберзахисту. Отримано результати, що підтверджують практичну придатність запропонованої системи для експлуатації USV у реальних морських умовах, а також її потенціал для застосування у критично важливих сценаріях, які потребують стійкого зв'язку з низькою затримкою.

Ключові слова: адаптивне передавання даних, безпілотні апарати, керування, маневрування, безпека навігації, канали зв'язку, управління процесами, алгоритм маршрутизації, оптимізація затримок, зменшення втрат, пакети даних, потокове відео, операційна ефективність, моніторинг стану, стиснення H.265, захищені VPN-тунелі, інтеграція 5G, моделі машинного навчання, прогнозування.

UDC 519.83
DOI: 10.20535/SRIT.2308-8893.2026.1.06

A PROBABILISTIC MODEL OF THE COLONEL BLOTTO GAME WITHOUT SYMMETRY AND HOMOGENEITY CONSTRAINTS

S.A. SMIRNOV, I.M. TERESHCHENKO

Abstract. The classic Colonel Blotto game for two players was considered. The probabilistic model of the payoff functions of the specified problem was investigated, and the game conditions are not subject to the restrictions of symmetry and homogeneity. The system of equations obtained using the method of Lagrange multipliers has a large dimension. In order to find a solution, a way to reduce the dimension was found. The found ratio between the resources of both players, distributed over the courts, made it possible to identify a parameter determined by the ratio of Lagrange multipliers from the corresponding functions for both players. For such a parameter, an interval constraint that it satisfies was found, and an equation is formulated to find it, which is solved numerically. The found value of the parameter makes it possible to calculate individual Lagrange multipliers and obtain the optimal distribution of players' resources in the form of a Nash equilibrium in pure game strategies. An example of a game under significantly different conditions for players was studied.

Keywords: conflict confrontation, optimal allocation of resources, two-person game, Colonel Blotto's game, probabilistic payoff model, Nash equilibrium, efficiency of resource use, 1-parametrization.

INTRODUCTION

Colonel Blotto's problem as a model of competitive struggle between two players on several platforms has long become a classic. It was first presented by Borel [1] in 1921, but despite its century-old history of applications, it still remains relevant and attracts the attention of researchers [2]. As a model of competitive struggle, it finds numerous applications in the social, economic spheres of activity, etc.

The game is described as follows. Two players compete against each other on several sites, which can be battlefields, election areas, product markets, etc. Each participant has a limited resource that must be distributed across the sites in such a way as to maximize their winnings, taking into account that each site has its own value for each of the rivals. In the auction model, the winner of a given site is the player who has allocated more resources to it than the opponent. In the case of equal allocated resources, a draw occurs.

More interesting for us is the case of the probabilistic model. Here, the probability of winning on a certain site is directly proportional to the resource allocated to it and inversely proportional to the sum of the resources allocated to it by both players. It should be noted that the following constraints are usually imposed on the game: symmetry, when the total resources of each player are the same, and homogeneity, when the worth of victory for any site coincides with that of its opponent. Thus, when searching for equilibrium in Colonel Blotto's problem, researchers limited themselves to symmetric or homogeneous cases, with the latter condition being used quite often [3–5].

Hence, the case where these two conditions are not met is of not only scientific interest. When trying to find equilibrium strategies for such a problem, it becomes necessary to solve systems of equations of high dimension. Using the approach proposed in [6], the system of equations is reduced to the one-dimensional case and represents by a single variable equation. In this case, we arrive at the precize solution of the asymmetric Colonel Blotto game without homogeneity constraints.

PROBLEM STATEMENT

Consider the probabilistic model of Colonel Blotto's game. Two opposing parties distribute their resources x_i and y_i, $i=\overline{1,n}$, across n sites. The resource constraints are determined by the following inequalities, $\sum_{i=1}^{n} x_i \le R_x$, $\sum_{i=1}^{n} y_i \le R_y$. The probability of the first player winning on the i-th court is given as follows:

$$p_i^x(x_i, y_i) = \frac{\alpha_i x_i^{r_i}}{\alpha_i x_i^{r_i} + \beta_i y_i^{r_i}},$$

where $r_i \in (0,1]$, and $\alpha_i > 0$, $\beta_i > 0$ are the efficiency coefficients of resource use on the corresponding sites for the first and second players. Similarly, the formula for the probability of winning $p_i^y(x_i, y_i)$ for the second player is:

$$p_i^y(x_i, y_i) = \frac{\beta_i y_i^{r_i}}{\alpha_i x_i^{r_i} + \beta_i y_i^{r_i}}.$$

In this case, the payoff functions take the form:

$$F_x(x,y) = \sum_{i=1}^{n} X_i p_i^x(x_i, y_i),$$

$$F_y(x,y) = \sum_{i=1}^{n} Y_i p_i^y(x_i, y_i),$$

where X_i, Y_i is the value of winning on the i-th court for each of the two players. A Nash equilibrium in pure strategies (x^*, y^*) is a pair of vectors that satisfies for any (x, y) satisfies: $F_x(x^*, y^*) \ge F_x(x^*, y)$, $F_y(x^*, y^*) \ge F_y(x, y^*)$.

Asymmetric game when $R_x \neq R_y$, with the following parameter values $r_i = 1$, $\alpha_i = 1$, $i \in N$ was considered in [3] for the case $X_i = Y_i = const$. The only Nash equilibrium in this case is the use of such pure strategies, when resources must be evenly distributed between sites. A continuation of these studies was the work [4], where an equilibrium in pure strategies was found for the case $X_i = Y_i = V_i$ with arbitrary $r_i \in (0,1]$, $\alpha_i > 0$, $i \in N$. This paper considers the situation when $X_i \neq Y_i$.

The aim of the work is to find the Nash equilibrium in pure strategies for the asymmetric and heterogeneous Colonel Blotto game for the case of the probabilistic model.

PROCEDURE FOR CONSTRUCTING AN OPTIMAL SOLUTION

To find a solution to the problem, we write the Lagrange function for each of the players:

$$L_x = F_x + \lambda_x (R_z - \sum_{i=1}^{n} x_i),$$

$$L_y = F_y + \lambda_y (R_y - \sum_{i=1}^{n} y_i).$$

Then we get the system of equations:

$$\frac{\partial L_x}{\partial x_i} = \frac{\partial F_x}{\partial x_i} - \lambda_x = 0,$$

$$\frac{\partial L_y}{\partial y_i} = \frac{\partial F_y}{\partial y_i} - \lambda_y = 0,$$

$$\frac{\partial L_x}{\partial \lambda_x} = R_x - \sum_{i-1}^{n} x_i = 0,$$

$$\frac{\partial L_y}{\partial \lambda_y} = R_y - \sum_{i-1}^{n} y_i = 0.$$

Starting from the system, we get:

$$\frac{\partial L_x}{\partial x_i} = X_i \frac{\alpha_i \beta_i r_i x_i^{r_i - 1} y_i^{r_i}}{(\alpha_i x_i^{r_i} + \beta_i y_i^{r_i})^2} = \lambda_x,$$

$$\frac{\partial L_y}{\partial y_i} = Y_i \frac{\alpha_i \beta_i r_i x_i^{r_i} y_i^{r_i - 1}}{(\alpha_i x_i^{r_i} + \beta_i y_i^{r_i})^2} = \lambda_y.$$

We have $(2n+2)$ equations with $(2n+2)$ variables: $x_1, \ldots, x_n, y_1, \ldots, y_n, \lambda_x, \lambda_y$, that is, the system can be solved. Consider the ratio $\frac{\lambda_x}{\lambda_y}$. We will get:

$$\frac{\lambda_x}{\lambda_y} = X_i \frac{\alpha_i \beta_i r_i x_i^{r_i - 1} y_i^{r_i}}{(\alpha_i x_i^{r_i} + \beta_i y_i^{r_i})^2} \cdot \frac{1}{Y_i} \frac{(\alpha_i x_i^{r_i} + \beta_i y_i^{r_i})^2}{\alpha_i \beta_i r_i x_i^{r_i} y_i^{r_i - 1}} = \frac{X_i}{Y_i} \cdot \frac{y_i}{x_i}.$$

From here $y_i = \frac{\lambda_x}{\lambda_y} \cdot \frac{Y_i}{X_i} x_i$. Let's mark $\lambda = \frac{\lambda_x}{\lambda_y}$, $C_i = \frac{Y_i}{X_i}$. Then

$$\frac{y_i}{x_i} = \lambda C_i . \tag{1}$$

We will use expression (1) to go from the multidimensional case with $(2n+2)$ equations and $(2n+2)$ variables to the one-dimensional case. To do this, we will find an equation for searching λ. Consider the expressions $R_x\lambda_x$ and $R_y\lambda_y$:

$$R_x\lambda_x = \sum_{i=1}^{n} x_i\lambda_x = \sum_{i=1}^{n} x_i X_i \frac{\alpha_i\beta_i r_i x_i^{r_i-1} y_i^{r_i}}{(\alpha_i x_i^{r_i} + \beta_i y_i^{r_i})^2} =$$

$$= \sum_{i=1}^{n} X_i \frac{\alpha_i\beta_i r_i x_i^{r_i} y_i^{r_i}}{(\alpha_i x_i^{r_i} + \beta_i y_i^{r_i})^2} \cdot \frac{/(x_i^{r_i})^2}{/(x_i^{r_i})^2} = \sum_{i=1}^{n} X_i \frac{\alpha_i\beta_i r_i (\frac{y_i}{x_i})^{r_i}}{(\alpha_i + \beta_i (\frac{y_i}{x_i})^{r_i})^2} = \tag{2}$$

$$= \sum_{i=1}^{n} X_i \frac{\alpha_i\beta_i r_i (\lambda C_i)^{r_i}}{(\alpha_i + \beta_i (\lambda C_i)^{r_i})^2}.$$

Similarly,

$$R_y\lambda_y = \sum_{i=1}^{n} Y_i \frac{\alpha_i\beta_i r_i (\lambda C_i)^{r_i}}{(\alpha_i + \beta_i (\lambda C_i)^{r_i})^2} . \tag{3}$$

From formulas (2) and (3) we determine the relationship between $R_x\lambda_x$ and $R_y\lambda_y$:

$$\frac{R_x\lambda_x}{R_y\lambda_y} = \frac{R_x}{R_y}\lambda = \sum_{i=1}^{n} X_i \frac{\alpha_i\beta_i r_i (\lambda C_i)^{r_i}}{(\alpha_i + \beta_i (\lambda C_i)^{r_i})^2} / \sum_{i=1}^{n} Y_i \frac{\alpha_i\beta_i r_i (\lambda C_i)^{r_i}}{(\alpha_i + \beta_i (\lambda C_i)^{r_i})^2} =$$

$$= \sum_{i=1}^{n} X_i \frac{\alpha_i\beta_i r_i C_i^{r_i}}{(\alpha_i + \beta_i (\lambda C_i)^{r_i})^2} / \sum_{i=1}^{n} Y_i \frac{\alpha_i\beta_i r_i C_i^{r_i}}{(\alpha_i + \beta_i (\lambda C_i)^{r_i})^2} .$$

Let's rewrite this formula in the following form:

$$\lambda \frac{R_x}{R_y} \sum_{i=1}^{n} Y_i \frac{\frac{\alpha_i}{\beta_i} r_i C_i^{r_i}}{(\frac{\alpha_i}{\beta_i} + (\lambda C_i)^{r_i})^2} = \sum_{i=1}^{n} X_i \frac{\frac{\alpha_i}{\beta_i} r_i C_i^{r_i}}{(\frac{\alpha_i}{\beta_i} + (\lambda C_i)^{r_i})^2} . \tag{4}$$

Let's find the limits within which the value lies λ. From (1) we have $y_i = \lambda C_i x_i$. Then $\sum_{i=1}^{n} y_i = R_y = \lambda \sum_{i=1}^{n} C_i x_i$. Hence

$$\min_i C_i \cdot \sum_{i=1}^{n} x_i \le \sum_{i=1}^{n} C_i x_i \le \max_i C_i \cdot \sum_{i=1}^{n} x_i .$$

Since $R_x = \sum_{i=1}^{n} x_i$, and $\sum_{i=1}^{n} C_i x_i = \frac{R_y}{\lambda}$, then

$$R_x \min_i C_i \le \frac{R_y}{\lambda} \le R_x \max_i C_i,$$

$$\frac{R_y}{R_x}(\max_i C_i)^{-1} \le \lambda \le \frac{R_y}{R_x}(\min_i C_i)^{-1},$$

$$\frac{R_y}{R_x}\min_i \frac{X_i}{Y_i} \le \lambda \le \frac{R_y}{R_x}\max_i \frac{X_i}{Y_i}.$$

Since, except for λ, all other parameters of equation (4) are known, its solution can be found on the specified segment by an appropriate numerical method, for example, the dichotomy method.

Let's find the optimal values for x_i^*, y_i^*.

$$\lambda_x x_i = X_i \frac{\alpha_i \beta_i r_i x_i^{r_i - 1} y_i^{r_i}}{(\alpha_i x_i^{r_i} + \beta_i y_i^{r_i})^2} x_i = X_i \frac{\alpha_i \beta_i r_i x_i^{r_i} y_i^{r_i}}{(\alpha_i x_i^{r_i} + \beta_i y_i^{r_i})^2} \cdot \frac{/(\beta_i x_i^{r_i})^2}{/(\beta_i x_i^{r_i})^2} =$$

$$= X_i \frac{\frac{\alpha_i}{\beta_i} r_i (\frac{y_i}{x_i})^{r_i}}{(\frac{\alpha_i}{\beta_i} + (\frac{y_i}{x_i})^{r_i})^2} = X_i \frac{\frac{\alpha_i}{\beta_i} r_i (\lambda C_i)^{r_i}}{(\frac{\alpha_i}{\beta_i} + (\lambda C_i)^{r_i})^2} = a_i.$$

Similarly,

$$\lambda_y y_i = Y_i \frac{\frac{\alpha_i}{\beta_i} r_i (\lambda C_i)^{r_i}}{(\frac{\alpha_i}{\beta_i} + (\lambda C_i)^{r_i})^2} = b_i.$$

In the right-hand sides of the equalities, all parameters are known, therefore a_i and b_i take specific values. Now we express λ_x in terms of R_x and a_i, and λ_y in terms of R_y and b_i, after which we find x_i^* and y_i^*.

$$\lambda_x x_i = a_i \Rightarrow \sum_{i=1}^{n} \lambda_x x_i = \sum_{i=1}^{n} a_i \Rightarrow \lambda_x \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} a_i \Rightarrow$$

$$\lambda_x R_x = \sum_{i=1}^{n} a_i \Rightarrow \lambda_x^{-1} = \frac{R_x}{\sum_{i=1}^{n} a_i}.$$

$$x_i^* = \frac{a_i}{\lambda_x} = \frac{R_x a_i}{\sum_{i=1}^{n} a_i}. \qquad (5)$$

Similarly, for λ_y:

$$\lambda_y y_i = b_i \implies \lambda_y^{-1} = \frac{R_y}{\sum_{i=1}^{n} b_i}.$$

$$y_i^* = \frac{b_i}{\lambda_y} = \frac{R_y b_i}{\sum_{i=1}^{n} b_i}. \tag{6}$$

Formulas (5) and (6) express the optimal solution to the problem — the Nash equilibrium in pure strategies. Then the payoff functions take the following form:

$$F_x(x,y) = \sum_{i=1}^{n} X_i \frac{\alpha_i x_i^{r_i}}{\alpha_i x_i^{r_i} + \beta_i y_i^{r_i}} = \sum_{i=1}^{n} X_i \frac{\frac{\alpha_i}{\beta_i}}{\frac{\alpha_i}{\beta_i} + (\lambda C_i)^{r_i}},$$

$$F_y(x,y) = \sum_{i=1}^{n} Y_i \frac{\beta_i y_i^{r_i}}{\alpha_i x_i^{r_i} + \beta_i y_i^{r_i}} = \sum_{i=1}^{n} Y_i \frac{(\lambda C_i)^{r_i}}{\frac{\alpha_i}{\beta_i} + (\lambda C_i)^{r_i}}.$$

RESULTS OF THE NUMERICAL EXPERIMENT

We will conduct a numerical experiment based on theoretical calculations. We will consider Colonel Blotto's game on five platforms and also record the values of some parameters:

$R_x = 100$.
$r_1 = 0.1$; $r_2 = 0.25$; $r_3 = 0.5$; $r_4 = 0.75$; $r_5 = 1$.
$\alpha_1 = 1.4$; $\alpha_2 = 1.9$; $\alpha_3 = 3.1$; $\alpha_4 = 3.7$; $\alpha_5 = 4.1$.
$\beta_1 = 1.3$; $\beta_2 = 1.8$; $\beta_3 = 2.8$; $\beta_4 = 3.5$; $\beta_5 = 5$.

We will consider three cases, where $R_y = 150$, $R_y = 250$, $R_y = 350$. For each case, we will calculate two options.

The first case.

Table 1. Option No.1 ($R_y = 150$)

i	1	2	3	4	5
X_i / Y_i	1.1/1.5	1.1/2.1	1.1/3.4	1.1/4.2	1.1/4.9
x_i^* / y_i^*	5.01/2.96	12.56/10.38	24.22/32.42	29.02/47.98	29.16/56.24
p_i^x / p_i^y	0.53/0.46	0.52/0.47	0.48/0.51	0.42/0.57	0.29/0.7
F_x / F_y	2.44/9.22				

Calculated value $\lambda = 0.407$.

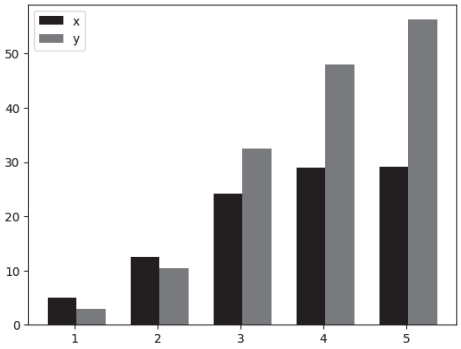


Fig. 1. Optimal solutions of players

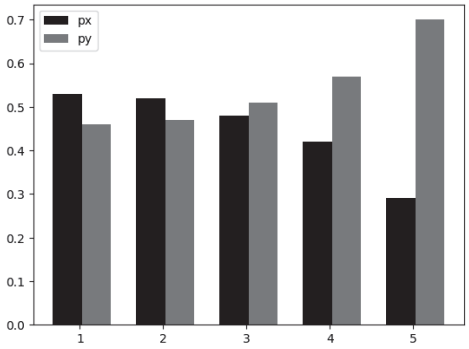


Fig. 2. Probabilities of players' winnings

Table 2. Option No. 2 ($R_y = 150$)

i	1	2	3	4	5
X_i / Y_i	5.5/1.5	5.5/2.1	5.5/3.4	5.5/4.2	5.5/4.9
x_i^* / y_i^*	3.92/2.19	9.18/7.19	18.69/23.68	29.11/45.57	39.07/71.35
p_i^x / p_i^y	0.53/0.46	0.52/0.47	0.49/0.5	0.43/0.56	0.3/0.69
F_x / F_y	12.47/9.1				

Calculated value $\lambda = 2.171$.

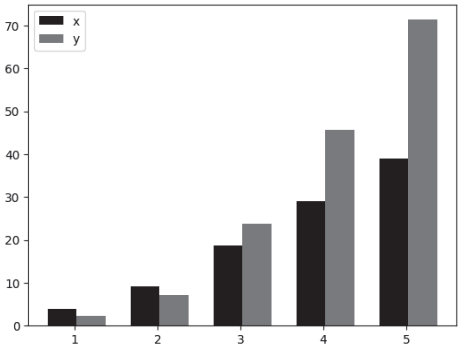


Fig. 3. Optimal solutions of players

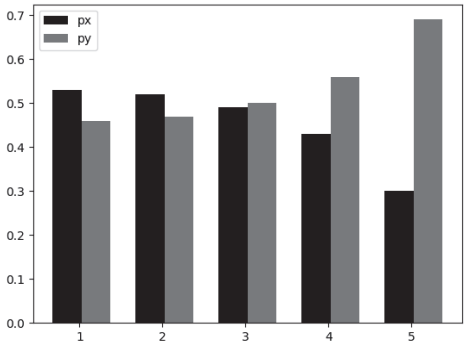


Fig. 4. Probabilities of players' winnings

The second case.

Table 3. Option No. 1 ($R_y = 250$)

i	1	2	3	4	5
X_i / Y_i	1.1/1.5	1.1/2.1	1.1/3.4	1.1/4.2	1.1/4.9
x_i^* / y_i^*	4.35/4.16	10.76/14.41	21.69/47.04	29.36/78.66	33.82/105.7
p_i^x / p_i^y	0.51/0.48	0.49/0.5	0.42/0.57	0.33/0.66	0.2/0.79
F_x / F_y	2.13/10.34				

Calculated value $\lambda = 0.684$.

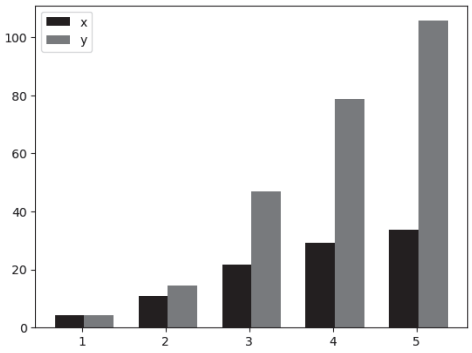


Fig. 5. Optimal solutions of players

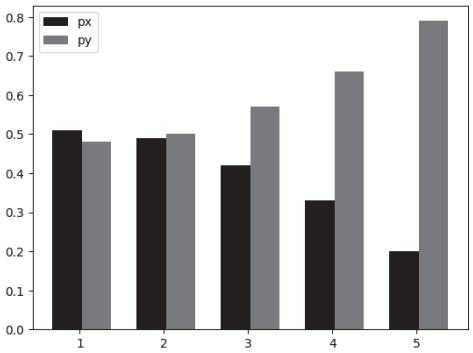


Fig. 6. Probabilities of players' winnings

Table 4. Option No. 2 ($R_y = 250$)

i	1	2	3	4	5
X_i / Y_i	5.5/1.5	5.5/2.1	5.5/3.4	5.5/4.2	5.5/4.9
x_i^* / y_i^*	4.28/4.01	9.64/12.67	18.84/40.07	29.14/76.55	38.07/116.67
p_i^x / p_i^y	0.52/0.47	0.49/0.5	0.43/0.56	0.33/0.66	0.21/0.78
F_x / F_y	10.87/10.24				

Calculated value $\lambda = 3.842$.

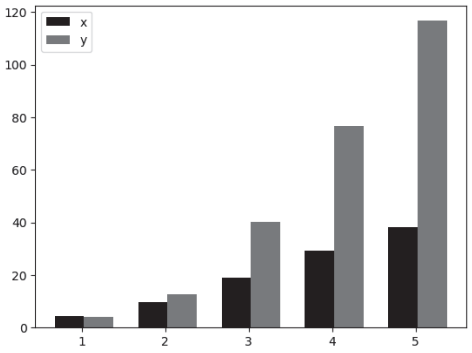


Fig. 7. Optimal solutions of players

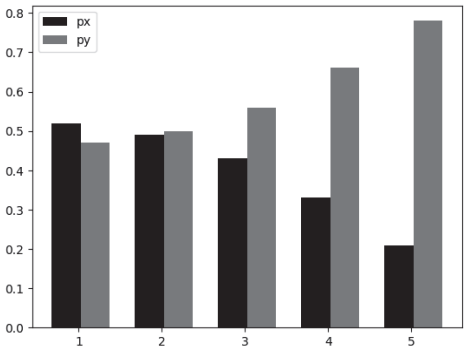


Fig. 8. Probabilities of players' winnings

The third case.

Table 5. Option No. 1 ($R_y = 350$)

i	1	2	3	4	5
X_i / Y_i	1.1/1.5	1.1/2.1	1.1/3.4	1.1/4.2	1.1/4.9
x_i^* / y_i^*	4.06/5.37	9.93/18.38	20.34/60.94	29.33/108.53	36.31/156.75
p_i^x / p_i^y	0.51/0.49	0.47/0.52	0.39/0.6	0.28/0.71	0.15/0.84
F_x / F_y	1.95/10.94				

Calculated value $\lambda = 0.967$.

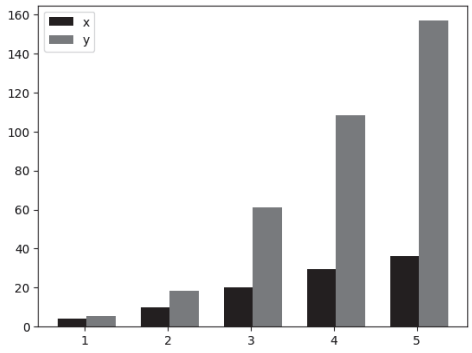


Fig. 9. Optimal solutions of players

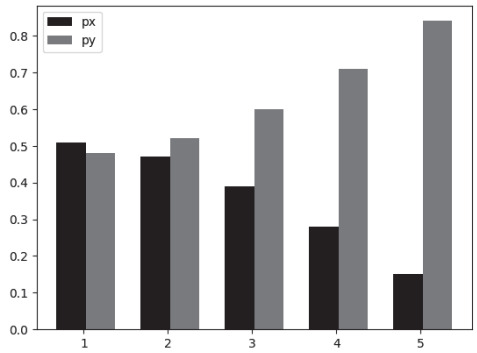


Fig. 10. Probabilities of players' winnings

Table 6. Option No. 2 ($R_y = 350$)

i	1	2	3	4	5
X_i / Y_i	5.5/1.5	5.5/2.1	5.5/3.4	5.5/4.2	5.5/4.9
x_i^* / y_i^*	4.77/6.34	10.42/19.4	19.5/58.78	29.19/108.67	36.1/156.78
p_i^x / p_i^y	0.51/0.48	0.47/0.52	0.38/0.61	0.28/0.71	0.15/0.84
F_x / F_y	9.83/10.97				

Calculated value $\lambda = 5.689$.

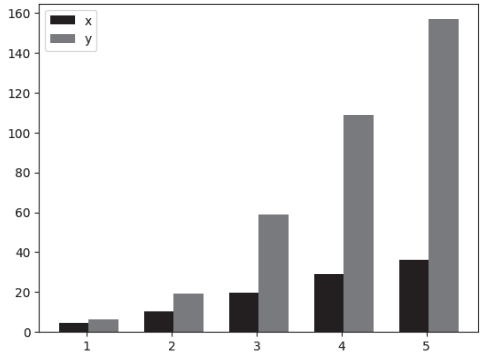


Fig. 11. Optimal solutions of players

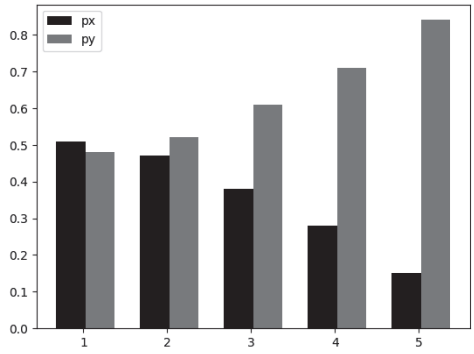


Fig. 12. Probabilities of players' winnings

Let us analyze the first case. From Table 1 it can be seen that the optimal values of the vector components x^* are greater than the corresponding vector components y^* at the first and second sites. At the same time, as follows from Fig. 1 and Fig. 2, despite the lack of advantage of the first player on the third court, the probability of his victory on the indicated court is almost equal to this value for the opponent. If the values of all the first player's sites increase fivefold, the overall picture presented in Table 2, Fig. 3 and Fig. 4 remains the same. Changes occur for the parameter λ.

Let's move on to the second case. The opponent increases his resources by 100. As can be seen from Table 3 and Fig. 5, the first player has a small advantage on the first site among the components of the optimal solution. A similar situation persists in the probability of winnings. However, as shown in Fig. 6, on the second court the probability of winning for the second player is slightly higher than that for the first. Increasing the value of each site fivefold, as follows from Table 4, Fig. 7 and Fig. 8, again does not change the situation as a whole, except for the value of the parameter λ. In general, the increase in the opponent's resources led to the loss of one site where the first player had previously won.

In the third case, we will again increase the opponent's resources by 100. As follows from Table 5, Fig. 9, Fig. 10 and Table 6, Fig. 11, Fig. 12, the components of the optimal vector of the first player are smaller than similar components of the second, but the probability of winning on the first and slightly smaller on the second platforms for the first player remains higher. Thus, the overall picture remains the same compared to the second case. It is clear that with a further increase in the opponent's resources, his winning probabilities will exceed those of the first player. Hence, the problem of the ratio of players' resources when the opponent's winning probabilities become larger on all platforms is of interest.

CONCLUSIONS

The case of a probabilistic model for two players is considered. A precise solution for the asymmetric Colonel Blotto game without the homogeneity constraint is found. The optimal allocation of players' resources is obtained in the form of a Nash equilibrium in pure strategies of the game.

A method for reducing the dimensionality of a system of equations obtained using the Lagrange multiplier method is proposed.

Of further interest is the study of solutions under conditions of incomplete information, that is, situations where the values of the coefficients are not precisely known.

REFERENCES

1. E. Borel, "La théorie du jeu les équations intégrales á noyau symétrique," *Comptes Rendus de l'Académie*, vol. 173. pp. 1304–1308, 1921.

2. Enric Boix-Adserà, Benjamin L. Edelman, Siddhartha Jayanti, *The Multiplayer Colonel Blotto Game*. doi: https://doi.org/10.48550/arXiv.2002.05240

3. L. Friedman, "Game-theory Models in the Allocation of Advertising Expenditure," *Operations Research*, vol. 6, pp. 699–709, 1958.

4. R.W. Robson, *Multi-Item Contest*; Working Paper No. 446. Australian National University, 2005, 27 p. Available: https://www.researchgate.net/publication/4980074_Multi-Item_Contests

5. B. Roberson, "The Colonel Blotto Game," *Economic Theory*, vol. 29, pp. 1–24, 2006. doi: https://doi.org/10.1007/s00199-005-0071-5

6. S. Smirnov, O. Glushchenko, K. Ilchuk, I. Makeenko, N. Oriekhova, "Assignments of factors levels for design of experiments with resource constraints," *Continuous and Distributed Systems. Theory and Applications. Ser. Solid Mechanics and Its Applications*, vol. 211. Springer, 2014. doi: https://doi.org/10.1007/978-3-319-03146-0_6

Received 26.04.2024

INFORMATION ON THE ARTICLE

Ivan M. Tereshchenko, ORCID: 0000-0003-0823-7507, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: ivan78ter@gmail.com

Sergey A. Smirnov, ORCID: 0000-0002-7233-5315, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: sergsmr@gmail.com

ЙМОВІРНІСНА МОДЕЛЬ ГРИ ПОЛКОВНИКА БЛОТТО БЕЗ ОБМЕЖЕНЬ СИМЕТРИЧНОСТІ ТА ОДНОРІДНОСТІ / С.А. Смирнов, І.М. Терещенко

Анотація. Розглянуто класичну гру полковника Блотто для двох гравців. Досліджено ймовірнісну модель вказаної задачі, причому на гру не накладаються обмеження симетричності та однорідності. З метою пошуку розв'язку знайдено спосіб пониження розмірності, оскільки одержана за допомогою методу множників Лагранжа система рівнянь має велику розмірність. Знайдене співвідношення між ресурсами обох гравців, що розподілені по майданчиках, дало змогу виділити параметр, який визначається співвідношенням множників Лагранжа з відповідних функцій для обох гравців. Для такого параметра знайдено інтервальне обмеження, яке він задовольняє, та для його пошуку сформульовано рівняння, яке розв'язується чисельно. Знайдене значення параметру дає можливість розрахувати окремі множники Лагранжа та отримати оптимальний розподіл ресурсів гравців у вигляді рівноваги Неша в чистих стратегіях гри. Досліджено приклад гри за суттєво відмінних умов для гравців.

Ключові слова: конфліктне протиборство, оптимальний розподіл ресурсів, гра двох осіб, гра полковника Блотто, ймовірнісна модель виграшу, рівновага Неша, ефективність використання ресурсів, 1-параметризація.

МАТЕМАТИЧНІ МЕТОДИ, МОДЕЛІ, ПРОБЛЕМИ І ТЕХНОЛОГІЇ ДОСЛІДЖЕННЯ СКЛАДНИХ СИСТЕМ

UDC 517.9:534.1
DOI: 10.20535/SRIT.2308-8893.2026.1.07

IMPLEMENTATION OF A GENERALIZED INTERMITTENCY SCENARIO IN THE RÖSSLER DYNAMICAL SYSTEM

O.O. HORCHAKOV, A.YU. SHVETS

Abstract. The realization of novel scenario involving transitions between different types of chaotic attractors is investigated for the Rössler system. Characteristic features indicative of the presence of generalized intermittency scenario in this system are identified. The properties of "chaos–chaos" transitions following the generalized intermittency scenario are analyzed in detail based on phase-parametric characteristics, Lyapunov characteristic exponents, phase portraits, and Poincaré sections.

Keywords: ideal dynamical system, regular and chaotic attractors, generalized intermittency scenario.

INTODUCTION

Scenarios of generalized intermittency describe the transition from a chaotic attractor of one type to a chaotic attractor of another type. Such scenarios were initially discovered in the study of non-ideal Sommerfeld–Kononenko-type dynamical systems [1, 2]. These scenarios generalize the Manneville–Pomeau scenarios [3, 4] and, in some cases, represent combinations of the Feigenbaum [5, 6] and Manneville–Pomeau scenarios. The review paper [7] presents implementations of various versions of the generalized intermittency scenario in non-ideal pendulum, hydrodynamic, and electroelastic systems.

Moreover, transitions of the "chaos–chaos" type following the generalized intermittency scenario have also been identified in non-isolated invariant sets, the so-called maximal attractors. Strictly speaking, these sets do not qualify as attractors in the classical sense. Nevertheless, even for such atypical attracting structures, the generalized intermittency scenario can still be observed [8, 9].

OBJECTIVE AND METHODOLOGY OF THE STUDY

It was established in [10, 11] that various types of the generalized intermittency scenario can be realized in the ideal Lorenz dynamical system. The objective of the present study is to provide numerical evidence supporting the realization of the generalized intermittency scenario in such classical dynamical system as the Rössler system. The investigation employs standard techniques of chaotic dynamics, including the Runge–Kutta method for constructing phase portraits of

attractors [12], the Benettin algorithm for computing the maximal Lyapunov exponent [13, 14], the Hénon method for constructing Poincaré sections [15], and a computational technique based on color-shaded encoding for visualizing the distribution of the invariant measure over the phase portrait of the attractor [16]. The detailed methodology for applying the above-mentioned numerical methods and algorithms is described in [16–18].

RÖSSLER SYSTEM

In [20], a nonlinear system of three differential equations was considered:

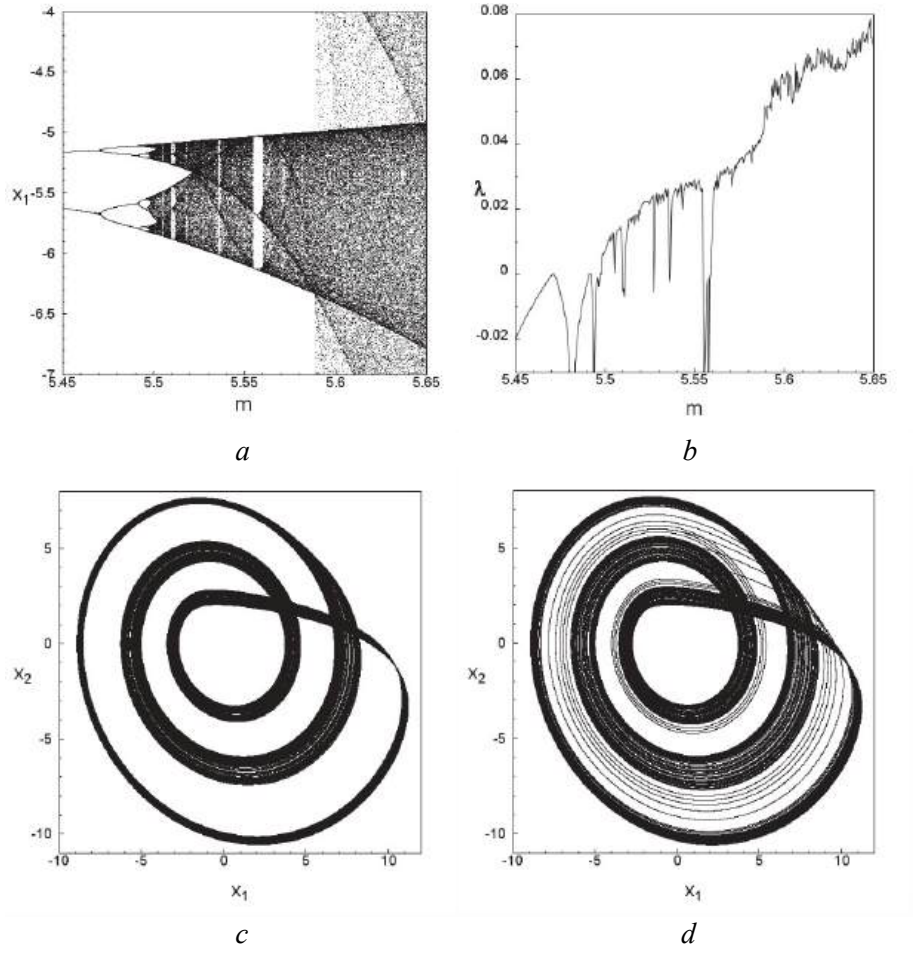
$$\begin{aligned} \dot{x}_1 &= x_2 - x_1; \\ \dot{x}_2 &= x_1 + e x_2; \\ \dot{x}_3 &= f + x_3(x_1 - m), \end{aligned} \tag{1}$$

Here x_1, x_2, x_3 are phase variables, and e, f, m are system parameters. This system later became known as the Rössler system. It should be noted that the first two equations of system (1) are linear, while the quadratic nonlinearity appears only in the third equation. Rössler proposed this system purely heuristically, without relying on any physical assumptions in its derivation. His goal was to construct a simple deterministic third-order system of differential equations exhibiting highly complex chaotic dynamics. Over time, Rössler revisited the analysis of system (1) in his later works [20–22]. Today, both the Rössler and Lorenz systems [23] are widely recognized as canonical examples of chaotic dynamics in low-dimensional deterministic systems.

Assume that the parameters of system (1) are $e = 0.2,\ f = 0.2$ and choose the parameter m as the bifurcation parameter. In Fig. 1, *a*, the phase–parameter characteristic of the system is shown, constructed using the Hénon method, as the parameter m varies within the interval $5.45 \le m \le 5.65$. Here the plane $x_2 = 0$ is chosen as the secant plane. Individual lines (branches) of the phase–parameter characteristic (the bifurcation tree) correspond to the limit cycles of system (1), while the densely black regions of the bifurcation tree correspond to the chaotic attractors of the system. An analysis of the phase–parameter characteristic shows that in the range $5.56 < m < 5.59$, system (1) undergoes transitions from limit cycles to chaotic attractors. These transitions occur via cascades of period-doubling bifurcations of limit cycles, that is, in full accordance with the Feigenbaum scenario [5, 6]. Once a chaotic attractor appears, it persists over a certain interval as the parameter *m* increases. When *m* reaches a certain critical value, the chaotic attractor disappears and a limit cycle again becomes the attractor of system (1). As *m* increases further, another transition from a regular regime to a chaotic one occurs according to the Feigenbaum scenario. It should be noted that short intervals of limit cycle existence are referred to as periodicity windows.

It should be noted that a positive maximal Lyapunov exponent is a necessary condition for the chaotic nature of a steady-state regime. Fig. 1, *b* shows the graph of the dependence of the maximal nonzero Lyapunov exponent λ_1 on the bifurcation parameter *m*. This graph was constructed using the algorithm proposed by Benettin et al. [13, 14]. Positive values of the Lyapunov exponent correspond to intervals of the parameter *m* for which chaotic attractors exist in system (1). The "drops" of the Lyapunov exponent graph into the region of negative values

correspond to the periodicity windows observed in Fig. 1, *a*. The most interesting region of the phase–parameter characteristic (Fig. 1, *a*) is the neighborhood of the point $m \approx 5.585$. As seen in Fig. 1, *a*, in the right-side neighborhood of $m \approx 5.585$, there is a significant increase in the area of the densely black region on the phase–parameter diagram. As established in [7, 11], such an increase in the corresponding area indicates the realization of the generalized intermittency scenario. Various versions of this scenario are described in [7–11]. Another indication of the realization of the generalized intermittency scenario is a noticeable increase in the maximum Lyapunov exponent at $m > 5.585$. We can see such increasement in Fig. 1, *b*.



a *b*

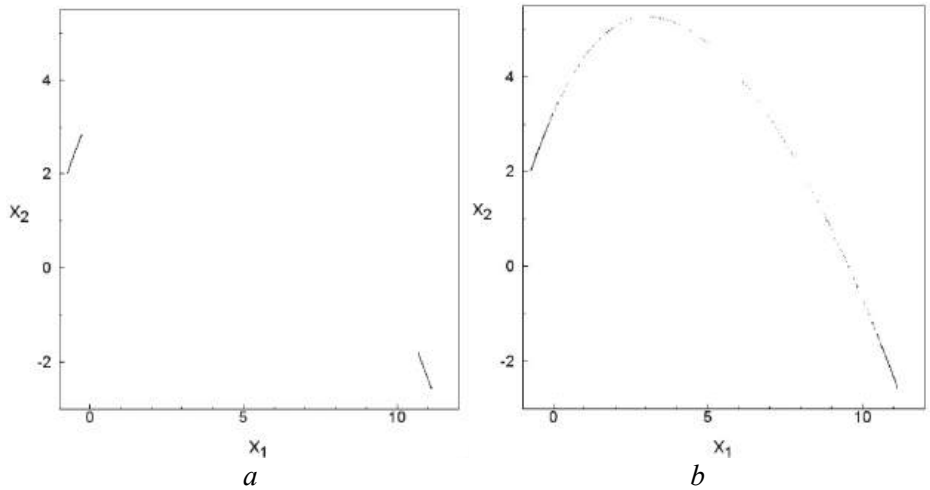
c *d*

Fig. 1. Phase-parametric characteristic – *a*; maximal non-zero Lyapunov exponent – *b*; distribution of the natural invariant measure at $m = 5.58$ – *c*; at $m = 5.59$ – *d*

Let us now examine in more detail the realization of the generalized intermittency scenario in the Rössler system by analyzing the distributions of natural invariant measures and Poincaré sections.

In Fig. 1, *c* is shown the distribution of the invariant measure over the phase portrait of the chaotic attractor at $m = 5.58$. As the parameter *m* increases, a hard bifurcation occurs in system (1), as a result of which the existing chaotic attractor disappears and a new type of chaotic attractor emerges. The distribution of the

invariant measure over the phase portrait of this new chaotic attractor, constructed at $\mathrm{m} = 5.59$, is shown in Fig. 1, *d*. The distributions of the invariant measure were constructed using the algorithm of computer encoding in shades of black [16, 17]. The trajectory motion along the new chaotic attractor exhibits phase alternation between two phase – a coarse-grain (rough) laminar phase and a turbulent phase. The coarse-grain laminar phase corresponds to chaotic wanderings of the trajectory in the region of localization of the disappeared chaotic attractor (dense black region in Fig. 1, *c*). At an unpredictable moment in time, the trajectory leaves the localization region of the vanished chaotic attractor and "escapes" to more distant areas of the phase space (gray points in Fig. 1, *d*). Such motions correspond to the turbulent phase of intermittency. Alternations between the coarse-grained laminar phase and the turbulent phase are observed an infinite number times. The transition time from one phase to another is also unpredictable. On average, the duration of the coarse-grained laminar phase exceeds that of the turbulent phase. This process fully corresponds to the scenario of generalized intermittency [7, 10, 11]. The scenario of generalized intermittency can also be identified by analyzing the Poincaré sections. In Fig. 2, the Poincaré sections of chaotic attractors at $m = 5.58$ and $m = 5.59$ are constructed using the Hénon method. Both sections exhibit a quasi-ribbon structure and represent chaotic sets of discrete points. It is worth noting that such a quasi-ribbon structure is characteristic of chaotic attractors in the Rössler system. As shown in Fig. 2, *b*, the structure of the Poincaré section at $m = 5.59$ contains all the fragments present in the Poincaré section of the chaotic attractor at $m = 5.58$ (Fig. 2, *a*).



a *b*

Fig. 2. Poincaré sections at $m = 5.58$ – *a*; at $m = 5.59$ – *b*

These fragments form the coarse-grained laminar phase of the attractor at $m = 5.59$. Accordingly, new points appear in the Poincaré section at $m = 5.59$, corresponding to the turbulent phase.

Let us now consider the bifurcations in the Rössler system as the parameter *m* varies within the interval $(16, 18.5)$. The values of the parameters *e* and *f* remain unchanged. As before, using the methods of Hénon, Benettin, and computer-based color coding, we construct a series of dynamic characteristics of the Rössler system. Thus, in Fig. 3, *a*, the phase–parameter characteristic of the Rössler system is

presented. The constructed bifurcation tree provides a clear representation of the types of attractors in system (1). The individual branches of the bifurcation tree correspond to limit cycles, while the densely black regions of the tree represent chaotic attractors. Moreover, this figure makes it possible to identify transition scenarios, including both "limit cycle-to-chaos" and "chaos-to-chaos" transitions.

The constructed bifurcation tree demonstrates a symmetry in the transitions to chaos, both with increasing and decreasing values of the parameter m. As m increases, starting from $m = 16.7$, in the system begins an infinite cascade of period-doubling bifurcations of limit cycles, followed by the emergence of a chaotic attractor with a relatively small localization region in the phase space. This represents a transition to chaos following the Feigenbaum scenario. A similar scenario is observed as m decreases, beginning from $m = 18.05$. Particular attention should also be paid to two bifurcation points: $m \approx 17.35$ and $m \approx 17.795$.

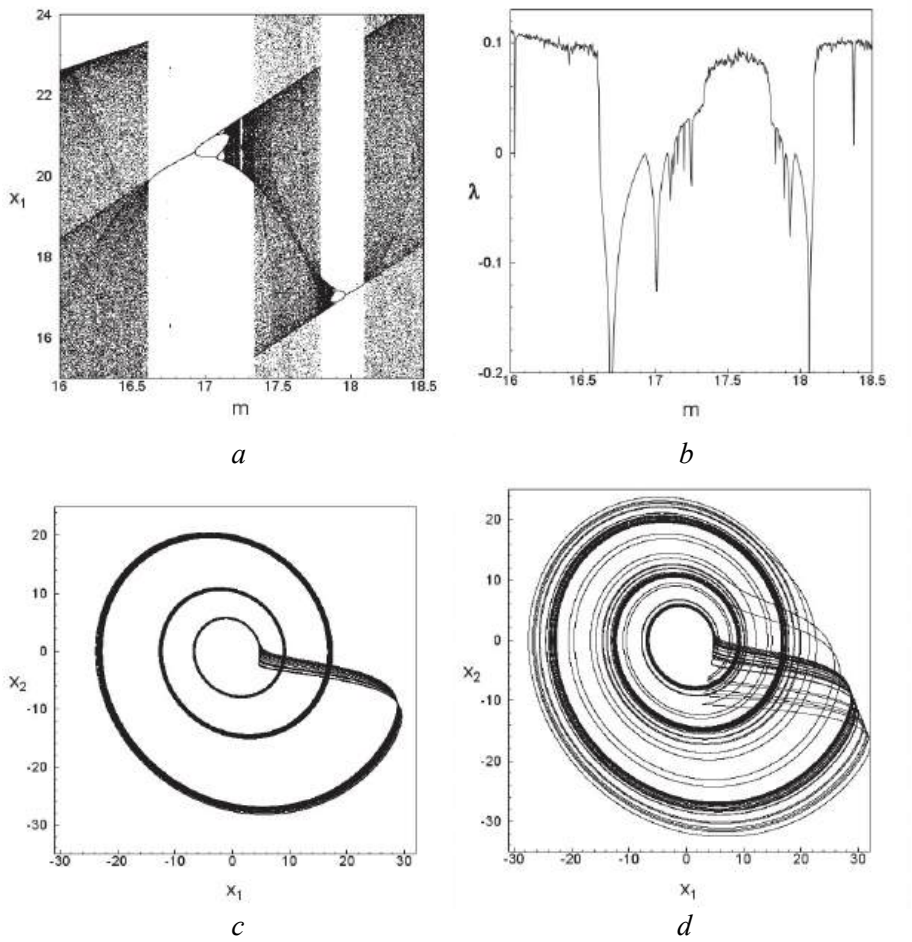


Fig. 3. Phase-parametric characteristic – *a*; Maximal non-zero Lyapunov exponent – *b*; and projections of distribution of the natural invariant measure at $m = 17.8$ – *c*; at $m = 17.79$ – *d*

In the right-hand neighborhood of $m \approx 17.35$ (and the left-hand neighborhood of $m \approx 17.795$), a significant increase in the area of the densely black chaotic region in Fig. 3, *a* is observed, indicating the realization of a generalized intermittency scenario of the transition from one type of chaotic attractor to another.

In addition, two bifurcation points are clearly visible at $m \approx 16.6$ and $m \approx 18.1$. As the system passes through these points, a "limit cycle–chaos" transition occurs following the Pomeau–Manneville scenario. In Fig. 3, *b*, the graph of the dependence of the maximal nonzero Lyapunov exponent on the bifurcation parameter *m* is presented. As seen from the graph, for $m > 17.35$ and $m < 17.795$, the value of the maximal Lyapunov exponent nearly doubles. This increase is further evidence of the realization of a generalized intermittency scenario in the Rössler system.

Finally, let us consider the realization of the generalized intermittency scenario through the phase portraits of chaotic attractors of different types.

In Fig. 3, *c*, the projection of the invariant measure distribution for the chaotic attractor at $m = 17.8$ is shown, while Fig. 3, *d* presents the projection of the invariant measure distribution for the chaotic attractor at $m = 17.79$. As the value of the parameter *m* decreases, the chaotic attractor that existed in the right-hand neighborhood of the bifurcation point $m = 17.795$ disappears, and for $m < 17.795$, a new type of chaotic attractor emerges. The motion of trajectories on this new attractor includes two phases, clearly identifiable in Fig. 3, *d*: a coarse-grain laminar phase and a turbulent phase. In the coarse-grain laminar phase (the densely black fragment in Fig. 3, *d*), the trajectory performs chaotic wandering in a neighborhood of the phase-space localization region of the attractor that existed for $m > 17.795$. The turbulent phase (the gray fragments in Fig. 3, *d*) corresponds to the trajectory's excursions into more distant regions of the phase space.

Similarly, the generalized intermittency scenario can be illustrated through Poincaré sections, as was done in Fig. 2. It should be noted that, in contrast to the previously analyzed case, the transition to chaos through the generalized intermittency scenario can occur both with increasing and decreasing values of the parameter *m*.

The implementation of the generalized intermittency scenario can also be observed in other regions of the parameter space of the Rössler system. Let us assume that $e = 0.2$ and $m = 17.4$, while the bifurcation parameter is chosen to be *f*.

We will investigate the dynamical behavior of system (1) within the range $0.2308 < f < 0.2311$. For these parameter values, the Rössler system has two coexisting attractors, each possessing its own basin of attraction. Fig. 4, a, b presents the phase–parameter characteristics of two different attractors constructed using the Hénon method.

As before, individual branches of the bifurcation trees correspond to limit cycles, while the densely black regions represent chaotic attractors. Despite a certain similarity between these phase–parameter characteristics, it is clearly seen—by examining the intervals of variation of the coordinate x_1—that the corresponding attractors are localized in different regions of the phase space.

Let us now focus exclusively on the realization of the generalized intermittency scenario. As noted earlier, an indicator of this scenario is a significant increase in the area of the densely black (chaotic) regions on the phase–parameter

characteristic. Such increases in the areas of the densely black regions can be observed on both phase–parameter characteristics. This indicates the possibility of a transition "chaotic attractor of one kind → chaotic attractor of another kind" according to the generalized intermittency scenario.

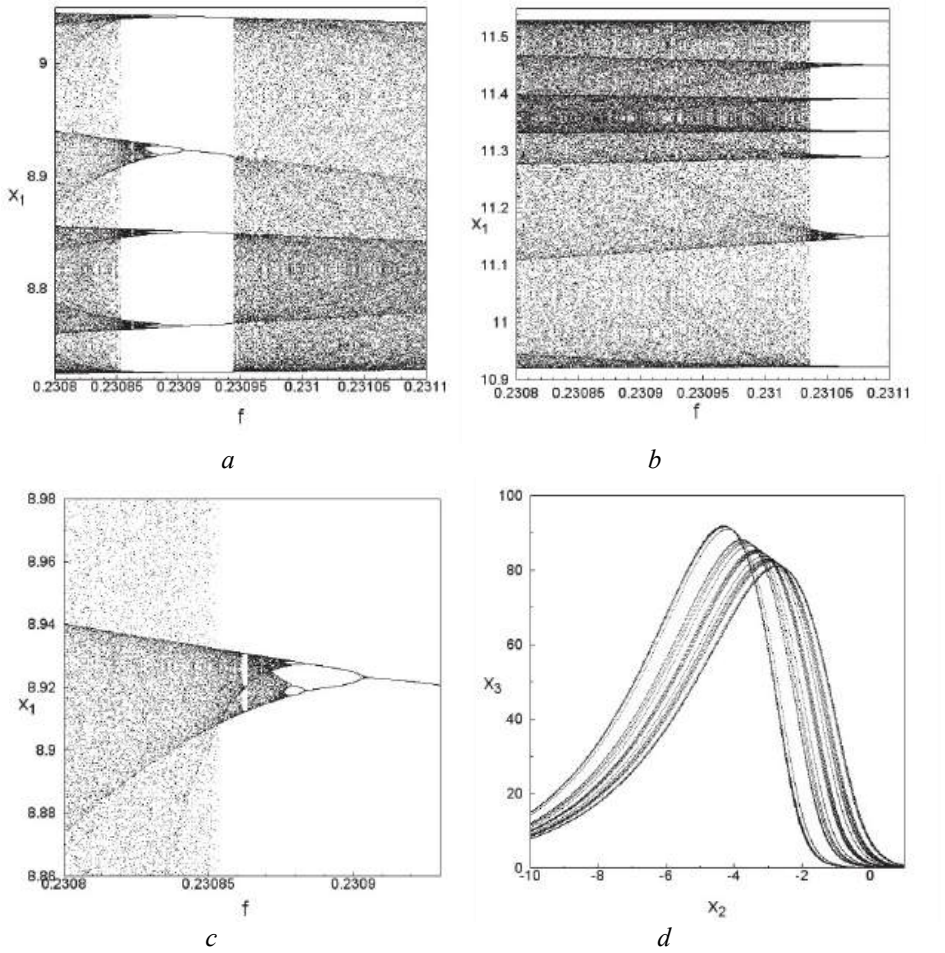


Fig. 4. Phase-parametric characteristics – *a*, *b*, *c*; fragment of distribution of invariant measure at $f = 0.23082$ – *d*

Let us examine this scenario in more detail using one of the coexisting attractors as an example. Fig. 4, *c* shows a fragment of the phase–parameter characteristic from Fig. 4, *a*. The enlarged scale in Fig. 4, *c* makes it possible to identify the bifurcation point $f = 0.23085$, at which a "chaos → chaos" transition occurs according to the generalized intermittency scenario. In Fig. 4, *d* is shown an enlarged fragment of the distribution of the invariant measure over the phase portrait of the attractor at $f = 0.23082$. This attractor appears as the parameter f decreases immediately after the bifurcation point $f \approx 0.23085$. The use of the enlarged scale makes it possible to clearly visualize the features of this distribution. One can distinguish a coarse-grain laminar phase of the trajectory (the densely black region in the figure) and a turbulent phase (the gray-shaded areas).

Let us emphasize once again that the coarse-grain laminar phase almost coincides with the region of localization in phase space of the chaotic attractor that

exists for $f > 0.23085$ and disappears after the bifurcation point is passed. Another confirmation of the generalized intermittency scenario is a noticeable increase in the value of the maximal Lyapunov exponent. Specifically, for the chaotic attractor at $f = 0.23086$, the maximal Lyapunov exponent is $\lambda_1 = 0.005$, while for the chaotic attractor at $f = 0.23082$, it increases to $\lambda_1 = 0.010$.

CONCLUSIONS

Thus, the generalized intermittency scenario, previously identified in non-ideal dynamical systems, is also realized in ideal dynamical systems such as the classical ideal Rössler system. Future research will focus on identifying the realization of other types of the generalized intermittency scenario in various ideal dynamical systems.

ACKNOWLEDGMENTS

This work was supported by a grant from the Simons Foundation International (SFI-PD-Ukraine-00014586, O.O. Horchakov).

REFERENCES

1. A. Sommerfeld, “Beitrage zum dynamischen Ausbau der Festigkeitslehre,” *Physikalische Zeitschrift*, 3, pp. 266–271, 1902.
2. V.O. Kononenko, *Vibrating system with a limited power-supply*. London: Iliffe,1969, 236 p.
3. P. Manneville, Y. Pomeau, “Different ways to turbulence in dissipative dynamical systems,” *Physica D. Nonlinear Phenom.*, vol. 1, issue 2, pp. 219–226, 1980. doi: https://doi.org/10.1016/0167-2789(80)90013-5
4. Y. Pomeau, P. Manneville, “Intermittent transition to turbulence in dissipative dynamical systems,” *Comm. Math. Phys.*, vol. 74, pp. 189–197, 1980. doi: https://doi.org/10.1007/BF01197757
5. M.J. Feigenbaum, “Quantitative universality for a class of nonlinear transformations,” *J. Stat. Phys.*, vol. 19, pp. 25–52, 1978. doi: https://doi.org/10.1007/BF01020332
6. M.J. Feigenbaum, “The universal metric properties of nonlinear transformations,” *J. Stat. Phys.*, vol. 2, pp. 669–706, 1979. doi: https://doi.org/10.1007/BF01107909
7. A. Shvets, “Overview of Scenarios of Transition to Chaos in Nonideal Dynamic Systems,” *13th Chaotic Modeling and Simulation International Conference. CHAOS 2020. Springer Proceedings in Complexity*, pp. 853–864, Springer, Cham, 2021. doi: https://doi.org/10.1007/978-3-030-70795-8_59
8. A. Shvets, S. Donetskyi, “New Types of Limit Sets in the Dynamic System “Spherical Pendulum—Electric Motor,” *Nonlinear Mechanics of Complex Structures. Advanced Structured Materials*, vol. 157, pp. 443–455, 2021. doi: https://doi.org/10.1007/978-3-030-75890-5_25
9. S.V. Donetskyi, A.Yu. Svets, “Generalization of the concept of attractor for pendulum systems with limited excitation,” *J. Math. Sci*., vol. 273, pp. 220–229, 2023. doi: https://doi.org/10.1007/s10958-023-06550-7
10. O. Horchakov, A. Shvets, “Generalized scenarios of transition to chaos in ideal dynamic systems,” *System Research and Information Technologies*, no. 3, pp. 64–73, 2024. doi: https://doi.org/10.20535/srit.2308-8893.2024.3.04
11. A. Shvets, “Generalised Intermittency in Non-ideal and “Classical” Dynamical Systems,” *Analytical and Approximate Methods for Complex Dynamical Systems.*

Understanding Complex Systems, pp. 75–87, Springer, Cham, 2025. doi: https://doi.org/10.1007/978-3-031-77378-5_5
12. E. Hairer, S.P. Nørsett, G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*. Berlin, Springer-Verlag, 1987.
13. G. Benettin, L. Galgani, J.-M. Strelcyn, "Kolmogorov entropy and numerical experiments," *Phys. Rev. A*, 14, 2338, 1976. doi: https://doi.org/10.1103/PHYSREVA.14.2338
14. G. Benettin, L. Galgani, A. Giorgilli, J.-M. Strelcyn, "Lyapunov Characterictic Exponents for smooth dynamical systems and for hamiltonian systems; A method for computing all of them. Part 2: Numerical application," *Meccanica*, vol. 15, pp. 21–30, 1980. doi: https://doi.org/10.1007/BF02128237
15. *Handbook of Applications of Cha*os Theory; Edited by C.H. Skiadas, Char. Skiadas. Chapman and Hall/CRC, 2016, 952 p. doi: https://doi.org/10.1201/b20232
16. M. Hénon, "A two-dimensional mapping with a strange attractor," *Communications in Mathematical Physics*, vol. 50**,** pp. 69–77, 1976. doi: https://doi.org/10.1007/BF01608556
17. S.P. Kuznetsov, Dynamic Chaos. M.: Fizmatlit, 2006, 292 p.
18. T.S. Krasnopolskaya, A.Yu. Shvets, "Properties of chaotic oscillations of the liquid in cylindrical tanks," *Prikl. Mekh*., 28(6), pp. 52–61, 1992.
19. T.S. Krasnopol'skaya, A.Y. Shvets, "Parametric resonance in the system: Liquid in tanks + electric motor," *Int. Appl. Mech*., vol. 29, pp. 722–730, 1993. doi: https://doi.org/10.1007/BF00847371
20. O.E. Rössler, "An Equation for Continuous Chaos," *Physics Letters A*, vol. 57, issue 5, pp. 397–398, 1976. doi: https://doi.org/10.1016/0375-9601(76)90101-8
21. O.E. Rössler, "Continuous Chaos – Four Prototype Equations," *Annals of the New York Academy of Sciences*, 316, pp. 376–392, 1979. doi: https://doi.org/10.1111/j.1749-6632.1979.tb29482.x
22. O.E. Rössler, C. Letellier, "The Phenomenon of Chaos," *Chaos. Understanding Complex Systems.* Springer, Cham, 2020. doi: https://doi.org/10.1007/978-3-030-44305-4_1
23. E.N. Lorenz, "Deterministic nonperiodic flow," *J. Atmos. Sci.*, vol. 20, issue 2, pp. 130–141, 1963. doi: https://doi.org/10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2

Received 19.06.2025

INFORMATION ON THE ARTICLE

Aleksandr Yu. Shvets, ORCID: 0000-0003-0330-5136, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, email: aleksandrshvetskpi@gmail.com

Oleksii O. Horchakov, ORCID: 0009-0006-3664-8812, Institute of Mathematics NAS of Ukraine, Ukraine, e-mail: o.horchakov@imath.kiev.ua

РЕАЛІЗАЦІЯ СЦЕНАРІЮ УЗАГАЛЬНЕНОЇ ПЕРЕМІЖНОСТІ В ДИНАМІЧНІЙ СИСТЕМІ РЕССЛЕРА / О.О. Горчаков, О.Ю. Швець

Анотація. Досліджено реалізацію нового сценарію переходу між різними типами хаотичних атракторів для системи Ресслера. Виявлено характерні ознаки, що вказують на наявність сценарію узагальненої переміжності в цій системі. Властивості переходів типу «хаос–хаос» за сценарієм узагальненої переміжності детально проаналізовано на основі фазо-параметричних характеристик, ляпуновських характеристичних показників, фазових портретів і перерізів Пуанкаре.

Ключові слова: ідеальна динамічна система, регулярний і хаотичний атрактори, сценарій узагальненої переміжності.

UDC 004.855.5:519.237.8
DOI: 10.20535/SRIT.2308-8893.2026.1.08

IMPROVING THE SOM ALGORITHM TO ENSURE STABILITY AND REPRODUCIBILITY OF DATA CLUSTERING RESULTS

O.V. IVASHCHENKO, S.S. FEDIN

Abstract. The article proposes a method to improve the Kohonen Self-Organizing Map (SOM) learning algorithm to ensure the stability and reproducibility of clustering results, an urgent task when working with large amounts of data. SOM is widely used in clustering and visualization tasks, especially in applications that require analyzing multidimensional data structures, such as telecommunications billing systems and financial analysis. The standard SOM implementation, which includes random weight initialization and stochastic sample selection during training, leads to significant cluster variability even when using the same input data and identical network training parameters. This makes it difficult to apply this algorithm in cases where stability and reproducibility of results are required. To solve this problem, we propose modifying the algorithm to include its own random number generator and introducing a seed parameter to fix the initial training conditions. This reduces variability and ensures reproducible clustering results, thereby increasing the reliability of the analysis and the suitability of the SOM algorithm for real business tasks. The proposed method has been tested on data from billing systems, where the reproducibility of clustering results is critical for effective work with customer segments, the development of targeted marketing strategies, and the creation of personalized tariff plans.

Keywords: Kohonen self-organizing maps (SOM), data clustering, seed parameter, reproducibility of results, random number generator.

INTRODUCTION

In today's environment, telecommunications companies process large amounts of data on a daily basis that contain valuable information about subscriber behavior and service usage. This data plays an important role in making strategic decisions, developing personalized offers, and increasing the competitiveness of companies. One of the key tasks is to apply clustering to organize and analyze the data, allowing to draw useful analytical conclusions for further use.

Among modern approaches to clustering and visualization of multidimensional data, Kohonen's Self-Organizing Maps (SOM) [1] occupy a special place, preserving the topological structure of the data and allowing an intuitive understanding of its grouping through visualization. At the same time, algorithmic features such as random initialization of weights and stochastic samples selection during training lead to the fact that the results can differ significantly for the same input data. This creates difficulties in cases where stability and reproducibility of

clustering results are required, which is important for making informed decisions in business and analytics.

This opens up the possibility of improving the method to increase the reproducibility and accuracy of clustering, which is important for scientists and analysts using this tool.

LITERATURE ANALYSIS AND PROBLEM STATEMENT

Kohonen's Self-Organizing Map is a popular method for data analysis and clustering that allows to identify similar groups of objects in a data set and simplify their structure for further use. However, despite the widespread use of SOM in industries such as telecommunications, finance, and engineering, the problem of stability and reproducibility of clustering results remains unresolved. This limits the practical application of the algorithm in tasks requiring accurate and stable data grouping.

Researchers have already proposed some improvements to the SOM algorithm to increase the accuracy and reliability of clustering.

For example, Panu Somervuo and Teuvo Kohonen [2] discuss clustering large protein sequence databases using an extended SOM that allows the creation of clusters of protein sequences without converting the data into histogram vectors. In his study, Mark Van Hulle [3] analyzes the basic SOM algorithm, its properties, and the possibility of extending it to work with categorical data, time series, and tree structures. Jens Claussen [4] proposes the Winner-Relaxing Self-Organizing Maps (WRSOM) approach, which ensures the stability of the cluster location. Despite the success of these approaches, the problem of variability of results is still relevant.

The study by Melody Kiang, Michael Hu, Dorothy Fisher [5] considers the use of an extended version of Kohonen's self-organizing maps to segment the market of telecommunications companies based on behavioral and demographic factors, including the frequency of long-distance calls, household structure, and so on. By using the enhanced SOM algorithm, they were able to achieve better results than standard clustering methods such as factor analysis and *k*-means. However, the authors note that the stability of the resulting clusters could be improved, which opens up prospects for further improving the efficiency of this method. Similarly, in a study by Wei Wang, Shiwei Xu, Hong Ouyang, and Xinyu Zeng [6], where SOM was used to optimize the parameters of the power systems of unmanned electric drive chassis, improvements were proposed by combining SOM with an advanced genetic algorithm that uses isolated niches to improve the accuracy of the results. This increased the convergence rate and improved the global search capability of the algorithm, providing a more accurate clustering of the initial populations. This approach demonstrates the potential of combining SOM with other algorithms to solve complex multitask optimization problems, which emphasizes the importance of further research to improve the stability and accuracy of SOM results for various engineering and practical applications.

The study by V. Dyachenko, O. Lyashenko, B. Ibrahim, O. Michal, and Y. Koltun [7] proposed a modification of Kohonen's self-organizing maps with a parallel learning algorithm that can significantly increase the speed of data

processing in multi-core processor systems. This approach demonstrates effectiveness in the tasks of clustering large amounts of data, ensuring the adaptation of the algorithm to a dynamic environment. The work of Rodrigo Cavalcanti, Bruno Pimentel, Carlos Almeida, and Renata Souza [8] presents a new variant of the Fuzzy Kohonen Clustering Network (FKCN), which uses the fuzzy *c*-means membership function instead of a fixed learning coefficient. This approach takes into account the intraclass and interclass variance, which allows to obtain better clustering results on real and synthetic data sets.

The works of N.I. Furmanova, O.Y. Farafonov, O.Y. Malyi, Y.O. Sitsilitsyn, V.O. Dyachenko, O.P. Mikhal, E.A. Egorova, V.G. Ivanov, E.S. Sakalo [9–11] consider various approaches to improving SOM, which demonstrate the wide application of this method in optimization, clustering, and data analysis. In particular, N.I. Furmanova, O.Y. Farafonov, and Y.O. Sitsilitsyn study the integration of SOM with genetic algorithms to reduce computational costs and avoid local minima in multidimensional optimization problems. V.O. Dyachenko and O.P. Mikhal proposed an improvement of SOM for work in distributed energy-critical sensor networks by parallel selection of several winning neurons, which reduces power consumption and optimizes computation time. E.A. Egorova, V.G. Ivanov, and E.S. Sakalo use the Kalman-Mayne filter to adapt SOM, providing accurate clustering even in the presence of noise in the data.

Despite the progress made, all authors note the importance of further research aimed at selecting optimal SOM parameters, such as the neighborhood function, initialization of the weights, and adaptation of the algorithm to dynamic data.

Given the need to increase the stability of SOM clustering results, the purpose of this study is to develop an improved approach to its application that ensures reproducibility of results.

The variability of SOM results is caused by several factors. First of all, the standard implementation of the algorithm randomly initializes the neural weights, resulting in different initial conditions even for the same input data. In addition, the learning process involves stochastic samples selection, which also introduces randomness at each stage of clustering. As a result, even if the algorithm is run repeatedly with the same data and parameters, SOM may generate different clusters.

The non-reproducibility of clustering complicates the practical application of SOM, especially in cases where stability of results is required. For example, in billing systems of telecommunication companies, the analysis of customer behavior data requires accurate and stable segmentation to generate personalized tariffs, etc. The variability of clustering results makes it difficult to accurately identify customer groups, which can lead to errors in understanding their needs and creating appropriate marketing strategies.

Thus, in order to eliminate the variability of SOM clustering results, it is necessary to develop a method that ensures reproducibility of the output data. In this study, we propose an approach that involves the introduction of a proprietary random number generator and a seed parameter for fixed initialization of network weights. This solution will eliminate the stochastic influence of the algorithm on the clustering results and increase the reliability of data analysis in telecommunication systems.

PURPOSE AND OBJECTIVES OF THE STUDY

The purpose of the study is to develop an improved algorithm for Kohonen's self-organizing maps that ensures stability and reproducibility of clustering results. This improvement is aimed at eliminating the variability of results arising from the random initialization of weight coefficients and stochastic selection of samples during training.

The following tasks were set to achieve this goal:

- identify the key factors that cause variability in SOM clustering results and assess their impact on the stability of the algorithm;
- to develop a method to improve the SOM algorithm by introducing its own random number generator and seed parameter to fix the initial training conditions;
- to test the effectiveness of the proposed method by analyzing data from telecommunications companies' billing systems and assess its impact on the stability and reproducibility of clustering results.

MATERIALS AND METHODS OF RESEARCH

The Kohonen algorithm

Self-organizing maps are based on Kohonen neural networks and are designed to visualize multidimensional objects on a two-dimensional map, where the distances between objects correspond to the distances between their vectors in a multidimensional space, and the feature values themselves are displayed in different colors and shades [12].

The basic idea behind SOM is to create a two-dimensional mapping structure in which neighboring nodes on the map reflect the similarity between data. Each node on the map has weights that represent vectors in the feature space. During SOM training, these weights are changed to match the structure and distribution of the data [13; 14].

The network construction is based on competitive learning, where the output nodes (neurons) compete with each other for "victory". In the course of the competition, during the training process, neurons are selectively tuned for different input examples [15].

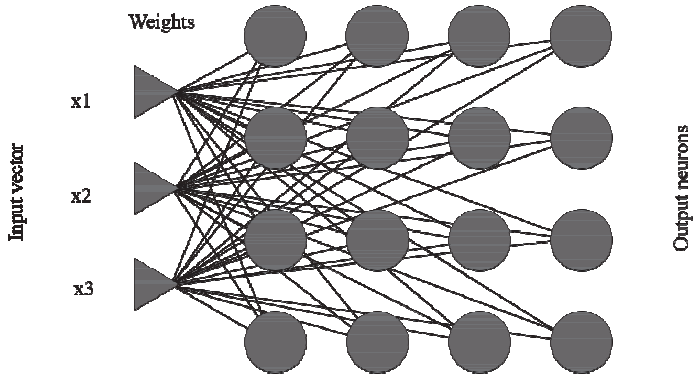


Fig. 1. Kohonen's network model

Input neurons form the input layer of the network, which contains one neuron for each input field. As in a regular network, input neurons do not participate in the training process. Their task is to transfer the values of the input fields of the initial sample to the neurons of the output layer. Each connection between neurons has a certain weight, which is randomly set in the interval [0;1] during initialization. The learning process consists in adjusting the weights. Unlike most neural networks, the Kohonen network has no hidden layers: the data from the input layer is sent directly to the output layer, whose neurons are arranged in a one- or two-dimensional grid of rectangular or hexagonal shape [16].

During SOM training, the following main stages are performed:

1. Competition: each output neuron calculates the distance between its weight vector and the input vector. The neuron with the smallest distance is declared the winner.

2. Cooperation: the winning neuron determines a group of neighboring neurons that also participate in the weight adjustment. This ensures the similarity of the weight vectors between neighboring neurons.

3. Adaptation: the weights of the winning neuron and its neighbors are adjusted to get closer to the input vector, promoting network self-organization and clustering.

The learning process of the Kohonen network involves a gradual decrease in the learning rate, which depends on the number of iterations. The training is divided into two phases: coarse tuning (with a larger influence radius and faster learning speed) and fine tuning (with a smaller radius and slower adaptation).

At the initial stage, if there is no a priori information about the distribution of data in the sample, the neuronal weights are initialized with random values. At the same time, the initial values of the learning rate and the learning radius R are set, which determines the number of neurons that are considered neighbors of the winning neuron and change their weights along with it. At the beginning of training, the radius R has a maximum value and gradually decreases with each iteration, which allows the network to accurately adapt to the data structure.

The Kohonen network training algorithm is based on the principles of unsupervised learning, i.e., without a teacher, and includes seven stages [15–18]:

1. Setting up the network structure (the number of neurons in the Kohonen layer).

2. Initialize the weight coefficients with random values according to the formula:

$$w_{ij} = \left(random\left(0;1\right)*\left(\max\left(x_{ni}\right)-\min\left(x_{ni}\right)\right)\right)+\min\left(x_{ni}\right),$$

where x_{ni} is an input vector; w_{ij} ia a vector of weight coefficients.

3. Competition. Supplying a random training example of the current training iteration to the network inputs and calculating the Euclidean distances from the input vector to the centers of all clusters:

$$D\left(W_j, X_n\right) = \sqrt{\sum_i \left(w_{ij} - x_{ni}\right)^2},$$

where x_{ni} is an input vector; w_{ij} is a vector of weight coefficients.

The output neuron whose weight vector has the smallest distance to the object feature vector is declared the winner.

4. Merge. All neurons located within the training radius relative to the winning neuron are identified.

5. Adjustment. According to the smallest of the values of *Rj*, the winning neuron *j* is selected, which is closest to the input vector in terms of values. For the selected neuron (and only for it), the weight coefficients are corrected:

$$w_{ij}^{new} = w_{ij}^{current} + l * \left(x_{ni} - w_{ij}^{current} \right),$$

where x_{ni} is an input vector; w_{ij}^{new} ia a new vector of weight coefficients; $w_{ij}^{current}$ is the current vector of weight coefficients; l is the learning rate coefficient.

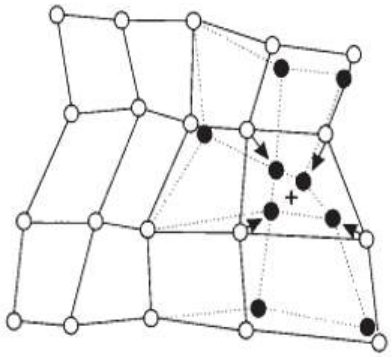


Fig. 2. Adjusting the weights of neurons

6. Correction. Changes the learning rate parameter according to the specified law.

$$l^{new} = l * exp\left(i^{current} - i \right),$$

where l^{new} is the adjusted learning rate parameter; l is the initial learning rate parameter; $i^{current}$ is the current iteration; i is the total number of iterations.

7. The cycle is repeated from stage 3 (competition) until the end condition is met: stabilization of the neural network outputs or the specified number of iterations.

Technical implementation of the SOM algorithm improvement

In order to ensure reproducibility of clustering results, the standard implementation of the Kohonen Self-Organizing Maps algorithm was improved in this study. The main goal of the improvements was to eliminate the variability of results caused by random initialization of weights and stochastic samples selection during training. The proposed solution includes the implementation of a proprietary random number generator with the ability to fix initial conditions using the seed parameter.

At the stage of improving the SOM algorithm, a special random number generator was implemented that uses the mathematical function of sine to generate values. The code of this generator is implemented as a RandomGenerator class in C# (Fig. 3). The main feature of this generator is the

ability to fix the initial state using the seed parameter, which reduces the variability of the initial conditions and, as a result, stabilizes the clustering results.

Implementation features:

- The seed: parameter is set when creating an instance of the RandomGenerator class, which determines the initial state of the generator. This ensures the determinism of the sequence of pseudo-random numbers.
- Generation algorithm: a pseudo-random number is calculated as the residual of multiplying a sine by a scaling factor of 10000. This allows to get a uniform distribution of values within [0; 1).
- Incrementing the seed: after each call to the Next() function, the seed value is incremented, ensuring a consistent change in the output values.

```
using System;

namespace SOM_Kohonen_WpfApp.Service
{
    3 references | Oleksandr, 75 days ago | 1 author, 1 change
    public class RandomGenerator
    {
        private int _seed;

        1 reference | Oleksandr, 75 days ago | 1 author, 1 change
        public RandomGenerator(int seed) => _seed = seed;

        1 reference | Oleksandr, 75 days ago | 1 author, 1 change
        public double Next()
        {
            var x = Math.Sin(_seed++) * 10000;
            return x - Math.Floor(x);
        }
    }
}
```

Fig. 3. Implementation of a random number generator using the seed parameter

Fig. 4. The main window of the application

Fig. 5. Neural network training window, entering the seed parameter

The generated generator is used to initialize the weights of the SOM neurons. Each time the algorithm is run with the same value of the seed parameter, the neuronal weights receive the same initial values, making it impossible to vary clusters due to random initialization. In addition, the seed parameter controls the order in which training samples are selected during network training, which eliminates the stochastic influence on the clustering process.

To ensure the convenience of working with the advanced algorithm, an interface based on Windows Presentation Foundation (WPF) was created. The interface is designed to provide the user with easy access to all key functions. In particular, the user can upload input data, configure training parameters such as map dimension, number of iterations and seed value, which allows to control the initial training conditions. In addition, after training is complete, the user can view the clustering results in the form of a Kohonen map, which provides visualization of the results and allows detailed analysis of the cluster distribution.

RESEARCH RESULTS

Using the created application based on the improved Kohonen algorithm, the customer base of the telecommunications company was clustered with different values of the seed parameter, the results of which are shown in Figs. 6 and 7. The data contained information about the demographic characteristics of customers, their activity and the intensity of service use, which made it possible to create clear customer segments by behavioral characteristics.

Clustering with different seed values

Fig. 6 shows the clustering results for different values of the seed parameter (85690 and 368). It can be seen that changing this parameter leads to a change in the shape and location of the clusters. For example, the clusters highlighted in the figure change their boundaries significantly: the same customer segment can move around the map and change its shape and size. This makes it difficult to identify stable customer groups and can lead to difficulties in analyzing them accurately.

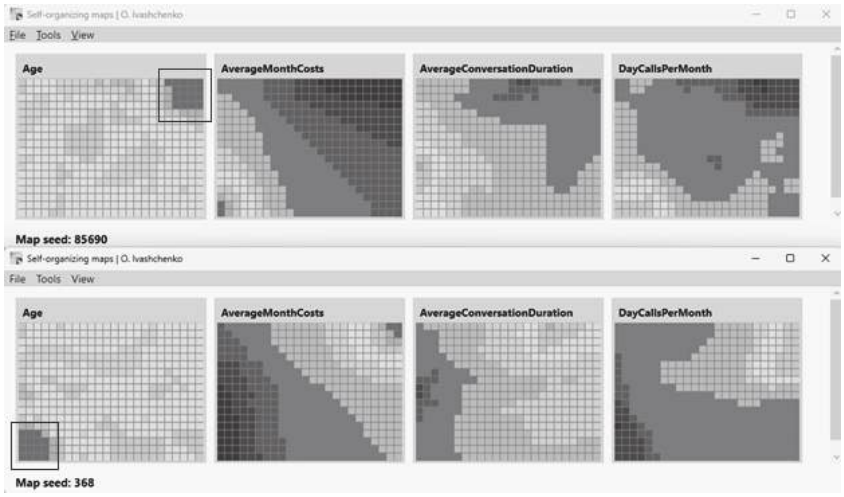


Fig. 6. Generated maps for different values of the seed parameter

Clustering with the same seed value

Fig. 7 shows the results of clustering using the same value of the seed parameter (6548). In this case, all resulting Kohonen maps are identical, regardless of the number of algorithm runs. The clusters highlighted in the figure have the same shape and location, confirming the stability of the algorithm. This shows that using the same value of the seed parameter guarantees not only the stability of individual clusters, but also the complete reproducibility of the entire Kohonen map.

Fig. 7. Generated maps for the same values of the seed parameter

Therefore, the use of a fixed seed parameter allows to achieve full reproducibility of the results, which is impossible in the case of random initialization of the weights.

The seed values (85690, 368, and 6548) used in this study were chosen randomly. However, each identical set of seed values guarantees identical clustering results, ensuring that the shape and location of the clusters remain consistent. This improvement eliminates the variability in results that occurred due to the random initialization of weights and samples selection during the algorithm. This emphasizes the reproducibility of results, which is an important aspect for scientific research and practical use of the SOM algorithm.

DISCUSSION OF THE OBTAINED RESULTS

The results of the study confirmed that the improved Kohonen Self-Organizing Map algorithm with the implemented seed parameter for fixed initialization of network weights ensures stability and reproducibility of clustering results. Using the same seed value eliminated the problem of variability caused by random initialization of weights and stochastic samples selection during network training. This is especially important in tasks where accurate group identification is critical for decision making.

A comparative analysis of the clustering results showed that with different values of the seed parameter (Fig. 6), there was significant variation in the size, shape, and location of the clusters. The same customer segment could change its boundaries or move around the map, making it difficult to interpret and analyze the data consistently. Instead, clustering with a fixed seed value (Fig. 7) ensured complete identity the Kohonen maps on each run of the algorithm, confirming the stability and reliability of the results.

The practical significance of the results obtained is particularly relevant for telecommunications companies, where the stability of customer segmentation plays a key role in the development of personalized tariff plans and marketing strategies. The improved algorithm provides reliable analysis of customer behavior data, which enables more accurate marketing budget calculations, minimizes customer churn, and increases the efficiency of customer base management.

Compared to other approaches to improve SOM, such as combining it with genetic algorithms or using modifications of WRSOM, the proposed method is simple to implement and does not require additional computational resources. This makes it an effective solution for tasks that require stable results at minimal technical costs.

Some limitations of the proposed approach should also be noted. The value of the seed parameter needs to be adapted for different data sets, which may require additional testing to achieve optimal results. In future research, it is advisable to consider automating the selection of the seed parameter or adapting it to work with dynamic data. This will increase the flexibility and versatility of the algorithm for a wider range of tasks.

Thus, the results of the study showed that the introduction of the seed parameter allows to achieve stable and reproducible clustering, which is important for scientific research and practical use in business intelligence, especially for telecommunication systems.

CONCLUSIONS

1. The key factors of variability in the results of SOM clustering are identified. The main reasons for the instability of the results are the random initialization of neuronal weights and the stochastic selection of training samples during network training. These factors lead to different locations and shapes of clusters for the same input data, making stable analysis impossible.

2. A method to improve the SOM algorithm is developed. It is proposed to introduce its own random number generator with the ability to fix the initial conditions using the seed parameter. This allows to set the same initial neuronal weights and a deterministic sequence of training samples selection, which eliminates variability and ensures the stability of the clustering results.

3. The effectiveness of the proposed approach is tested. The results of clustering the customer base of a telecommunications company have shown that using the same value of the seed parameter ensures full reproducibility of Kohonen maps. This allows for stable identification of customer groups, simplifying data analysis for the development of targeted marketing strategies and personalized tariff plans.

REFERENCES

1. "Teuvo Kohonen, Timo Honkela, Kohonen Network," *Scholarpedia*. 2007. Accessed on: 12 June 2024. Available: http://www.scholarpedia.org/article/Self-organizing_feature_map
2. Panu Somervuo, Teuvo Kohonen, "Clustering and Visualization of Large Protein Sequence Databases by Means of an Extension of the Self-Organizing Map," *Lecture Notes in Computer Science*, vol. 1967, 2000. doi: https://doi.org/10.1007/3-540-44418-1_7
3. Marc M. Van Hulle, "Self-Organizing Maps," *Handbook of Natural Computing*. Springer, Berlin, Heidelberg, pp. 585–622, 2012.
4. Jens Christian Claussen, "Winner-Relaxing Self-Organizing Maps," *Neural Computation*, 17(5), pp. 996–1009, 2005. doi: https://doi.org/10.1162/0899766053491922
5. Melody Y. Kiang, Michael Y. Hu, Dorothy M. Fisher, "An extended self-organizing map network for market segmentation — a telecommunication example," *Decision Support Systems*, vol. 42, issue 1, October 2006, pp. 36–47. doi: https://doi.org/10.1016/j.dss.2004.09.012
6. W. Wang, S. Xu, H. Ouyang, X. Zeng, "Parameter Optimization of the Power and Energy System of Unmanned Electric Drive Chassis Based on Improved Genetic Algorithms of the KOHONEN Network," *World Electric Vehicle Journal*, 14(9), 260, 2023. doi: https://doi.org/10.3390/wevj14090260
7. V. Diachenko, O. Liashenko, B.F. Ibrahim, O. Mikhal, Yu. Koltun, "Kohonen network with parallel training: Operation structure and algorithm," *Int. J. Adv. Trends Comp. Sci. Eng.*, vol. 8, no. 1.2, pp. 35–38, 2019. doi: https://doi.org/10.30534/ijatcse/2019/0681.22019
8. Rodrigo B. de C. Cavalcanti, Bruno Pimentel, Carlos W.D. de Almeida, Renata M.C.R. de Souza, "A Multivariate Fuzzy Kohonen Clustering Network," *IEEE Transactions on Neural Networks*, vol. 31, no. 4. pp. 75–82, 2020. doi: https://doi.org/10.1109/IJCNN.2019.8852243
9. N.I. Furmanova, O.Y. Farafonov, O.Y. Malyi, Y.O. Sitsilitsyn, "Improvement of the method of searching for solutions to solve the optimization problem using a genetic algorithm by preliminary clustering," (in Ukrainian), *Instrumentation Technology*, no. 2, pp. 6–9, 2017. Available: https://elar.tsatu.edu.ua/server/api/core/bitstreams/f045c4ca-7d17-4c9c-a1cb-7b2df9aafe7e/content
10. V.O. Dyachenko, O.F. Mikhal, "Prospects for the use of of the classical Kohonen algorithm in distributed energy-critical sensor networks," (in Ukrainian), *Control, Navigation and Communication Systems*, no. 4, pp. 75–79, 2023. doi: https://doi.org/10.26906/SUNZ.2023.4.075
11. E.A. Egorova, V.G. Ivanov, E.S. Sakalo, "Optimization of process of the Kohonen self-organizing map based on the Kalman-Mayne filter," (in Russian), *Control, Navigation and Communication Systems*, issue 4(8), pp. 52–55, 2008. Available: https://dspace.nlu.edu.ua/bitstream/123456789/6713/1/Ivanov_52-55.pdf
12. Achraf Khazri, "Self-Organizing Maps (Kohonen's maps)," *Medium*, [website]. 2019. Available: https://medium.com/data-science/self-organizing-maps-1b7d2a84e065
13. T. Kohonen, "The self-organizing map," *Proceedings of the IEEE*, 78(9), pp. 1464–1480, 1990.
14. E.O. Kaminsky, *Forecasting system for supporting activities using deep learning. Bachelor's thesis.* NTUU "KPI", Kyiv, 2023, p. 78.
15. *Kohonen Networks*, [website]. Accessed on: 13.11.2022. Available: https://ppt-online.org/46514
16. T. Kohonen, *Self-Organizing Maps*. Springer-Verlag Berlin Heidelberg, 2001.

17. S. Kaski, "Self-Organizing Maps," in *C. Sammut, G.I. Webb (Eds.) Encyclopedia of Machine Learning*. Springer, Boston, 2011, pp. 886–888.
18. "Self-Organizing Maps, SOM," *Studfile.net*. Accessed on: 15 Nov. 2022. Available: https://studfile.net/preview/3021431/

Received 25.12.2024

INFORMATION ON THE ARTICLE

Oleksandr V. Ivashchenko, ORCID: 0009-0007-5470-9137, National Transport University, Ukraine, e-mail: alexander.ivashchenkoo@gmail.com

Serhii S. Fedin, ORCID: 0000-0001-9732-632X, National Transport University, Ukraine, e-mail: sergey.fedin1975@gmail.com

УДОСКОНАЛЕННЯ АЛГОРИТМУ SOM ДЛЯ ЗАБЕЗПЕЧЕННЯ СТАБІЛЬНОСТІ ТА ВІДТВОРЮВАНОСТІ РЕЗУЛЬТАТІВ КЛАСТЕРИЗАЦІЇ ДАНИХ / О.В. Іващенко, С.С. Федін

Анотація. Запропоновано метод удосконалення алгоритму навчання самоорганізаційних карт Кохонена (Self-Organizing Maps, SOM) для забезпечення стабільності та відтворюваності результатів кластеризації, що є актуальним завданням у ході роботи з великими обсягами даних. SOM широко застосовується у задачах кластеризації та візуалізації, особливо у сферах, де необхідно аналізувати багатовимірні структури даних, зокрема у білінгових системах телекомунікаційних компаній, фінансовому аналізі тощо. Стандартна реалізація SOM, яка включає випадкову ініціалізацію ваг і стохастичний вибір зразків під час навчання, призводить до значної варіативності кластерів навіть за умови використання однакових вхідних даних та ідентичних параметрів налаштування тренування мережі. Це ускладнює застосування цього алгоритму у випадках, коли потрібна стабільність та відтворюваність результатів. Для вирішення цієї задачі запропоновано модифікацію алгоритму, що включає власний генератор випадкових чисел і введення параметра seed для фіксації початкових умов навчання. Це дає змогу знизити варіативність і забезпечити відтворюваність результатів кластеризації для підвищення достовірності аналізу та придатності алгоритму SOM за використання в реальних бізнес-завданнях. Запропонований метод протестовано на даних білінгових систем, де відтворюваність результатів кластеризації має критичне значення для ефективної роботи з клієнтськими сегментами, розроблення таргетованих маркетингових стратегій, персоналізованих тарифних планів тощо.

Ключові слова: самоорганізаційні карти Кохонена (SOM), кластеризація даних, параметр seed, відтворюваність результатів, генератор випадкових чисел.

UDC 004.89:556.3/5:528.8
DOI: 10.20535/SRIT.2308-8893.2026.1.09

LONG-TERM MONITORING OF SURFACE WATER QUALITY AND GROUNDWATER POTENTIAL USING COMPUTATIONAL INTELLIGENCE, GIS TECHNOLOGIES, AND REMOTE SENSING

S.V. KLIMOV, T.V. STAROVOIT

Abstract. Water scarcity and declining water quality due to population growth, urbanization, industrialization, and climate change highlight the importance of effective water management. Advances in remote sensing, cloud computing, and computational intelligence underscore the need to utilize modern technologies for monitoring surface water quality. This research involves the development of hybrid intelligent models using Landsat and Sentinel-2 images and WISE data with hybrid deep learning networks to evaluate surface water quality and groundwater potential. Correlation analysis revealed strong connections between remote sensing data and water quality parameters (such as chlorophyll-a, dissolved oxygen, nitrogen, and phosphorus). The hybrid models surpassed traditional machine learning methods, demonstrating their effectiveness in real-world water management.

Keywords: computational intelligence, fuzzy logic, remote sensing, satellite imagery, surface water quality monitoring, groundwater potential assessment, hybrid neural networks, NEFCLASS-EM, TS-FNN, Fuzzy C-Means, K-Means.

INTRODUCTION

Water is crucial for human health, food security, economic growth, energy production, and ecosystems. However, factors such as population growth, urbanization, industrial development, increased demand, and water misuse have made water scarce and expensive, particularly in developing countries. To address this issue, various strategies have been developed to improve water quality and quantity by 2030 [1]. In Europe, the Water Framework Directive (WFD) [2] aims to achieve good status for water resources. To assess the status, it's essential to monitor biological, hydro morphological, and physicochemical water quality indicators. According to this directive, rivers with a catchment area of more than 10 km^2 and lakes with an area of more than 0.5 km^2 should be included in the assessment and monitoring of water status [2].

Water quality parameters are traditionally determined by collecting samples on-site and analyzing them in the laboratory [3]. This method provides high accuracy, but is also labor-intensive and time-consuming, requiring significant financial investment. In addition, the traditional method determines the concentration of the required indicators only at the point of sampling. Meanwhile, the water quality in water bodies is rarely constant due to unpredictable events, such as accidental or deliberate leaks from industrial facilities and other factors. This makes accurate water quality monitoring a challenging task.

To overcome these limitations, we used remote sensing technology (RS), which has the advantage of large spatial coverage and high temporal resolution, which has been used to identify and monitor water resources more efficiently and effectively [4–6]. Remote monitoring of water quality indicators is based on establishing a correlation between the monitoring data and the corresponding surface reflection. Spectral characteristics of water are functions of hydrological, biological, and chemical characteristics of water [7]. Specifically, the amount of wave radiation at different wavelengths reflected from the water surface can be used directly or indirectly to detect water quality indicators [8].

Pure water can reflect light with a wavelength of more than 600 nm, which provides a high blue-green reflectance while absorbing radiation in the near-infrared (NIR) spectrum and beyond. Increasing the chlorophyll concentration increases the absorption of red (R) light and strongly absorbs blue (B) light, while the peak of reflection is located in the green (G) part of the spectrum [9]. The transparency of water depends on the total concentration of suspended solids. This concentration is a measure of the weight of inorganic particles suspended in the water column and is responsible for most of the scattering. By affecting the scattering of light, the suspended solids concentration (SSC) in water directly controls the transparency and oxygen content of a water body [10]. An increased concentration of SSC causes a shift in the peak from the G to the R region and increases the reflectance of water in the NIR region.

The relationship between surface reflectance and the concentration of water quality parameters is indirect and non-linear. This makes their estimation problematic, especially when based on traditional empirical algorithms. Over the past decade, the advancement in computing power and the development of artificial intelligence and machine learning (ML) algorithms have led to an increased use of these technologies to solve this problem. The most common machine learning models used in water quality assessment tasks are Random Forest (RF), Support Vector Machine (SVM), and Artificial Neural Network (ANN).

Studies [11–14] have demonstrated that Artificial Neural Networks (ANNs) and Support Vector Machines (SVMs) deliver excellent performance in monitoring both optically active and inactive water quality indicators. Generally, artificial neural networks, as a linear approximation method, offer greater flexibility for monitoring water quality indicators. However, the accuracy of machine learning models typically depends on the chosen model and the quality of the training data. Developing ANN models requires large training datasets and significant experience to construct the optimal architecture for artificial neural networks. Using too many layers can lead to overfitting, which involves fitting noise in the training data [15]. Conversely, a small number of layers can lead to underfitting, where the model cannot adequately represent the complexity of the data [15].

This study aims to develop and compare hybrid computational intelligence models, specifically neuro-fuzzy neural networks, using spatial data from remote sensing and geographic information systems (GIS). It proposes combining neuro-fuzzy neural networks with metaheuristic and remote sensing algorithms to assess water quality from satellite images and evaluate their effectiveness. These hybrid models, integrating remote sensing and GIS, offer innovative methods for assessing water quality and groundwater potential. They can accurately identify factors contributing to water quality deterioration and unexpected surface formations in inland water bodies, while also examining their long-term impact on ecological status.

MATERIALS AND METHODS

Research area

We chose the territory of Kyiv and the Kyiv region (Ukraine) as the research area (Fig. 1). The study area is located in south-eastern Europe. Most of the rivers belong to the Black Sea basin. The largest river is the Dnipro, which we chose for our research. We used data from the Water Information System of Europe (WISE) to train artificial neural networks and neuro-fuzzy neural networks.

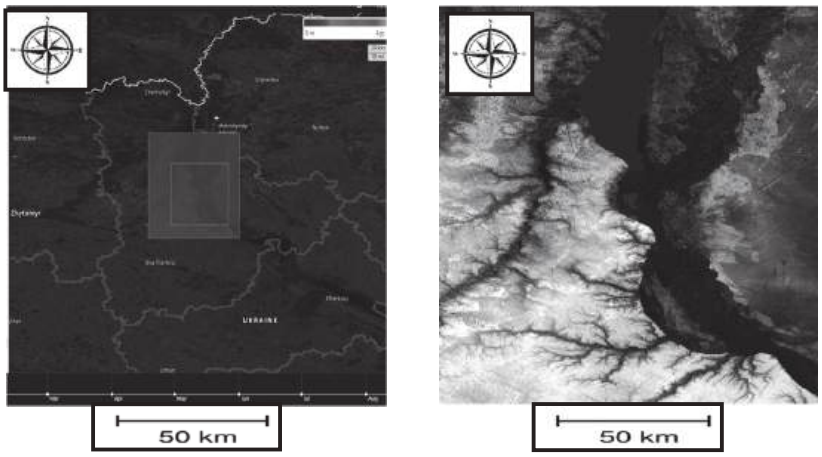


Fig. 1. Location of the study area in Kyiv and Kyiv region, Ukraine. The green rectangle marks the region of interest including key sampling sites along the Dnipro River

Data Preparation

Remote monitoring of water quality indicators is based on the correlation between on-site measurements and the corresponding surface reflectance. For this study, Landsat and Sentinel-2 satellite images of the surface over Europe from 2010 to 2024 were used. In total, more than two thousand images were analyzed to create time series and train monitoring models. Also used some of the materials from the DHI educational resource [16].

Landsat satellites achieve maximum surface coverage once every 16 days, with a spatial resolution of 30 m for multispectral bands. The Google Earth Engine API, integrated into Google Colab, was used as an access point to the imagery.

The surface reflectance values for each point were obtained from available Landsat Surface Reflectance Level 2A images. Cloud and shadow masking were performed to ensure clear water pixels. The resulting table included the identifier of the monitoring stations, the corresponding surface reflectance value, and the date of the survey. The surface reflectance was filtered by date to match the on-site data, with a maximum time interval of 3 days between the on-site sampling and the satellite overpass.

We used Pearson correlation analysis to explore the connection between remote sensing and in situ data, using the correlation coefficient (r). Based on the correlation, we identified a specific set of input data for each water quality indicator. The data was then standardized to a normal distribution with a mean of 0 and a standard deviation of 1, and then divided into training and test sets (80 % and 20 %, respectively).

Fuzzy neural network model

Fuzzy theoretical TS systems can apply fuzzy mathematical rules to generate more complex nonlinear functions. This allows the system to reduce the number of fuzzy rules needed when dealing with problems involving multiple variables [17]. The fuzzy theoretical system TS is typically defined using "if-then" logic, and its fuzzy conclusion is expressed as follows [18]:

$$R^i : If\ x_1\ is\ A_1^i, x_2\ is\ A_2^i, \dots\ x_k\ is\ A_k^i, \tag{1}$$

$$Then\ y_i = p_0^i + p_1^i x_1 + \cdots + p_k^i x_k,$$

where A_j^i is a fuzzy

set of a fuzzy system; p_j^i (j = 1, 2, …, k) are the parameters of the fuzzy system; y_i is the initial value obtained by the fuzzy rule; input part i.e. If is fuzzy and output part i.e. $Then$ is deterministic part. This vague conclusion indicates that the output is a linear combination of the inputs [18].

Suppose that for inputs $x = [x_1, x_2, \dots, x_k]$, the degree of membership of each input variable x_j is first calculated by a fuzzy rule [18]:

$$\mu_{A_j^i} = \exp(-(x_j - c_j^i)^2 / b_j^i), \tag{2}$$

$$(j = 1, 2, \dots, k; i = 1, 2, \dots, n),$$

where c_j^i and b_j^i are the center and width of the membership function, k is an input parameter, and n is the number of fuzzy subsets [18].

Fuzzy operations are performed on each of the above degrees of membership, and the fuzzy operator is used as a concatenated multiplicative operator [18]:

$$\omega^i = u_{A_j^1}(x_1) \times u_{A_j^2}(x_2) \times u_{A_j^k}(x_k), \tag{3}$$

$$(i = 1, 2, \dots, n).$$

According to the results of the fuzzy calculation, the initial value of the model y_i is obtained [18]:

$$y = \frac{\sum_{i=1}^n \omega^i (p_0^i + p_1^i x_1 + \cdots + p_k^i x_k)}{\sum_{i=1}^n \omega^i}. \tag{4}$$

A Fuzzy Neural Network (FNN) is a hybrid model that integrates the learning capabilities of artificial neural networks with the interpretability of fuzzy logic systems. The network typically consists of four layers: an input layer, a fuzzification layer, a rule base, and an output layer.

The algorithm of this type of neural network is described using equation (5).

$$\mu_i(x_i) = \exp\left(-\frac{(x_i - c_i)^2}{2\sigma_i^2}\right), \tag{5}$$

where $x = (x_1, x_2 \ldots , x_n)$ input vector; and $\mu_i(x_i)$ is the membership function. The activation of rule j is described by equation (6). The output of the network is described by equation (7).

$$w_j = \prod_{i=1}^{n} \mu_i^j (x_i), \quad (6)$$

$$y = \frac{\sum_{j=1}^{r} w_j f_j(x)}{\sum_{j=1}^{r} w_j}, \; where \; f_j(x) = a_j^T x + b_j . \quad (7)$$

A Takagi–Sugeno Fuzzy Neural Network (TS-FNN) extends the FNN by using linear functions in the consequents of fuzzy rules. Instead of constant outputs, each rule produces an output of the form:

$$y_j = a_1^j x_1 + a_2^j x_2 + \cdots + a_n^j x_n + b^j. \quad (8)$$

The final output is a weighted sum of these rule outputs, normalized by the total rule strength. TS-FNNs offer higher precision and faster convergence compared to classical FNNs.

In the fuzzy neural network NEFCLASS (Neuro-Fuzzy CLASSification) classification is performed with automatic rule learning. Inputs are fuzzified by linguistic variables (low, medium, high) [19]:

$$\mu A_k(x) = max\left(0.1 - \left|\frac{x - c_k}{\sigma_k}\right|\right). \quad (9)$$

The rules are presented in the form: $If\ x_1\ is$ "High" *and* x_2 is "Low" $then\ Class = Water$

The process of class estimation through fuzzy inference and max-aggregation is described in equation (10), [20].

$$y = \arg\max_c \left(\max_j \left(\min_i \mu_i^j (x_i)\right)\right). \quad (10)$$

The NEFCLASS-EM model enhances NEFCLASS with a metaheuristic optimizer, such as the Electromagnetic Algorithm (EM). NEFCLASS-EM is calculated in a similar way as in equations (9), (10), but with optimization of parameters (centers, widths) through metaheuristics (equation (11)).

$$\theta^{(t+1)} = \theta^{(t)} + F_{EM}\left(\theta^{(t)}\right), \quad (11)$$

where F_{EM} – is the vector of the force of attraction/repulsion between candidates. A schematic representation of the architecture of the four hybrid neural network models is shown in Fig. 2.

Fig. 2 illustrates the structural differences between four neuro-fuzzy models: FNN, TS-FNN, NEFCLASS, and NEFCLASS-EM. Each model includes key functional components such as input processing, fuzzification, rule evaluation, and output generation. The FNN and TS-FNN architectures are organized and layered, with TS-FNN producing linear outputs. NEFCLASS and NEFCLASS-EM focus on classification tasks, where NEFCLASS-EM integrates a metaheuristic optimizer (EM) to enhance the rule parameter optimization.

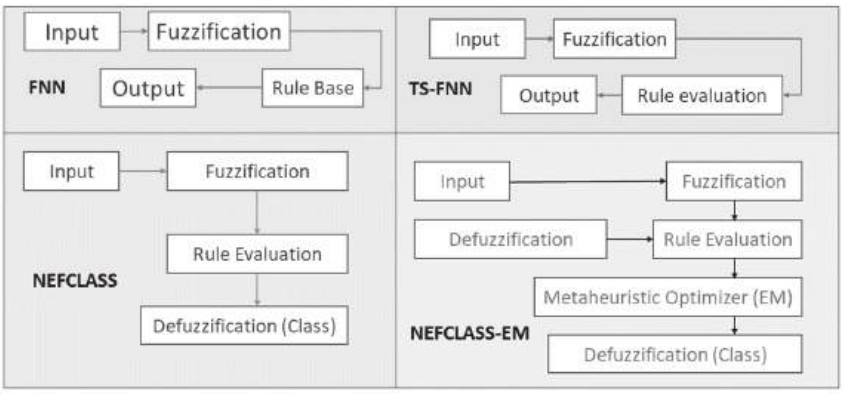


Fig. 2. Architectures of the fuzzy neural network models used in the study

In this study, fuzzy neural networks were applied to predict water quality indicators and groundwater potential. A fuzzy neural network model usually has four levels: an input level, a fuzzification level, a fuzzy rule calculation level, and an output level [21]. The input layers of the network model are connected through the vector x_i, so the number of nodes of the model network is consistent with the dimension of the input vector. The fuzzification layer uses the membership function of equation (3) to fuzziness the input values to obtain the membership value $u_{A_j^i}$. The value of ω at the fuzzy computing level is obtained by using equation (4) of successive phase multiplication, and then the output value of the output data level in this fuzzy model system is obtained by equation (8). The fuzzy neural network learning algorithm is as follows [21]:

Step 1. Calculation error:

$$e = \frac{1}{2}(y_d - y_c)^2, \tag{12}$$

where the expected output of the network is y_c and the error between the expected output and the actual output is e [21].

Step 2. Correction of the coefficient:

$$p_j^i(k) = p_j^i(k-1) - \alpha \frac{\partial e}{\partial p_j^i}, \tag{13}$$

$$\frac{\partial e}{\partial p_j^i} = \frac{(y_d - y_c)\omega^i}{\sum_{i=1}^m \omega^i \cdot x_j},$$

where p_j^i is the coefficient of the neural network; α is the learning rate of the network; x_j is the input parameters of the network; ω^i is the continuous product of the membership of the input parameters [21].

Step 3. Correction of parameters:

$$c_j^i(k) = c_j^i(k-1) - \beta \frac{\partial e}{\partial c_j^i}, \tag{14}$$

$$b_j^i(k) = b_j^i(k-1) - \beta \frac{\partial e}{\partial b_j^i},$$

where b_j^i and c_j^i are the width and center value of the representative membership function in the fuzzy rule [21].

Indicators of assessment of accuracy of models

We used four metrics to evaluate the accuracy of the models: overall accuracy (OA), misclassification error (ME), omission error (OE), and ROC-AUC value. OA is the sum of pixels correctly classified as water divided by the total number of water pixels represented by the confusion matrix. OE is the number of pixels that belong to water but are classified as other surface types that can be identified by the error matrix column. The ROC-AUC value [22] is the area under the curve of the ratio of sensitivity (equation 15) to specificity (equation 16). This value ranges from 0.50 to 1. The higher the value, the better the performance. If the value exceeds 0.70, the classification result is reliable [20].

$$Sensitivity = \frac{TPR}{TPR+FNR}, \tag{15}$$

$$Specificity = \frac{TNR}{TNR+FPR}, \tag{16}$$

where TPR (true positive) and FNR (false negative) are pixels correctly and incorrectly classified as water, and TNR (true negative) and FPR (false positive) are pixels correctly and incorrectly classified as non-water [23].

To assess water quality, a water quality model's performance can be measured using various metrics. These include the coefficient of determination (R^2, Eq. 17), the mean absolute error (MAE, Eq. 18), the root means square error (RMSE, Eq. 19), the mean square error (MSE, Eq. 20), residual prediction deviation (RPD, Eq. 21), and confidence interval (CI, Eq. 22), [23].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - y_i')^2}{\sum_{i=1}^{n}(y_i' - \bar{y})^2}, \tag{17}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - y_i'|, \tag{18}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2}, \tag{19}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2, \tag{20}$$

$$RPD = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - y_i')^2}}, \tag{21}$$

$$CI = \left[1 - \frac{\sum_{i=1}^{n}|y_i' - y_i|}{\sum_{i=1}^{n}(|y_i' - \bar{y'}| + |y_i - \bar{y}|)}\right] \times \left[1 - \frac{\sum_{i=1}^{n}(y_i' - y_i)^2}{\sum_{i=1}^{n}(y_i' - \bar{y'})^2}\right]. \tag{22}$$

R^2 values should not be too high, as excessively high values lead to overfitting and lack of model portability, since this metric is sensitive to outliers [23]. Therefore, in practice, R^2 is often used in combination with RMSE, RPD, CI, and other indicators to balance the fitting accuracy and computational complexity [23, 24].

RESULTS OF SIMULATION

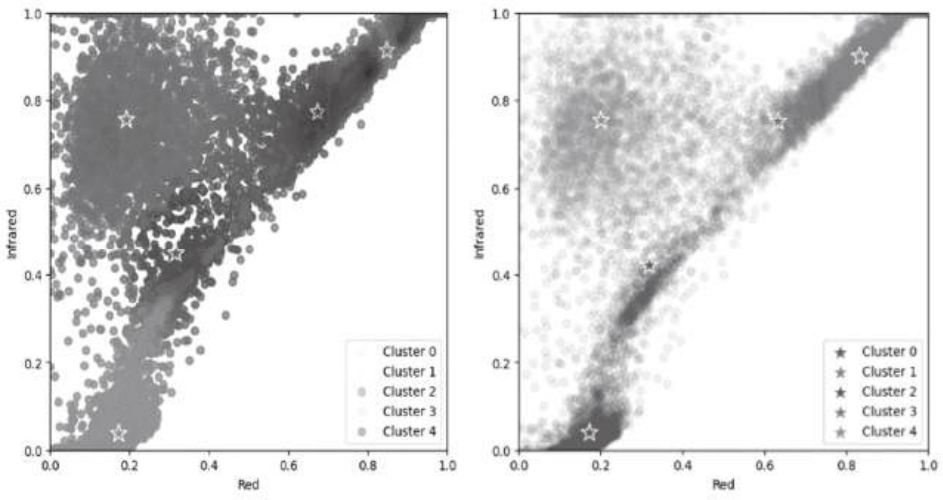
Aerospace research and geoinformation modeling

Satellite technology has a significant advantage in its ability to capture light beyond the visible spectrum, which is undetectable to the human eye. Infrared light, a type of radiation, can be detected by satellites like Sentinel-2. This infrared radiation can provide valuable information about surface temperature, vegetation conditions, and atmospheric conditions. For instance, healthy vegetation reflects more infrared light than unhealthy vegetation or non-vegetated surfaces.

The raster image measures 1009 pixels in height and 1014 pixels in width, covering approximately 10 kilometers in both dimensions. It contains four spectral bands: blue, green, red, and near infrared. The satellite captures data across 11 different wavelengths, and we specifically selected the blue, green, red, and infrared spectra for analysis.

The near-infrared range is very helpful for analyzing vegetation. In this setup, healthy vegetation appears bright red, while non-vegetated surfaces appear in other colors. This method improves the visibility of different ground cover types and allows us to see details that are not visible in normal light.

We first rendered the image in different color spaces and then focused on the red and infrared ranges. To do this, we created a scatter diagram where the reflection coefficient of the red pixel is shown on the x-axis and the infrared pixel on the y-axis (Fig. 3). Next, we converted the images into tabular data. Each row in the resulting table represents one pixel for a specific date. For instance, the first row corresponds to the pixel (1, 1) on February 22, 2022. Each column represents the intensity of a spectral band (blue, green, red, infrared).



a *b*

Fig. 3. Visualization of the results of clear and fuzzy clustering, where *a* – clear clustering; *b* – fuzzy clustering of pixels

To recognize clusters that correspond to water bodies, we applied the following intelligent algorithms: Fuzzy Neural Networks (FNN), Takagi–Sugeno Fuzzy Neural Networks (TS-FNN), NEFCLASS, and NEFCLASS-EM.

DEVELOPMENT OF COMPUTATIONAL INTELLIGENCE MODELS

Fuzzy neural network model

Fuzzy neural networks (FNNs) are hybrid models that combine fuzzy logic with the computational power of neural networks. They are designed to handle uncertainties and imprecise data in tasks such as classification, clustering, and regression. In the context of satellite imagery, FNNs are particularly useful because satellite data often contain noise, incomplete information, and inherent fuzziness. This is especially true when dealing with natural phenomena such as cloud cover, landforms, vegetation, and water bodies.

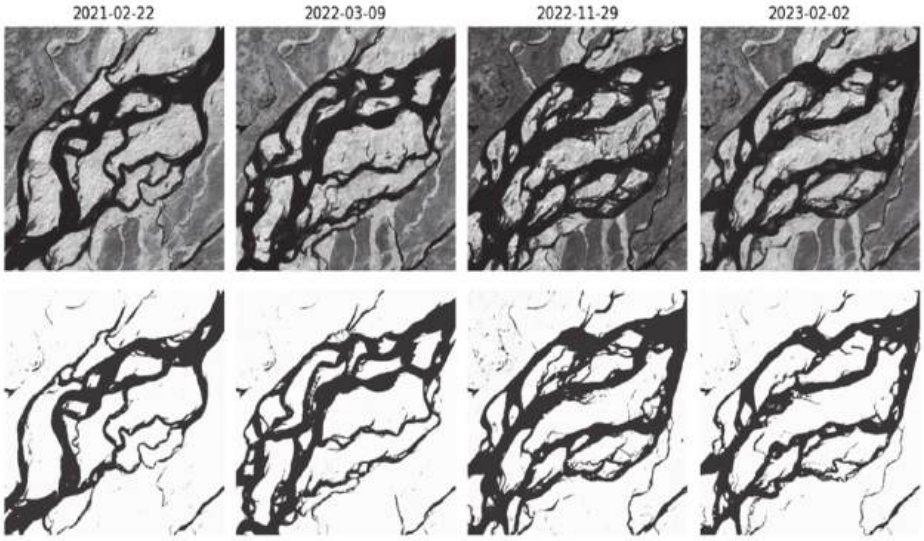


Fig. 4. Visualization of water recognition results on satellite images using a fuzzy neural network (FNN) is visualized

Fig. 4 shows the water extraction result, where we can see that the fuzzy neural network recognized water in satellite images very well. FNNs allow the use of fuzzy membership values (rather than binary solutions) to model these ambiguous or mixed pixels. Unlike traditional neural networks, FNNs assign membership degrees to different classes, so a single pixel can belong to multiple categories with different probabilities or degrees (e.g., 70 % forest, and 30 % water).

Takagi–Sugeno fuzzy neural network (TS-FNN) model

Takagi–Sugeno models are a type of fuzzy logic inference system. In these models, the output of fuzzy rules can be a linear combination of input variables or a constant. In TS-FNN, each fuzzy rule corresponds to a linear model or constant that is learned, and the output is calculated as the weighted average of the rule's output. The implementation of the TS-FNN model is depicted in Fig. 5. The model is trained to minimize the mean squared error (MSE) between the predicted outputs and the actual labels. Post-training, the network predicts cluster membership for each pixel in the satellite image, and the assigned clusters are then visualized.

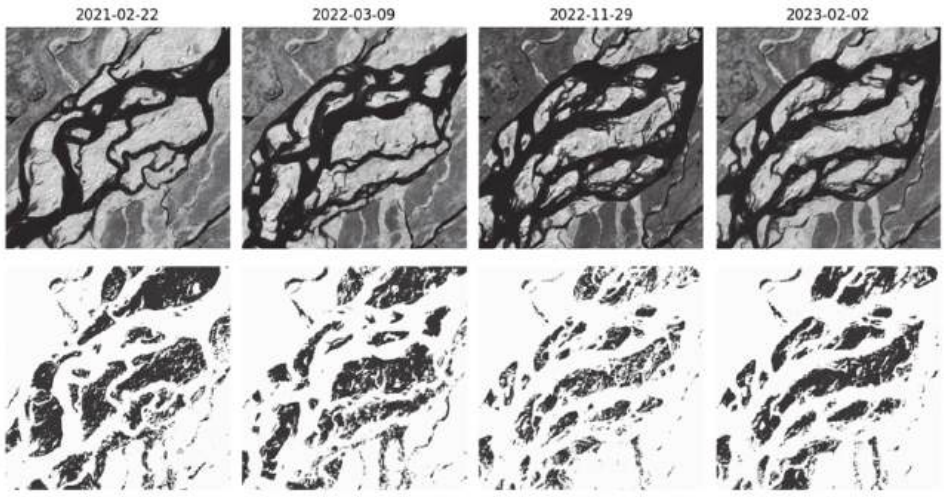


Fig. 5. Visualization of water recognition results on satellite images using Takagi–Sugeno fuzzy neural network (TS-FNN)

Fuzzy rules in TS-FNN are better interpreted because the output is a linear function of the input variables, making it easier to understand how the model makes its decisions.

NEFCLASS fuzzy neural network model

NEFCLASS is a neuro-fuzzy system specifically created for classification tasks. It integrates fuzzy logic with a feed-forward neural network framework. In this system, fuzzy rules are acquired from input data. The model generates fuzzy classification rules automatically, which are then refined using neural network training methods. The neural network adapts the fuzzy rules to minimize classification errors.

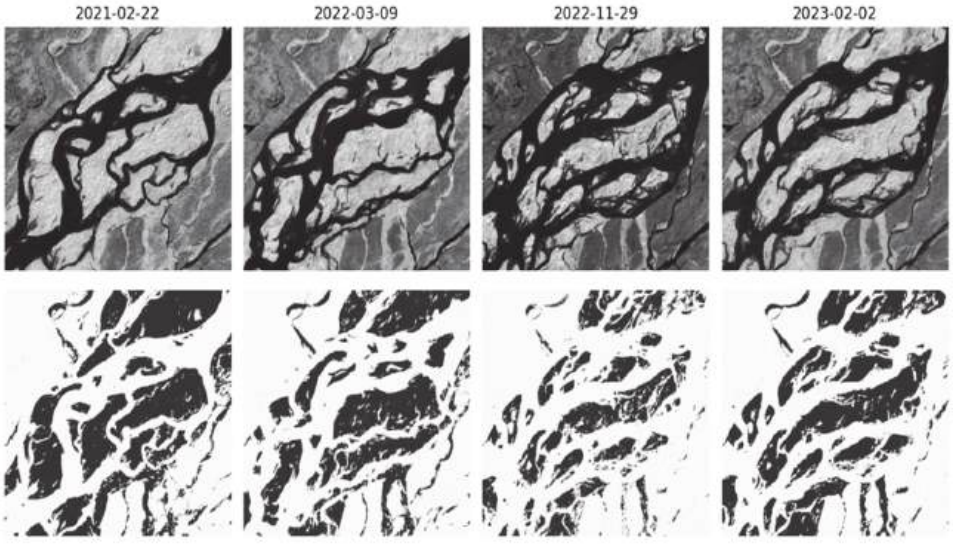


Fig. 6. Visualization of the results of water recognition on satellite images using the NEFCLASS fuzzy neural network

For simplicity, we implemented NEFCLASS-like functionality using a neural network to represent fuzzy rules, but we manually defined the fuzzification process. The result is shown in Fig. 6.

To implement this model, we utilized the same approach as in our previous models. This involved defining fuzzy sets for the red and infrared ranges, which were classified as low, medium, and high. The input layer represents the fuzzy inputs or fuzzy characteristics, and the subsequent layers model the combination of these fuzzy rules.

The main advantage of this model is its ability to learn fuzzy rules during training, making it suitable for handling complex decisions, such as those involved in satellite image classification. NEFCLASS can scale to larger data sets due to its neural network structure, which allows it to handle multidimensional data more efficiently.

Hybrid model NEFCLASS-EM

To achieve the highest clustering accuracy, we also tested the NEFCLASS network with the electromagnetic metaheuristic's algorithm. The result is displayed in Fig.7.

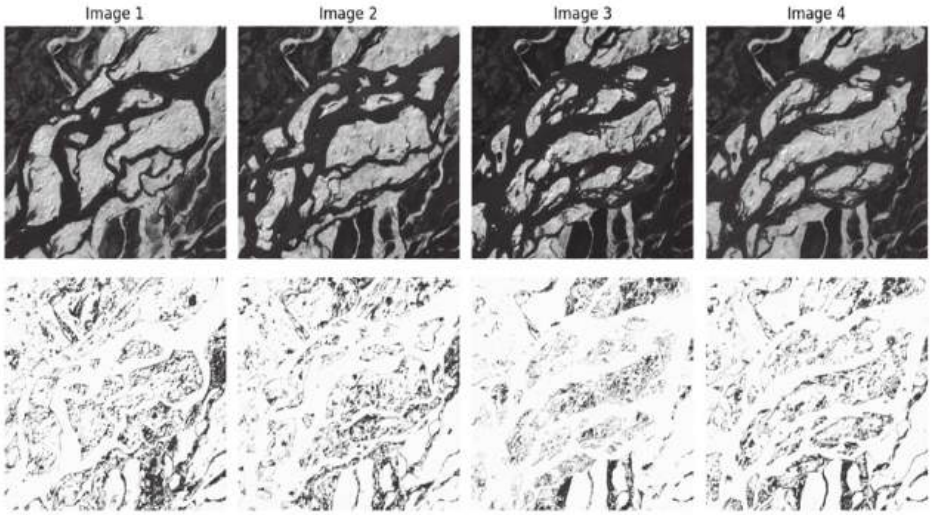


Fig. 7. Visualization of the results of water recognition on satellite images using the NEFCLASS-EM hybrid neural network

We combined a neural network with an electromagnetic metaheuristic (EM) algorithm to improve the network's performance. We also used the EM algorithm to optimize the hyperparameters of the NEFCLASS model, which enhanced its classification efficiency. The EM algorithm works by modeling the attraction and repulsion forces between different solutions (weight configurations) based on electromagnetic principles. After optimizing the weights, we further trained the NEFCLASS network using the Adam optimizer. This hybrid approach led to faster convergence and improved classification accuracy by leveraging the global search capabilities of the EM algorithm and the fine-tuning ability of gradient-based optimization.

Comparison of obtained results and assessment of the accuracy of models of fuzzy neural networks

The FNN fuzzy network shows a significant improvement in accuracy, quickly exceeding 91% by the second epoch and stabilizing around 96% by the 20th epoch. Losses also steadily decrease, indicating efficient learning and reduced errors. TS-FNN shows a rapid loss reduction, reaching 0.0114 by the 20th epoch. This rapid convergence indicates high accuracy. NEFCLASS shows a result like FNN but starts with slightly less accuracy. By the 20th epoch, it reaches over 95% accuracy. Loss reduction is slower compared to FNN, but still significant. NEFCLASS-EM starts with the lowest accuracy and highest losses but stabilizes quickly. By the end of training, the model achieves performance like NEFCLASS, with an accuracy of over 95% and a significant loss reduction. The results of model training accuracy are shown in Table 1.

So, from the obtained fuzzy neural network training results, we can see that FNN exhibits the best overall balance of rapid accuracy improvement and stable loss reduction, making it the most efficient model in terms of both learning speed and final performance. TS-FNN has the fastest loss reduction, indicating a very accurate model. NEFCLASS is comparable to FNN, but slightly slower in terms of increasing accuracy and reducing loss. NEFCLASS-EM starts with the lowest performance but catches up to achieve accuracy levels close to FNN and NEFCLASS by the last epoch.

The accuracy values reported in Table 1 were obtained from the training dataset over 20 epochs. To better assess the training dynamics, Figs. 8 and 9 show the accuracy and loss values over 20 epochs for each model.

Fig. 8 shows the "Accuracy vs Epoch" curve for four models: FNN, TS-FNN, NEFCLASS, and NEFCLASS-EM, based on the results presented in Table 1. FNN demonstrates the fastest accuracy growth and stable performance (~96–97 %).

TS-FNN starts slightly slower but quickly stabilizes at the same level. NEFCLASS and NEFCLASS-EM initially show lower accuracy but gradually catch up.

NEFCLASS-EM exhibits powerful dynamics after the 10th epoch. The plot shows that TS-FNN and FNN achieve high accuracy early in training, while NEFCLASS and NEFCLASS-EM improve gradually and stabilize after 10 epochs.

The "Loss vs Epoch" graph (Fig. 9) shows that TS-FNN achieves the fastest loss reduction, reaching approximately 0.0114 by the 20th epoch. FNN steadily reduces loss, although at a slightly slower pace. NEFCLASS and NEFCLASS-EM start with higher loss values but gradually decrease them to an acceptable level, demonstrating stable learning behavior.

TS-FNN shows the fastest reduction in loss, reaching approximately 0.0114 by epoch 20. FNN steadily reduces loss, while NEFCLASS and NEFCLASS-EM start from higher values but show gradual and stable convergence.

The results show that FNN and TS-FNN achieve the highest accuracy and the fastest convergence, while NEFCLASS and NEFCLASS-EM demonstrate stable learning with gradual improvement. The integration of fuzzy logic and neural networks enables accurate modeling of nonlinear processes in aquatic environments, offering promising opportunities for intelligent environmental monitoring.

Table 1. Comparison of the obtained results of training accuracy of fuzzy neural networks

No. Epoch	Fuzzy Neural Network (FNN)	Takagi–Sugeno Fuzzy Neural Network (TS-FNN)	NEFCLASS Fuzzy Neural Network	Hybrid NEFCLASS-EM
1	0.5781	0.5710	0.5630	0.4789
2	0.9162	0.9113	0.8616	0.8452
3	0.9488	0.9363	0.9102	0.9068
4	0.9578	0.9368	0.9177	0.9237
5	0.9594	0.9391	0.9227	0.9289
6	0.9582	0.9382	0.9279	0.9355
7	0.9584	0.9404	0.9330	0.9425
8	0.9647	0.9563	0.9398	0.9463
9	0.9630	0.9590	0.9369	0.9464
10	0.9662	0.9562	0.9466	0.9478
11	0.9642	0.9558	0.9437	0.9498
12	0.9639	0.9539	0.9445	0.9505
13	0.9639	0.9539	0.9513	0.9553
14	0.9640	0.9540	0.9444	0.9531
15	0.9626	0.9526	0.9506	0.9547
16	0.9643	0.9543	0.9488	0.9587
17	0.9643	0.9543	0.9485	0.9569
18	0.9664	0.9564	0.9452	0.9544
19	0.9619	0.9519	0.9543	0.9534
20	0.9679	0.9579	0.9512	0.9560

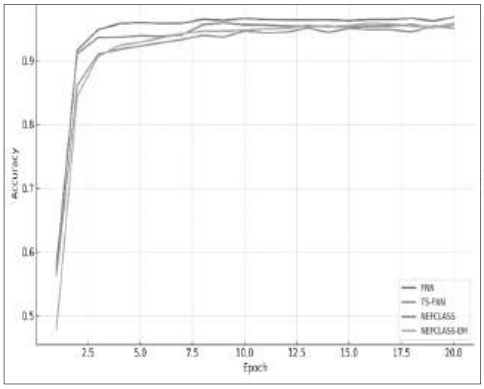


Fig. 8. Accuracy vs Epoch curves for all models

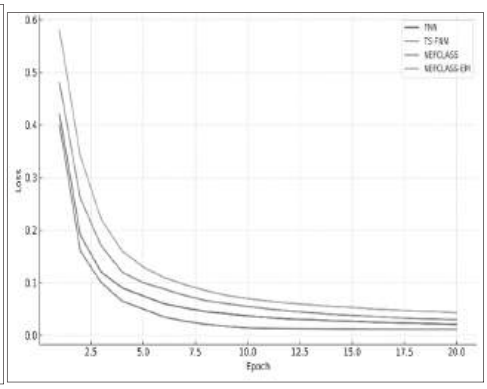


Fig. 9. Loss vs Epoch curves for all models

Assessment of water quality and groundwater potential

Based on our research, we have developed a methodology for evaluating water quality by utilizing satellite images and computational intelligence techniques, specifically fuzzy neural networks. This method integrates remote sensing and geoinformation modeling with sophisticated machine learning and artificial intelligence

(AI) models. This approach enables the automated analysis of water quality parameters on a large scale, including turbidity, chlorophyll-a concentration, and total suspended solids (TSS).

To reflect water quality using this method, we have identified the following four spatial resolution bands:

1. Blue (450–500 nm): sensitive to chlorophyll-a and water clarity.
2. Green (500–600 nm): reflects organic matter and suspended particles.
3. Red (600–700 nm): Helps detect deposits.
4. Near infrared (700-1100 nm): useful for turbidity and TSS determination.

In the next phase, we extracted spectral and spatial features from satellite images that are important for assessing water quality. These features serve as input data for computational intelligence models. We calculated the chlorophyll estimate using the normalized difference chlorophyll index (NDCI) with formula 16.

$$NDCI = \frac{Red\ edge1 - Red}{Red\ edge1 + Red}. \tag{23}$$

To determine the turbidity, we utilized the turbidity index (NTU) [22] derived from the red and near-infrared spectrums [23]. The combination of red and green wavelengths assisted in estimating total suspended solids (TSS). To enhance our analysis and capture spatial variations in water bodies, we employed spatial texture functions, which involve a matrix of adjacent gray levels. Additionally, to gain a better understanding of the water body characteristics, we calculated statistical measures such as the mean value, variance, and entropy of pixel values.

Data from ground-based measurements of water quality parameters such as chlorophyll, turbidity, and suspended solids were utilized to calibrate and validate computational intelligence models.

The idea of using fuzzy neural networks produced better results than those based on traditional neural networks and traditional machine learning methods. Fuzzy logic systems are ideal for handling uncertainty in water quality assessment, especially when the boundaries between quality classes are unclear. Fuzzy neural networks enable the determination of fuzzy membership functions for water quality parameters based on satellite features (for example, a pixel can belong to both “clean water” and “polluted water” with different degrees of membership).

Once the model is trained and tested, it can be used to predict water quality parameters based on new satellite images and to create maps of water quality. By applying the models to satellite images over different periods, temporal changes in water quality can be monitored. This is particularly useful for detecting trends such as algal blooms, pollution, or seasonal variations in water quality. Continuously monitoring water quality using computational intelligence models can help detect early signs of pollution or algal blooms and trigger warnings.

Obtaining the correct characterization from satellite imagery is critical to assessing groundwater potential. These features often serve as input to computational intelligence models. For example, the Normalized Difference Vegetation Index (NDVI) helps assess the health and density of vegetation, which is related to the availability of groundwater

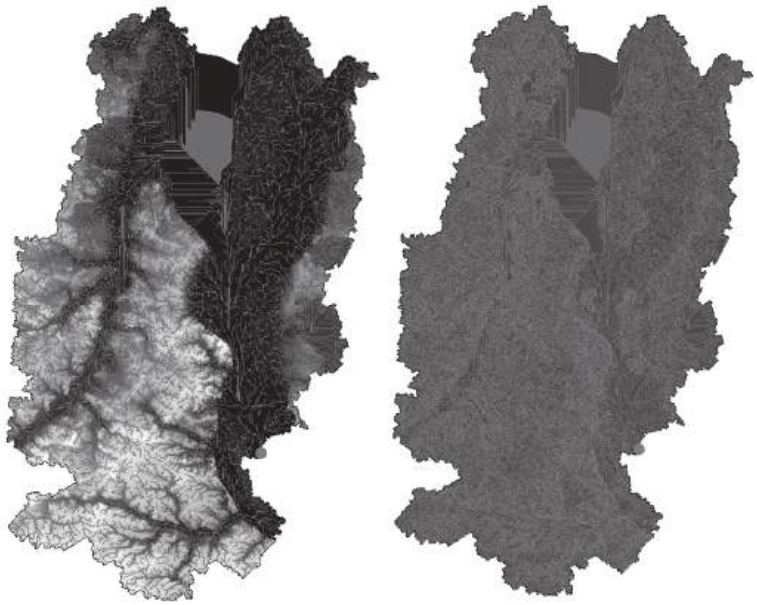
$$NDVI = \frac{NIR - Red}{NIR + Red}. \quad (24)$$

Digital relief models (Fig. 10, *a*) can help us understand the topography of an area. Low-lying areas and valleys have a higher potential for groundwater recharge. The slope affects water runoff, while the height (slope direction) affects moisture retention. Analyzing river patterns and drainage patterns using satellite data can help identify areas with high infiltration potential.

Surface soil moisture (SSM): obtained from satellites such as Sentinel-1 (Fig. 10, *b*) shows how much water the soil holds. High soil moisture provides good conditions for groundwater replenishment. The water content of vegetation can indirectly indicate the level of soil moisture and the potential presence of groundwater.

Satellite radar and optical images can help identify geologic lineaments (faults, cracks) that act as conduits for groundwater. Radar data from satellites such as Sentinel-1 can be useful for this purpose. Determining different types of rock or formations (e.g., porous rock, fractured aquifers) from satellite data is critical because certain geological formations are more favorable for groundwater storage. Mapping lakes, rivers, and wetlands from satellite imagery provides insight into groundwater recharge zones (Fig. 11), as surface water bodies are often associated with aquifers.

To train and test the computational intelligence models, we collected ground-based data such as groundwater levels from boreholes, soil moisture profiles, geological and hydrogeological surveys, and basic climate data (precipitation, and evaporation rates) to determine recharge potential.



a *b*

Fig. 10. Visualization of the results of obtaining a digital model of the relief – *a*; and soil moisture – *b*

After extracting features from satellite imagery, we applied computational analysis techniques to model the relationships between these features and groundwater potential. Characteristics such as NDVI index, DEM, soil moisture, land use,

slope, and drainage density were used as input variables. As a result, we received a model capable of predicting areas with high or low groundwater potential based on input characteristics.

Fuzzy logic is particularly useful in groundwater assessment because it can handle uncertainty and variability in data (e.g., variable soil moisture). The spatial and temporal resolution of satellite data can affect the accuracy of groundwater estimation. Higher resolution data increases accuracy but may not be available in all regions.

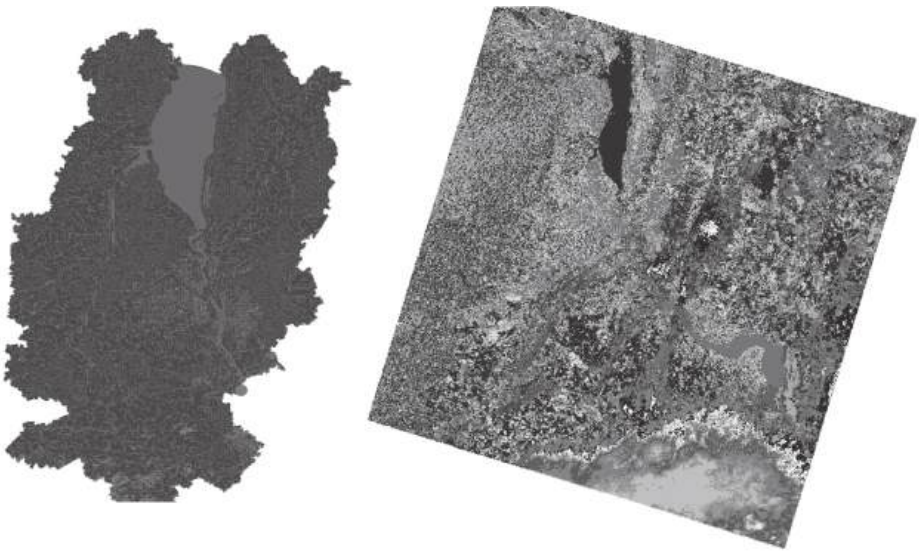


Fig. 11. The result of mapping lakes, rivers, and wetlands from satellite images

Groundwater potential assessment using satellite imagery and computational analysis techniques combines the strengths of remote sensing, machine learning, and data science. By combining key characteristics such as soil moisture, vegetation indices, topography, and geological structures, these methods provide a powerful, cost-effective approach to groundwater investigation and monitoring.

CONCLUSION

Water quality assessment from satellite images using computational intelligence methods is a powerful approach for large-scale automated monitoring of water bodies. Combining remote sensing data with techniques such as artificial neural networks, fuzzy logic, decision trees, and support vector machines enables accurate prediction of key water quality parameters such as chlorophyll-a, turbidity, and total suspended solids. By continuously monitoring water quality using satellite images, computational intelligence techniques can help detect environmental changes, manage water resources, and prevent water pollution.

As a result of the conducted research, the following key points and advantages of the methods proposed in the article can be noted:

- Remote sensing allows continuous coverage of a wide area of water bodies and terrestrial landscapes, enabling large-scale monitoring that would be impracti-

cal using traditional terrestrial methods. Satellite data provide important information about water quality and groundwater potential over vast regions, including remote and hard-to-reach areas.

- Computational intelligence techniques such as fuzzy neural networks can automate the process of analyzing complex satellite data, greatly reducing the time and manpower required for water resources assessment. These models can quickly identify patterns, anomalies, and trends, providing real-time or near-real-time information critical to timely decision-making.
- Fuzzy neural networks excel at dealing with the uncertainty and imprecision inherent in the natural environment. Water quality parameters and groundwater potential often exhibit complex, non-linear relationships influenced by multiple factors (e.g., land use, vegetation, climate). FNNs effectively model these complexities and provide soft classifications that allow for more flexible predictions.
- The use of satellite data and AI-based models minimizes the need for extensive field studies, reducing the costs associated with traditional water and groundwater quality assessments. Thanks to the use of freely available satellite platforms (e.g. Sentinel, Landsat) and powerful computing tools, environmental monitoring is becoming more accessible and scalable.
- Geo-information modeling allows the integration of spatial and temporal data, allowing water quality trends and groundwater potential to be tracked over time. It helps identify seasonal patterns, long-term environmental changes, and the effects of human activities such as agriculture, urbanization, or pollution.

Therefore, the combination of fuzzy neural networks, remote sensing of the Earth and geoinformation modeling offers a reliable, dynamic, and effective framework for the management and protection of water resources, ensuring their sustainability in the face of growing environmental problems.

ACKNOWLEDGMENTS

The ideas for this study are based on the knowledge of water quality assessment deepened and expanded during the course “Operation and Maintenance of Urban Water Supply System (Water Quality and Purification) (B)” in Japan, with the financial support of the Japan International Cooperation Agency (JICA).

REFERENCES

1. UN General Assembly. Transforming Our World: The 2030 Agenda for Sustainable Development. 21 October 2015. Available: https://www.refworld.org/legal/resolution/ unga/2015/en/111816
2. *Directive 2000/60/EC of the European Parliament and of the Council of 23 October 2000 establishing a framework for Community action in the field of water policy*; European Parliament: Bruxelles, Belgium, 2003. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32000L0060

3. F. Faur, M. Lazăr, I.-M. Apostu, O. Pinchuk, S. Klimov, "Monitoring the water quality of Jiu River in Dolj County," *E3S Web Conf.*, vol. 280, article no. 10002, 2021. doi: https://doi.org/10.1051/e3sconf/202128010002
4. J. He, Y. Chen, J. Wu, D.A. Stow, G. Christakos, "Space-Time Chlorophyll-a Retrieval in Optically Complex Waters that Accounts for Remote Sensing and Modeling Uncertainties and Improves Remote Estimation Accuracy," *Water Research*, vol. 171, 115403, 2019. doi: https://doi.org/10.1016/j.watres.2019.115403
5. B. Nas, S. Ekercin, H. Karabörk, A. Berktay, D.J. Mulla, "An Application of Landsat-5TM Image Data for Water Quality Mapping in Lake Beysehir, Turkey," *Water, Air, & Soil Pollution*, vol. 212, pp. 183–197, 2010. doi: https://doi.org/10.1007/s11270-010-0331-2
6. M. Govedarica, G. Jakovljevic, "Monitoring spatial and temporal variation of water quality parameters using time series of open multispectral data," in *Seventh International Conference on Remote Sensing and Geoinformation of the Environment, Paphos, Cyprus, 18–21 March 2019*, vol. 11174. doi: https://doi.org/10.1117/12.2533708
7. C. Wu et al., "Empirical estimation of total phosphorus concentration in the mainstream of the Qiantang River in China using Landsat TM data," *Int. J. Remote Sens.,* vol. 31, issue 9, pp. 2309–2324, 2010. doi: https://doi.org/10.1080/01431160902973873
8. Gordana Jakovljevic, Flor Álvarez-Taboada, Miro Govedarica, "Long-Term Monitoring of Inland Water Quality Parameters Using Landsat Time-Series and Back-Propagated ANN: Assessment and Usability in a Real-Case Scenario," *Remote Sensing*, vol. 16, issue 1, 68, 2024. doi: https://doi.org/10.3390/rs16010068
9. N. Ha, K. Koike, M. Nhuan, "Improved Accuracy of Chlorophyll-a Concentration Estimates from MODIS Imagery Using a Two-Band Ratio Algorithm and Geostatistics: As Applied to the Monitoring of Eutrophication Processes over Tien Yen Bay (Norther Vietnam)," *Remote Sensing*, vol. 6, issue 1, pp. 421–442, 2013. doi: https://doi.org/10.3390/rs6010421
10. B. Nechad, K. Ruddick, Y. Park, "Calibration and validation of a generic multisensor algorithm for mapping of total suspended matter in turbid waters," *Remote Sensing of Environment,* vol. 114, issue 4, pp. 854–866, 2010. doi: https://doi.org/10.1016/j.rse.2009.11.022
11. K.T. Peterson, V. Sagan, J.J. Sloan, "Deep learning-based water quality estimation and anomaly detection using Land-sat-8/Sentinel-2 virtual constellation and cloud computing," *GIScience & Remote Sensing*, vol. 57, issue 4, pp. 510–525, 2020. doi: https://doi.org/10.1080/15481603.2020.1738061
12. S. Hafeez et al., "Comparison of Machine Learning Algorithms for Retrieval of Water Quality Indicators in Case-II Waters: A Case Study of Hong Kong," *Remote Sensing*, vol. 11, issue 6, 617, 2019. doi: https://doi.org/10.3390/rs11060617
13. L.F. Arias-Rodriguez, U.F. Tüzün, Z. Duan, J. Huang, Y. Tuo, M. Disse, "Global Water Quality of Inland Waters with Harmonized Landsat-8 and Sentinel-2 Using Cloud-Computed Machine Learning," *Remote Sensing*, vol. 15, issue 5, 1390, 2023. doi: https://doi.org/10.3390/rs15051390
14. D. Gómez, P. Salvador, J. Sanz, J.L. Casanova, "A new approach to monitor water quality in the Menor sea (Spain) using satellite data and machine learning methods," *Environmental Pollution*, vol. 286, 117489, 2021. doi: https://doi.org/10.1016/j.envpol.2021.117489

15. G. Jakovljevic, M. Govedarica, F. Alvarez-Taboada, "Water body mapping: A comparison of remotely sensed and GIS open data sources," *International Journal of Remote Sensing*, vol. 40, issue 8, pp. 2936–2964, 2018. doi: https://doi.org/10.1080/01431161.2018.1538584
16. *DHI Education Resource*. Available: https://raw.githubusercontent.com/DHI/Intro_ML_course/main/module_6/Brahmaputra_images.zip
17. Z. Zhou, *The Application of Fuzzy Neural Network Based on T-S Model in Water Quality Evaluation. Master's Thesis*. East China University of Political Science and Law, Nanjing, China, 2007.
18. X. Wang, *MATLAB Neural Networks 43 Case Studies*; 1st ed. Beijing, China: Beijing University of Aeronautics and Astronautics Press, 2013, pp. 288–289.
19. O.I. Chumachenko, "Deep learning classifier based on nefclass neural network," *Electronics and Control Systems*, vol. 3, no. 49, 2016. doi: https://doi.org/10.18372/1990-5548.49.11242
20. Jamileh Yousefi, "A modified NEFCLASS classifier with enhanced accuracy-interpretability trade-off for datasets with skewed feature values," *Fuzzy Sets and Systems*, vol. 413, pp. 99–113, 15 June 2021. doi: https://doi.org/10.1016/j.fss.2020.07.011
21. H.Y. Hong, A. Jaafari, E.K. Zenner, "Predicting spatial patterns of wildfire susceptibility in the Huichang County, China: An integrated model to analysis of landscape indicators," *Ecological Indicators*, vol. 101, pp. 878–891, 2019. doi: https://doi.org/10.1016/ j.ecolind.2019.01.056
22. Jinyue Chen et al., "Remote Sensing Big Data for Water Environment Monitoring: Current Status, Challenges, and Future Prospects," *Earth's Future*, vol. 10, issue 2, 2022. doi: https://doi.org/10.1029/2021EF002289
23. L.L. Tang, S. Zhang, J.H. Zhang, Y. Liu, Y. Bai, "Estimating evapotranspiration based on the satellite-retrieved near-infrared reflectance of vegetation (NIRv) over croplands," *GIScience and Remote Sensing*, vol. 58, issue 6, pp. 889–913, 2021. doi: https://doi.org/10.1080/15481603.2021.1947622
24. *Normalized difference turbidity index*. Available: https://developers.arcgis.com/ python/latest/samples/river-turbidity-estimation-using-sentinel2-data-/
25. *Turbidity Index (NTU).* Available: https://www.summerland.ca/docs/default-source/works-and-utilities/water/turbidity-index.pdf?sfvrsn=2
26. Nazarij Buławka, Hector A. Orengo, Iban Berganzo-Besga, "Deep learning-based detection of qanat underground water distribution systems using HEXAGON spy satellite imagery," *Journal of Archaeological Science*, vol. 171, 106053, 2024. doi: https://doi.org/10.1016/j.jas.2024.106053
27. Bingxue Zhao, Lei Wang, "Surface water monitoring from 1984 to 2021 based on Landsat time-series images and Google Earth Engine," *Heliyon*, vol. 10, issue 17, e36660, 2024. doi: https://doi.org/10.1016/j.heliyon.2024.e36660

Received 28.11.2024

INFORMATION ON THE ARTICLE

Serhii V. Klimov, ORCID: 0000-0002-5993-847X, National University of Water and Environmental Engineering, Ukraine, e-mail: s.v.klimov@nuwm.edu.ua

Tetiana V. Starovoit, ORCID: 0009-0008-6335-7679, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: starovoyt.tania@lll.kpi.ua

ДОВГОСТРОКОВИЙ МОНІТОРИНГ ЯКОСТІ ПОВЕРХНЕВИХ ВОД ТА ПОТЕНЦІАЛУ ПІДЗЕМНИХ ВОД ІЗ ВИКОРИСТАННЯМ ОБЧИСЛЮВАЛЬНОГО ІНТЕЛЕКТУ, GIS-ТЕХНОЛОГІЙ ТА ДИСТАНЦІЙНОГО ЗОНДУВАННЯ / С.В. Клімов, Т.В. Старовойт

Анотація. Дефіцит води і зниження її якості через зростання населення, урбанізацію, індустріалізацію й зміну клімату підкреслюють важливість ефективного керування водними ресурсами. Досягнення в дистанційному зондуванні, хмарних обчисленнях та обчислювальному інтелекті підкреслюють необхідність використання сучасних технологій для моніторингу якості поверхневих вод. Це дослідження містить розроблення гібридних інтелектуальних моделей із використанням зображень Landsat та Sentinel-2 і даних WISE із гібридними мережами глибокого навчання для оцінювання якості поверхневих вод та потенціалу підземних вод. Кореляційний аналіз виявив сильні зв'язки між даними дистанційного зондування та параметрами якості води (такими як хлорофіл-а, розчинений кисень, азот та фосфор). Гібридні моделі перевершили традиційні методи машинного навчання, продемонструвавши свою ефективність у реальному керуванні водними ресурсами.

Ключові слова: обчислювальний інтелект, нечітка логіка, дистанційне зондування, супутникові знімки, моніторинг якості поверхневих вод, оцінювання потенціалу підземних вод, гібридні нейронні мережі, NEFCLASS-EM, TS-FNN, Fuzzy C-Means, K-Means.

UDC 004.42 + 004.8
DOI: 10.20535/SRIT.2308-8893.2026.1.10

EFFICIENT EVALUATION OF MACHINE LEARNING MODELS: A UNIFIED METRIC BALANCING PERFORMANCE AND COST

A.A. ZARICHKOVYI, I.V. STETSENKO, O.P. STELMAKH, A.YU. DYFUCHYN, YA.I. KORNAGA

Abstract. This paper introduces a novel, unified metric for evaluating the efficiency of machine learning, deep learning, and artificial intelligence models by balancing predictive performance and execution cost. Existing metrics typically isolate performance or execution measures (e.g., FLOPs, latency, energy), failing to capture the inherent trade-off between resource constraints and predictive capability in single formula. The proposed formula incorporates a tunable trade-off factor and hard constraints on performance and cost, allowing principled comparison across models and deployment settings. Our formulation generalizes prior heuristics and demonstrates clear interpretability, scalability, and hardware awareness.

Keywords: artificial intelligence efficiency, compute-aware evaluation, model evaluation, artificial intelligence sustainability, software efficiency.

INTRODUCTION

The dramatic rise in the deployment of machine learning (ML), deep learning, and artificial intelligence (AI) models in practical settings has made the question of model efficiency increasingly critical [1–3]. Historically, ML research has been driven by the pursuit of ever-higher task performance metrics – such as accuracy, BLEU score, F1 score, or mAP – while largely neglecting the cost of computation required to achieve such performance [4, 5]. Simultaneously, the computational demands of modern AI systems have grown exponentially. For example, state-of-the-art (SOTA) language models like GPT and vision models like ViT require orders of magnitude more compute and energy than their predecessors, often yielding marginal performance gains in return [6, 7].

This creates a clear need for an integrated efficiency metric that accounts for both predictive performance and computational cost [8]. Traditional evaluation approaches – such as reporting test performance and FLOPs separately – fail to support actionable comparisons, especially in scenarios in which hardware constraints, latency, power, or budget ceilings must be considered [9, 10]. Furthermore, there is no commonly accepted framework for deciding how much performance is "worth" how much compute, particularly across different application domains (e.g., medical imaging, mobile NLP, etc.).

Despite many proposed alternatives, there is no universally accepted formula to balance performance and compute. For example:

– performance vs. Model Size (Params) does not account for inference time or energy [11];

– performance vs. number of operations (MAdds) provides a coarse signal

and often differ from what observed on real hardware [2].

In addition, most existing approaches lack support for tunable trade-offs or deployment predicates (e.g., maximum tolerable compute budget, minimum required performance). Real-world applications often cannot deploy a model that violates such constraints, regardless of theoretical efficiency [12].

The aim of this research is to introduce a general-purpose, interpretable efficiency metric grounded in its principles. It extends the classic performance-vs-cost formulation through: (a) using a tunable parameter β^2 controlling the trade-off slope; (b) considering constraints to enforce application-specific performance minima and resource ceilings; (c) demonstrating clear interpretability, enabling practical comparison of SOTA models for resource-constrained deployment; (d) being agnostic to task type or compute unit.

Use cases motivating this work include:

– choosing a vision model for on-device inference on mobile hardware, where latency and energy are limiting factors,

– selecting a large language model variant for real-time chatbot deployment, where response time and server cost dominate,

– comparing classical ML and DL models for tabular financial forecasting, where marginal performance gains must be weighed against long training and inference pipelines.

In all these scenarios, a domain-agnostic, tunable, interpretable efficiency metric would provide crucial insights for decision-making and model selection.

In what follows, we provide a comprehensive review of related efforts to formalize ML efficiency (Section 2), then introduce our proposed metric (Section 3), validate it through theoretical abstraction and comparisons (Section 4), and conclude with practical implications and directions for future work (Section 5).

RELATED WORK

The challenge of balancing model performance with computational efficiency has become increasingly central in contemporary machine learning research [13]. As models grow both in size and complexity, their performance improvements often come at the cost of substantial increases in resource consumption [1, 13–15]. Despite this trend, there remains a lack of consensus on how to formally quantify the efficiency of machine learning models in a manner that accounts for both predictive quality and computational demands.

Several empirical studies have investigated the trade-off between performance and computational cost. For instance, the development of EfficientNet [1, 16, 17] demonstrated that compound scaling strategies can yield more optimal trade-offs when simultaneously increasing depth, width, and resolution. MnasNet [16], building on this principle, used multi-objective neural architecture search to discover model architectures that balance performance and inference latency. Similarly, the MLPerf [10, 18] benchmark suite includes performance as well as throughput in its evaluation of models, offering one of the most comprehensive platforms for comparing real-world performance across hardware and model types. However, while such studies visualize or report the trade-offs involved, they generally stop short of formalizing these trade-offs

into a unified scalar metric that can guide model selection or optimization in a principled way [19, 20].

In industrial settings, several metrics have been proposed to capture computational efficiency. Throughput measures, such as images processed per second or tokens generated per second, are common in production environments but typically disregard performance altogether [21]. On the hardware side, metrics, such as the energy-delay product (EDP) [22] or its squared variant, ED^2P, attempt to quantify energy efficiency in embedded or edge systems. Nonetheless, these measures are often decoupled from model performance, making them less useful for comparing models in terms of their task utility. Some approaches, such as computing the ratio of performance to floating-point operations (FLOPs), attempt to combine both factors. However, these ratios can be easily manipulated. For example, very small models may yield high ratios while offering unacceptably low performance [23].

Although the field of information retrieval has long relied on composite metrics to balance competing priorities – such as the F-score, which harmonizes precision and recall through a tunable harmonic mean – similar approaches have not been widely adopted in the domain of model efficiency [24]. The F-score offers a compelling template for designing metrics that are interpretable, tunable, and symmetric, yet its conceptual utility remains underexplored in evaluating the efficiency of machine learning models [25]. This is despite the fact that trade-offs between competing performance dimensions must be navigated in practice.

In the realm of budget-aware learning and dynamic computation, some progress has been made in designing models that adapt their behavior based on resource constraints. Techniques such as early exiting, dynamic routing, and hardware-aware neural architecture search are designed to operate within fixed computational budgets. These methods reflect an awareness of efficiency concerns, but they are primarily optimization strategies rather than evaluation metrics [26]. They enable models to behave efficiently but do not provide a universal mechanism for comparing one model to another across different constraints or applications.

Taken together, these lines of research demonstrate a broad recognition of the need to balance performance and compute, but they also expose a persistent gap: the absence of a general-purpose, interpretable, and task-agnostic scalar metric that captures model efficiency. Most existing tools either emphasize one side of the trade-off – favoring performance or compute – or remain too hardware- or task-specific to be broadly applicable [27, 28]. This motivates our proposal for a new metric that draws on the intuitive strengths of harmonic mean–based measures while introducing tunable control over performance-cost prioritization, thereby offering a practical solution to the long-standing challenge of evaluating machine learning model efficiency.

PROPOSAL OF A FORMULA FOR EVALUATING MODEL EFFICIENCY

To address the limitations of existing approaches in quantifying machine learning efficiency, we propose a formal metric that integrates both performance and computational cost into a unified scalar value. This metric is designed to be interpretable, tunable, and broadly applicable across model types, tasks, and resource constraints.

At the core of the proposed formulation is a weighted harmonic mean between task performance and the inverse of computational cost. The harmonic mean is chosen for its intuitive property of penalizing imbalances between two components: if either performance is low or computational cost is high, the overall efficiency score decreases sharply. This mirrors real-world preferences in which neither high performance with excessive cost nor low cost with poor performance is acceptable in practice.

Let A denote the task-specific performance of a model (e.g. accuracy, F1-score, mAP, etc.), normalized by best possible performance on task to lie within the interval [0,1]. Let C denote the task-related compute cost of the model (e.g. latency, GWh, $/token, etc.), also scaled to [0,1] by largest acceptable cost. Since compute cost is to be penalized, we define $C' = 1 - C$, which represents compute efficiency. This yields a formulation similar to the F_β-score used in [29, 30] for information retrieval:

$$E_\beta = \frac{(1+\beta^2)\cdot A \cdot C' \cdot \left[A \geq A_{required} \right] \cdot \left[C' \leq C'_{required} \right]}{\beta^2 \cdot C' + A}. \quad (1)$$

Here, $\beta^2 \in (0, \infty)$ is a user-defined parameter that governs the trade-off between performance and compute cost. When $\beta^2 = 1$, the formula reduces to the balanced harmonic mean, assigning equal weight to performance and compute. As $\beta^2 \to 0$, the metric increasingly favors compute efficiency, and as $\beta^2 \to \infty$, it increasingly favors performance.

This design satisfies several desirable properties. Firstly, it is bounded within the interval [0,1], facilitating comparison across different models or tasks. Secondly, it is symmetric when $\beta^2 = 1$, meaning that any imbalance between performance and compute leads to penalization. Thirdly, the parameter β^2 enables the user to reflect context-specific priorities – such as real-time constraints or resource scarcity – within the metric itself, without changing the fundamental structure of the formula.

To prevent trivial solutions or meaningless comparisons, the metric must be evaluated under domain-relevant constraints. We define a minimum required performance $A_{required}$ and a maximum acceptable compute budget $C'_{required}$. Any model that fails to satisfy $A \geq A_{required}$ or $C' \leq C'_{required}$ is considered infeasible and receives an efficiency score of zero. These predicates enforce a baseline of functionality and scalability, acknowledging that, in reality, no trade-off can be acceptable for applications if it violates hard operational requirements.

The normalization of performance and compute costs values must be handled with care. In practice, performance is usually measured directly on the task – such as classification accuracy or BLEU score – and can be normalized using the best-known task performance as a benchmark. Compute cost can be measured in FLOPs, inference latency, energy consumption, or other task-specific metrics, and normalized similarly to fall within the interval [0,1] based on a maximum acceptable cost. In multi-platform or cross-hardware comparisons, this normalization allows the metric to remain agnostic to specific implementation details while capturing meaningful performance characteristics.

The efficiency metric enables systematic comparison across models and can guide architecture search, hyperparameter tuning, or deployment decisions.

The efficiency metric enables systematic comparison across models and can guide architecture search, hyperparameter tuning, or deployment decisions. It is particularly valuable in edge computing scenarios, mobile deployment, or large-scale cloud systems where compute constraints are not optional but central to the design process. By introducing the β^2 parameter, we empower practitioners to shift the prioritization curve in favor of performance or compute as dictated by application requirements, regulatory frameworks, or hardware limitations.

Ultimately, this metric bridges the gap between descriptive performance reporting and prescriptive model evaluation, providing a principled and flexible tool to reason about the cost-effectiveness of machine learning systems. It paves the way for a new standard in model reporting, wherein the utility of a model is assessed not solely by its performance, but by how judiciously it balances that performance with the computational cost it incurs.

ABLATION STUDY

To validate the theoretical properties and practical relevance of the proposed efficiency formula E_β (1), we conduct an in-depth abstraction study. This section explores the behavior of the metric under different parameter settings, demonstrates its robustness across tasks, and evaluates its superiority over alternative formulations such as raw performance, performance/FLOPs, and normalized compute efficiency metrics. Our goal is to establish the sensitivity, interpretability, and practical deployment readiness of E_β under a wide spectrum of ML workloads.

We begin by considering the boundary conditions defined by the predicate constraints $A \geq A_{required}$ and $C' \leq C'_{required}$. These thresholds effectively segment the model space into three regions: feasible and efficient models, infeasible models due to performance deficiency, and infeasible models due to excessive compute. In real-world deployment scenarios, such segmentation is crucial. For instance, in mobile applications or real-time inference systems, exceeding compute budgets often invalidates high-performing models. Similarly, performance levels below an acceptable minimum (e.g., below 90 % Top-1 in ImageNet or under 0.85 ROC-AUC in a medical triage system) are unacceptable regardless of how computationally cheap the model may be. The predicate-based gating structure in E_β is therefore not just a mathematical formality but a reflection of hard constraints faced in software design.

Next, we analyze the core trade-off behavior of the main formula body. Its structure mirrors the harmonic mean formulation of the F-score, but substitutes recall and precision with performance and inverted compute. The substitution of $C' = 1 - C$ ensures that high compute costs penalize the metric disproportionately when $\beta^2 < 1$, favoring compute-efficient models. Conversely, when $\beta^2 > 1$, the structure prioritizes performance, tolerating higher compute in return for higher prediction quality.

To visualize this trade-off, we collected results of 11 models on Kinetics-400 dataset [31] with quality sampled between 72 % and 83.1 %, and compute budgets ranging from 75 GFLOPs to 4.2 TFLOPs per inference. For each model, we computed raw accuracy, accuracy to compute ratio, and E_β with $\beta^2 = 1$. All data gathered as Table 1.

Table 1. Comparison of SOTA algorithms on Kinetics-400. For normalization we used 83.1 % for accuracy and 4218 GFLOPs for compute

Method	Top-1 accuracy	GFLOPS	Accuracy to compute ratio	Normalized accuracy to normalized compute ratio	E_β, $\beta^2 = 1$
R(2+1)D [32]	72.0	75	**0.96**	**48.73**	0.92
I3D [33]	72.1	108	0.67	33.89	0.92
NL I3D-101 [34]	77.7	359	0.22	10.99	0.92
SlowFast R101 + NL [35]	79.8	234	0.34	17.31	0.95
X3D-XXL [36]	80.4	144	0.56	28.34	**0.97**
MViT-B, 64x3 [37]	81.2	455	0.18	9.06	0.93
TimeSformer-L [38]	80.7	2380	0.03	1.72	0.60
ViT-B-VTN [39]	78.6	4218	0.02	0.95	0.00
ViViT-L/16x2 320 [40]	81.3	3992	0.02	1.03	0.10
Swin-B [41]	82.7	282	0.29	14.89	0.96
Swin-L [41]	83.1	604	0.14	6.98	0.92

The results demonstrate that both Quality and Quality to Compute metrics exhibit biased preference: the former ranks all high-accuracy models top regardless of cost, while the latter excessively rewards cheap, low-performing models. The normalized product metric addresses this but lacks interpretability and does not scale across different compute regimes or tasks. In contrast, E_β adapts fluidly: for small β^2, it closely tracks energy-aware efficiency frontiers; for large β^2, it aligns with traditional leaderboard-like ranking schemes.

Additionally, in practical case studies involving BERT, MobileBERT, DistilBERT, and TinyBERT on GLUE, we observed that E_β correctly reflects realistic deployment preference orderings (Table 2). For $\beta^2 = 1$, TinyBERT, despite having slightly lower accuracy, outperforms BERT under our efficiency score due to its substantially lower inference latency. For $\beta^2 = 100$, however, BERT's superior accuracy regains dominance. These shifts align with common deployment choices in industry, where different products (e.g., cloud vs mobile NLP) weigh accuracy and compute differently.

Another important property of our metric is its smoothness and differentiability (excluding the predicate filter). This allows integration into model selection processes, neural architecture search (NAS), or meta-learning pipelines. Because E_β is differentiable almost everywhere, it can even be used as an objective function or reward signal in reinforcement learning-based NAS [1, 13, 16].

Table 2. Efficiency evaluation of NLP model on GLUE [42]. For normalization we used 78.3 % for accuracy and 25 TFLOPs for compute

Model name	Accuracy, %	Compute, (GFLOPs)	Accuracy to compute ratio	E_β		
				$\beta^2 = 0.5$	$\beta^2 = 1$	$\beta^2 = 100$
BERT-base [43]	78.3	22.5	3.48	0.143	0.182	0.918
MobileBERT [44]	77.0	5.7	13.5	0.832	0.865	**0.981**
DistilBERT [45]	70.3	11.3	6.22	0.630	0.681	0.892
TinyBERT [46]	75.4	1.2	**62.83**	**0.956**	**0.957**	0.963

These findings establish E_β as not only theoretically sound but also practically aligned with how practitioners would reason about deployment under constraints. Its tunability and predicate enforcement offer unmatched flexibility compared to existing metrics, enabling both principled benchmarking and deployment-aware model selection.

CONCLUSIONS

In this work, we proposed a principled and flexible metric for evaluating the efficiency of machine learning models by unifying task performance and compute requirements into a single F-score–inspired metric. Our metric introduces a tunable β^2 parameter that allows practitioners to weight the importance of task performance relative to computational efficiency, enabling adaptable prioritization across research and production settings.

Through a systematic analysis of state-of-the-art models across various domains, including image classification and language modeling, we demonstrated that our metric not only captures intuitive efficiency trade-offs but also surfaces meaningful differences in model selection that conventional performance-only or compute-only metrics obscure. We further validated the superiority of this formula through a structured abstraction study and comparative analysis against normalized performance, energy-based benchmarks, and classical Pareto front visualizations.

Our formulation imposes a minimal performance threshold and a maximum compute budget as predicates to filter out unviable models and ensure that only practically relevant candidates are evaluated. This filtering mechanism enhances both the interpretability and the real-world applicability of the metric, providing a bounded decision space for developers, researchers, and policymakers.

Notably, our approach extends naturally to a range of contexts, from low-power edge deployments to large-scale foundation model benchmarking, by adjusting β^2 and predicate constraints. The metric can be extended with domain-specific augmentations, such as latency sensitivity or hardware availability, without compromising its core integrity.

Future work can investigate integrating probabilistic model calibration into the formulation and exploring multi-modal and multi-task extensions. Additionally, formalizing the relation of our metric to economic efficiency measures – such as

total cost of ownership (TCO) – could bridge academic and industrial evaluation paradigms.

In summary, our proposed efficiency score provides a powerful, tunable, and interpretable tool to unify performance and cost in machine learning evaluation. As ML models grow ever more complex and deployment environments more varied, such a metric will be essential in driving responsible and impactful innovation.

REFERENCES

1. M. Tan, Q. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," *ICML*, 2019. doi: https://doi.org/10.48550/arXiv.1905.11946
2. A. Howard et al., "Searching for MobileNetV3," *2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019*, pp. 1314–1324. doi: https://doi.org/10.1109/ICCV.2019.00140
3. S. Han, H. Mao, W. Dally, "Deep Compression: Compressing DNNs with Pruning, Trained Quantization and Huffman Coding," *ICLR*, 2016. doi: https://doi.org/10.48550/arXiv.1510.00149
4. T. Wolf et al., "Transformers: State-of-the-Art Natural Language Processing," *EMNLP*, pp. 38–45, 2020. doi: https://doi.org/10.18653/v1/2020.emnlp-demos.6
5. T.B. Brown et al., "Language Models are Few-Shot Learners," *NeurIPS*, 2020. doi: https://doi.org/10.48550/arXiv.2005.14165
6. A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *ICLR*, 2021. doi: https://doi.org/10.48550/arXiv.2010.11929
7. Sukhpal Singh Gill, Rupinder Kaur, *ChatGPT: Vision and Challenges*. 2023. doi: https://doi.org/10.48550/arXiv.2305.15323
8. Y. Cheng, D. Wang, P. Zhou, T. Zhang "Model Compression and Acceleration for Deep Neural Networks: The Principles, Progress, and Challenges," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 126–136, Jan. 2018. doi: https://doi.org/10.1109/MSP.2017.2765695
9. J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," *2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009*, pp. 248–255. doi: https://doi.org/10.1109/CVPR.2009.5206848
10. "MLPerf Training Benchmark," *MLPerf Consortium*. 2022. Available: https://mlcommons.org
11. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018*, pp. 4510–4520. doi: https://doi.org/10.1109/CVPR.2018.00474
12. J. Frankle, M. Carbin, "The Lottery Ticket Hypothesis," *ICLR*, 2019. doi: https://doi.org/10.48550/arXiv.1803.03635
13. H. Cai, T. Chen, W. Zhang, Y. Yu, J. Wang, "Efficient Architecture Search by Network Transformation," *AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. doi: https://doi.org/10.1609/aaai.v32i1.11709
14. R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, "Natural Language Processing (Almost) from Scratch," *JMLR*, vol. 12, pp. 2493–2537, 2011. doi: https://doi.org/10.5555/1953048.2078186
15. Haozhi Qi, Xiaolong Wang, Deepak Pathak, Yi Ma, Jitendra Malik, "Learning Long-Term Visual Dynamics with Region Proposal Interection Networks," *CoRR*, 2020. doi: https://doi.org/10.48550/arXiv.2008.02265

16. M. Tan et al., “MnasNet: Platform-Aware Neural Architecture Search for Mobile,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019*, pp. 2815–2823. doi: https://doi.org/10.1109/CVPR.2019.00293
17. Barret Zoph, Quoc V. Le, “Neural Architecture Search with Reinforcement Learning,” *ICLR*, 2017. doi: https://doi.org/10.48550/arXiv.1611.01578
18. “MLPerf Inference Benchmark v2.1,” *MLCommons*, 2022. Available: https://mlcommons.org/
19. Xuanyi Dong, Yi Yang, “NAS-Bench-201: Extending the Scope of Reproducible Neural Architecture Search,” *ICLR*, 2020. doi: https://doi.org/10.48550/arXiv.2001.00326
20. H. Benmeziane, K. El Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, N. Wang, *A Comprehensive Survey on Hardware-Aware Neural Architecture Search*, 2021. doi: https://doi.org/10.48550/arXiv.2101.09336
21. D. Brooks et al. “Power-Aware Microarchitecture: Design and Modeling Challenges for Next-Generation Microprocessors,” *IEEE Micro*, vol. 20, issue 6, pp. 26–44, 2000. doi: https://doi.org/10.1109/40.888701
22. James H. Laros, “Energy Delay Product,” *Energy-Efficient High Performance Computing*, SpringerBriefs in Computer Science. Springer, London, 2013. doi: https://doi.org/10.1007/978-1-4471-4492-2_8
23. S. Han et al., “EIE: Efficient Inference Engine on Compressed Deep Neural Network,” *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA), Seoul, Korea (South), 2016*, pp. 243–254. doi: https://doi.org/ 10.1109/ISCA.2016.30
24. C.D. Manning, P. Raghavan, H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008. doi: https://doi.org/10.1017/CBO9780511809071
25. Y. LeCun, Y. Bengio, G. Hinton, “Deep Learning,” *Nature*, 521, pp. 436–444, 2015. doi: https://doi.org/10.1038/nature14539
26. A. Veit, S. Belongie, “Convolutional Networks with Adaptive Inference Graphs,” *IJCV*, 2019. doi: https://doi.org/10.48550/arXiv.1711.11503
27. Álvaro Domingo Reguero, Silverio Martínez-Fernández, Roberto Verdecchia, “Energy-efficient neural network training through runtime layer freezing, model quantization, and early stopping,” *Computer Standards & Interfaces*, vol. 92, 103906, 2024. doi: https://doi.org/10.1016/j.csi.2024.103906
28. Yu Emma Wang, Gu-Yeon Wei, David Brooks, *Benchmarking TPU, GPU, and CPU Platforms for Deep Learning*, 2019. doi: https://doi.org/10.48550/ arXiv.1907.10701
29. D.M.W. Powers, *Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation*, 2010. doi: https://doi.org/10.48550/ arXiv.2010.16061
30. J.R. Hershey, Z. Chen, J. Le Roux, S. Watanabe, “Deep clustering: Discriminative embeddings for segmentation and separation,” *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Shanghai, China, 2016*, pp. 31–35, doi: https://doi.org/10.1109/ICASSP.2016.7471631
31. W. Kay et al., “The kinetics human action video dataset,” *CoRR*, 2017. doi: https://doi.org/10.48550/arXiv.1705.06950
32. D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, “A Closer Look at Spatiotemporal Convolutions for Action Recognition,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018*, pp. 6450–6459. doi: https://doi.org/10.1109/CVPR.2018.00675
33. J. Carreira, A.Zisserman, “Quo Vadis, Action Recognition? A new model and the kinetics dataset,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017. doi: https://doi.org/10.48550/ arXiv.1705.07750

34. X. Wang, R. Girshick, A. Gupta, K. He, "Non-local Neural Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018,* pp. 7794–7803. doi: https://doi.org/10.1109/ CVPR.2018.00813
35. C. Feichtenhofer, H. Fan, J. Malik, K. He, "SlowFast Networks for Video Recognition," *2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019*, pp. 6201–6210. doi: https://doi.org/10.1109/ICCV.2019.00630
36. C. Feichtenhofer, "X3D: Expanding Architectures for Efficient Video Recognition," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020*, pp. 200–210. doi: https://doi.org/10.1109/ CVPR42600.2020.00028
37. H. Fan et al., "Multiscale Vision Transformers," *2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021*, pp. 6804–6815. doi: https://doi.org/10.1109/ICCV48922.2021.00675
38. G. Bertasius, H. Wang, L. Torresani, "Is space-time attention all you need for video understanding?" *CoRR*, 2021. doi: https://doi.org/10.48550/arXiv.2102.05095
39. D. Neimark, O. Bar, M. Zohar, D. Asselmann, "Video transformer network," *CoRR*, 2021. doi: https://doi.org/10.48550/arXiv.2102.00719
40. A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, C. Schmid, "ViViT: A Video Vision Transformer," *2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021*, pp. 6816–6826. doi: https://doi.org/10.1109/ICCV48922.2021.00676
41. Z. Liu et al., "Video Swin Transformer," *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022*, pp. 3192–3201. doi: https://doi.org/10.1109/CVPR52688.2022.00320
42. Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, Samuel R. Bowman, "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding," *CoRR*, 2018. doi: https://doi.org/10.48550/ arXiv.1804.07461
43. Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018 doi: https://doi.org/10.48550/arXiv.1810.04805
44. Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, Denny Zhou, *MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices*. 2020. doi: https://doi.org/10.48550/arXiv.2004.02984
45. Sahana Viswanath et al., "The DistilBERT Model: A Promising Approach to Improve Machine Reading Comprehension Models," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 8, pp. 293–309, 2023. doi: https://doi.org/10.17762/ijritcc.v11i8.7957
46. Xiaoqi Jiao et al., *TinyBERT: Distilling BERT for Natural Language Understanding*. 2019. doi: https://doi.org/10.48550/arXiv.1909.10351

Received 27.12.2024

INFORMATION ON THE ARTICLE

Alexander A. Zarichkovyi, ORCID: 0000-0002-4132-6424, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: alexander.zarichkovyi@gmail.com

Inna V. Stetsenko, ORCID: 0000-0002-4601-0058, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: stiv.inna@gmail.com

Oleksandr P. Stelmakh, ORCID: 0000-0003-3147-579X, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: stelmahwork@gmail.com

Anton Yu. Dyfuchyn, ORCID: 0000-0002-1722-8840, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: difuchin@gmail.com

Yaroslav I. Kornaga, ORCID: 0000-0001-9768-2615, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Ukraine, e-mail: slovyan_k@ukr.net

ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ МОДЕЛЕЙ МАШИННОГО НАВЧАННЯ: УНІФІКОВАНА МЕТРИКА БАЛАНСУВАННЯ ПРОДУКТИВНОСТІ ТА ВАРТОСТІ / О.А. Зарічковий, І.В. Стеценко, О.П. Стельмах, А.Ю. Дифучин, Я.І. Корнага

Анотація. Подано нову уніфіковану метрику для оцінювання ефективності моделей машинного навчання, глибокого навчання та штучного інтелекту шляхом балансування продуктивності та вартості виконання. Наявні метрики зазвичай ізольовано враховують лише продуктивність або лише обчислювальні характеристики (наприклад, FLOPs, затримку, енергоспоживання), не відображаючи притаманний компроміс між обмеженими ресурсами та здатністю до передбачення в єдиній формулі. Запропоновано формулу, яка містить налаштовуваний фактор компромісу та жорсткі обмеження на продуктивність і вартість, що дає змогу здійснювати принципове порівняння між моделями та середовищами розгортання. Формалізація узагальнює попередні евристики та демонструє чітку інтерпретованість, масштабованість і врахування особливостей апаратного забезпечення.

Ключові слова: ефективність штучного інтелекту, обчислювально-орієнтоване оцінювання, оцінювання моделей, сталість штучного інтелекту, ефективність програмного забезпечення.

UDC 004.8:519.816
DOI: 10.20535/SRIT.2308-8893.2026.1.11

PHYSICAL-INFORMED NEURAL NETWORK IN SIGNAL PROCESSING AND NETWORK TRAFFIC COMMUNICATIONS

O.V. ZOLOTUKHIN, M.S. KUDRYAVTSEVA, Y.V. BODYANSKIY, V.O. FILATOV, A.V. ANTILIKATOROV, D.V. KALININ

Abstract. Problem of signal processing and network traffic optimization is solved at the hardware level and is interesting, modern and relevant from the point of view of the application level. It is necessary to propose an approach that combines machine learning methods with network bandwidth tasks and traffic over the network. To solve this problem, it is proposed to use concept of Informed Machine Learning (IML), that is the Taxonomy of IML, the principles of constructing deep machine learning systems based on information about the physical properties of the data transmission network under study. The platform for developing is deep machine learning models using PINN neural networks. The PINN represents the class of deep learning algorithms that can integrate data with or without physical processes description. As an algorithm it is proposed to use the popular algorithm in the field of deep learning – ADAM (Adaptive Moment Estimation) for optimizing network traffic. Using the PINN trained with the ADAM algorithm to transmit data the efficiency has increased. Thanks to this method, it is possible to obtain a low noise signal in practice, due to which network traffic is optimized.

Keywords: Adaptive Moment Estimation, artificial intelligence, artificial neural network, Data Visualization, dynamic neuron, gradient decent, Informed Machine Learning, learning algorithm, network bandwidth utilization, network traffic optimization, neuro-fuzzy logic, Python.

INTRODUCTION

The main task of optimizing information networks is to maintain the required level of performance characteristics and network capacity under conditions of load changes.

To assess the quality of any network, characteristics such as data transfer speed, reliability of transmitted information, bandwidth and reliability of the communication channel are used.

The main characteristic of information transmission channels is their bandwidth, that is the amount of data that is going threw a network with a given speed depending on the channel capacity.

Currently, the load on the data transmission channel can vary by an order of magnitude depending on the time of day, and the nature of the transmitted data is determined by the user's field of activity (for example, large videos, project files, high quality images), therefore, optimization problems must be solved in real time.

First of all, it is necessary to formulate criteria for the effectiveness of the information network. Most often, these criteria are performance and reliability, which in turn require the selection of specific evaluation indicators. For example, the performance of an information network is determined by the response time (the time interval between the occurrence of a user request for any network service and

the receipt of a response to this request), and the reliability of the network is determined by the readiness ratio of the network equipment to perform it's functions at an arbitrary point in time.

In addition, it is necessary to determine many variable network parameters that directly or indirectly affect the performance criteria. Settings can be devices, protocols, or transmission technologies.

Thus, to optimize data transmission over a network, it is advisable to consider three successive stages of network management.

1. Bringing the network into working condition, which usually includes:
- searching for faulty network elements at the physical or data link levels;
- checking the compatibility of equipment and software (at the network level);
- selection of correct values for key parameters of programs and devices that ensure data transfer between network nodes — addresses of networks and nodes, protocols used, types of frames, packets (at the transport level).

In this case, optimization comes down to diagnosing faults and bringing the network into an operational state.

2. Primary configuration — selection of parameters that significantly affect the characteristics of the network. If the network is efficient, but communication is very slow due to unacceptable latency or communication sessions are frequently interrupted, then you need to look for the key factors that degrade the network. Typically, the reasons for a noticeable decrease in network performance or unstable network operation are found in an incorrectly operating element or an incorrectly set parameter, but due to the large number of such parameters, solving this problem may require long-term monitoring of network operation, collecting statistics, and searching through options. At this stage, a certain threshold value of the efficiency indicator is also set and it is required to find a network variant for which this value would be no worse than the threshold.

3. The final configuration of network parameters is directly optimizing the network operation. In the case of a normally operating network, further improving it's quality, as a rule, requires finding some optimal combination of values for a large number of parameters.

During the final network setup, in which the parameters of the operating network, for example, the frame size or the size of the window of unacknowledged packets, can be varied in order to improve performance (for example, the average response time) by at least a few percent.

As a rule, network optimization is understood as some kind of compromise approach. It is necessary to select values of network parameters such that it's efficiency indicators are at least not lower than the maximum permissible values specified when choosing the global Quality of Service level.

In real conditions, it is enough to find a solution close to the optimal one, i.e. it is necessary to find some rational version of the structure and parameters of the network.

MATERIALS AND METHODS

The business of geographically distributed companies with an extensive branch network largely depends on the speed and stability of information exchange between departments. The WANs (Wide Area Networks) used for this usually do not always meet the requirements for the productive operation of such critical

applications as VoIP, video conferencing, Enterprise Resource Planning systems and other software.

WAN optimization solutions help companies increase data transfer speeds without expanding the channel, and network administrators can improve application performance, avoid data transfer delays and packet losses, and ensure guaranteed service quality.

At present, when companies are actively centralizing IT infrastructures to improve security and reduce management costs, and end users are increasingly working in mobile office mode, the most pressing task for network administrators is increasing data transfer speeds.

The modern network architecture is presented on Fig. 1. The distributed network architecture is presented in the form of a chain, which consists of: the user's local network, the user's provider, the global Internet, the recipient's provider and the recipient's local network.

Based on this complex hierarchical architecture, the problem of packet data transmission becomes clear — the amount of data transmitted exceeds the amount of data received due to network noise.

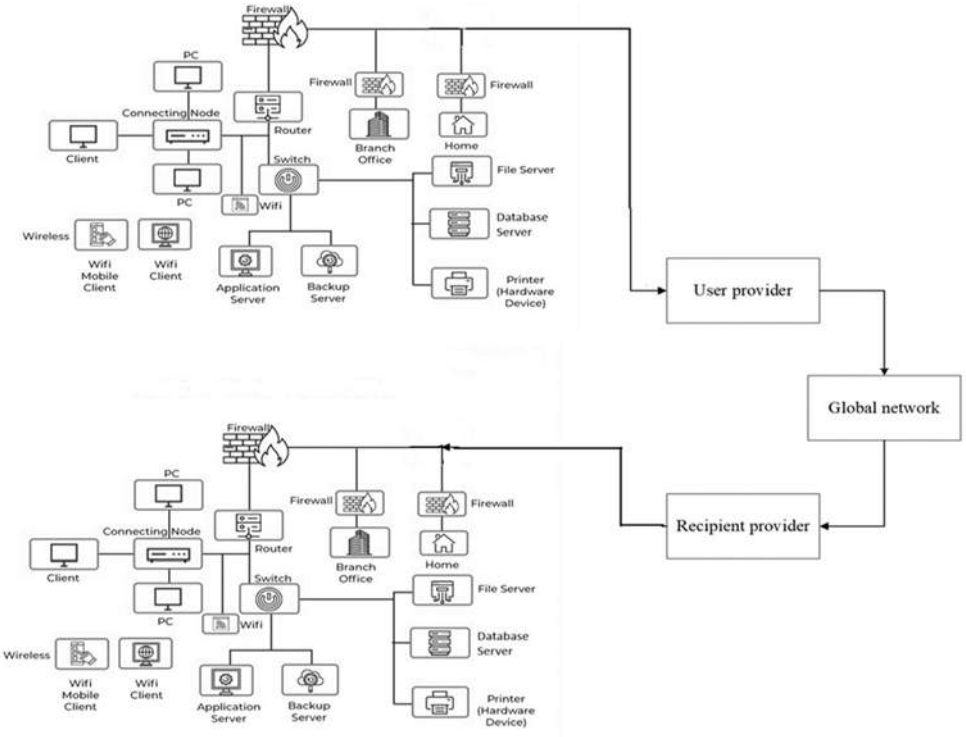


Fig. 1. The distributed network architecture

So, today problem of signal processing and network traffic optimization is solved at the hardware (or protocol level) and is interesting, modern and relevant from the point of view of the application level. Representation of the protocol and user data transfer levels you can see on the Fig. 2.

It is necessary to propose an approach that combines machine learning methods, for example, neural networks with network bandwidth tasks and traffic over the network.

This will allow us to present not only a theoretical, but also a practical solution to the problem.

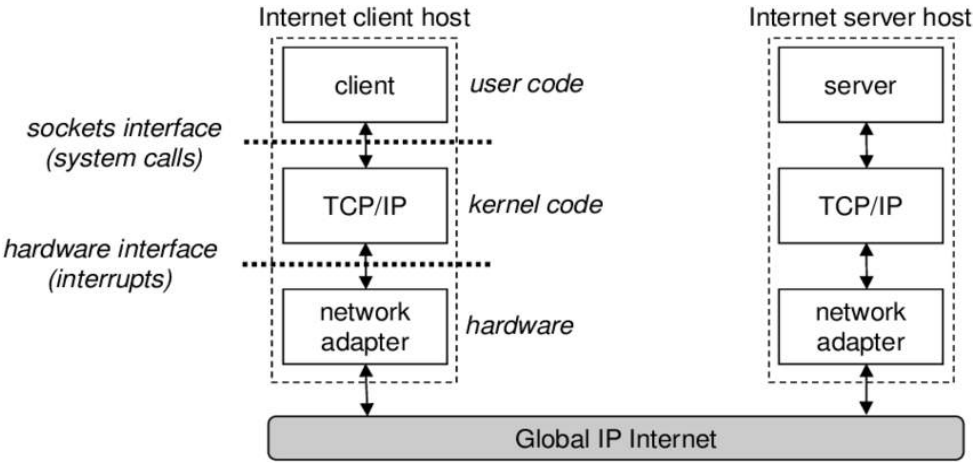


Fig. 2. Representation of the protocol and user data transfer levels

One way to optimize of data traffic at the hardware level is solved by configuring protocol parameters for packet transmission: packet size and number of packets. The packet transmission rate is determined by the speed of light and depends on the number of nodes from point A to point B of network.

For example, we established a connection with a remote computer, sent 16 packets, received confirmation that 12 packets were received, and in some packets the checksum does not match. We transmit the following packets, to them we re-add packets in which the correct checksum was not received. This is already an optimization compared to the fact that it is possible to transmit data in one packet and wait for confirmation each time, and in case of incorrect transmission of the packet, resend it.

To solve this problem in [1, 2] works it is proposed to use concept of Informed Machine Learning (IML). IML describes learning from a hybrid information source that consists of data and prior knowledge. The prior knowledge comes from an independent source, is given by formal representations, and is explicitly integrated into the machine learning pipeline.

Physics-informed machine learning integrates seamlessly data and mathematical physics models, even in partially understood, uncertain and high-dimensional contexts [1].

Taxonomy of Informed Machine Learning is represented on Fig. 3. This taxonomy serves as a classification framework for informed machine learning and structures approaches according to the three above analysis questions about the knowledge source, knowledge representation and knowledge integration. Authors identified for each dimension a set of elements that represent a spectrum of different approaches [2].

This paper proposes to use the Taxonomy of Informed Machine Learning, the principles of constructing deep machine learning systems based on information about the physical properties of the information data transmission network under study. The platform for developing intelligent tools is deep machine learning models using PINN neural networks.

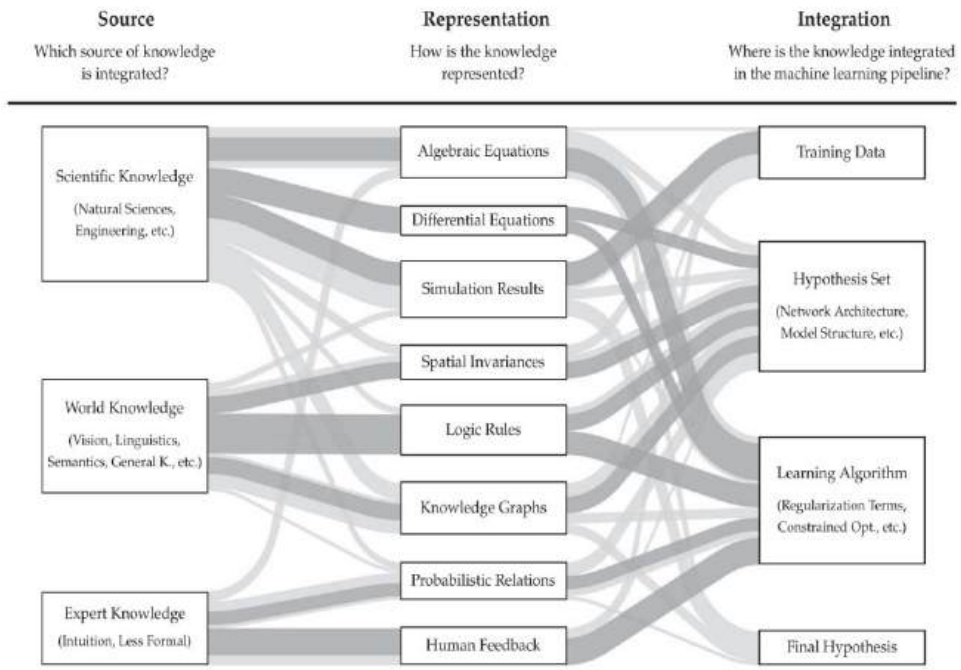


Fig. 3. Taxonomy of Informed Machine Learning

ARCHITECTURE OF PHYSICS-INFORMED NEURAL NETWORK

Thus, it is necessary to explore the gray box model in state space. A gray box represents a model that learns from data, guided by information about the applied physical properties or laws. Such models can be further used for adaptive control and self-organization. Peculiarity PINN is to initially take into account the underlying description of the physical interpretation of partial or ordinary differential equations, that is, the physics of the problem, rather than trying to derive a solution based solely on the data, that is, by approximating a set of state-value pairs with a neural network. Therefore, the use of state space models is considered.

So, the PINNs represent the class of deep learning algorithms that can seamlessly integrate data and abstract mathematical operators, including partial differential equations (PDE) with or without missing physics.

According to [1, 2] to design the PINN network for signal processing and network traffing optimisation it is necessary to implement the chain: Scientific Knowledge-Algebraic or Differential Equations-Learning algorithms (Fig. 4).

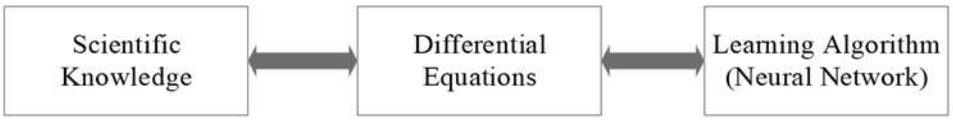


Fig. 4. Stages of the approach based on Physics-Informed Neural Networks

Let’s explore this method from the point of view of using a simple neural network architecture.

A regular Neural Network approximates various functions well (Fig. 5).

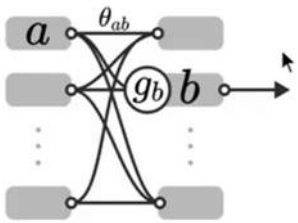


Fig. 5. A regular Neural Network

According to the Tsybenko theorem [3], an artificial neural network of feed-forward with one hidden layer can approximate any continious functionof many variables using the input vector and the weights of each vector with any accuracy, i.e.

$$b = g_b(\sum_a \theta_{ab} \cdot a)$$

Neural networks provide access to the computational graph (Fig. 6), that is access to all elementary operations on numbers that enter or exit the neural network [4].

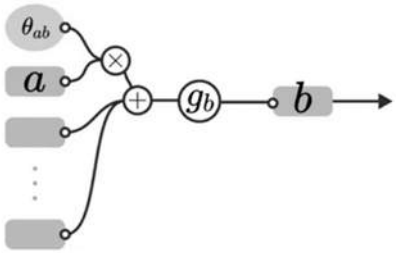


Fig. 6. The computational graph

The graph gives the property of auto differentiation. This allows to find the derivative of a neural network based on its parameters. This is used to update the parameters using the backpropagation algorithm (Fig. 7) through the derivative of the error and update the weights Θ. This is how a neural network is trained.

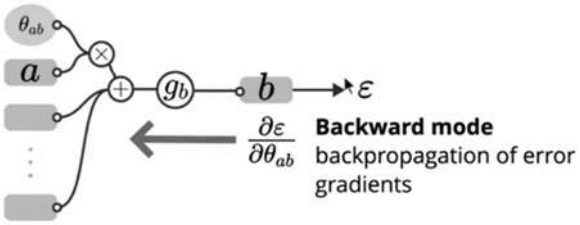


Fig. 7. The computational graph with the backpropagation algorithm

With the help of the computational graph, it is possible to differentiate the neural network with respect to intermediate results of the calculation of any input

parameter a, $\frac{dF}{da}$, or with respect to the input parameters X, for example, with respect to some physical parameter X [5].

You can write an equation in the form of partial derivatives $PDE\ (F, \frac{dF}{dx}, \ldots)$(Fig. 8), which can be used in physics (these can be coordinates or time of a process or other physical parameters).

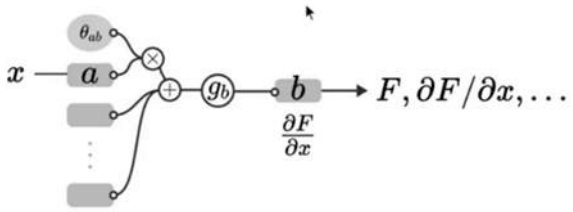


Fig. 8. The computational graph with partial derivatives

Let's consider the learning methods of a neural network. There are three categories of methods to train a neural network:

- observational bias involves supervisor training, submiting a lot of data on the physical parameters of the system;
- inductive bias involves constructing a neural network so that it complies to physical laws, for example using a convolutional neural network;
- learning bias involves to reduce the error through the derivatives of the parameters.

Let's consider the Physics-Informed Neural Network architecture for network traffic optimization.

The PINN for network traffic optimization represents a function of variables: the time of data transfer, the volume of data transfer in packets, the number of nodes (Fig. 9).

So, to solve the problem of digital signal processing and network traffic optimization, you can use dynamic neurons that are described by differential or difference equations. The behavior of such dynamic neurons is significantly determined by their prehistory.

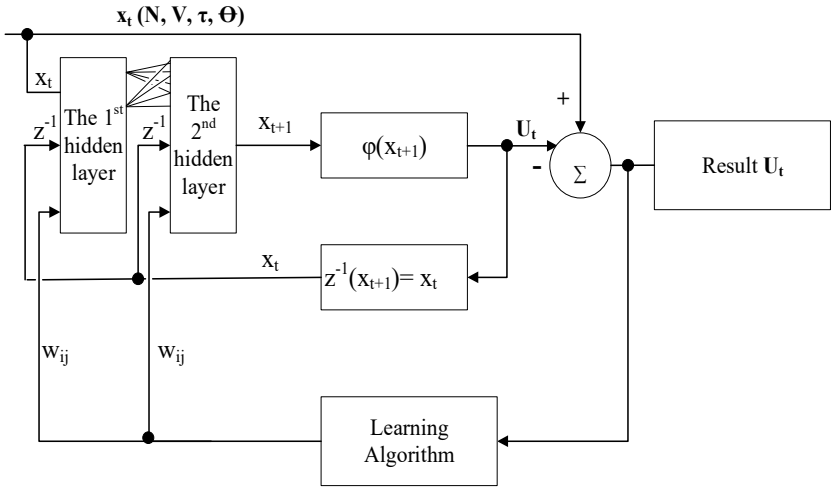


Fig. 9. The PINN neural network for network traffic optimization

The input parameters of the neural network are: the time — τ; the amount of transmitted data in packets – V; the number of nodes — N; the output parameter is the network bandwidth utilization – U. The time includes: time of protocol connection and confirmation of two devices, $\tau\ con$; packet transferring time, $\tau\ pass$; processing time, τ proc: $\tau = 2* \tau_{con} + \tau_{pass} + \tau_{proc}$.

What is network bandwidth utilization? In the simplest terms, network bandwidth utilization is the rate at which data can flow through the network. Traditionally measured in Mbits per second (Mbps), higher bandwidth allows more traffic flow from one device to another. Utilization is the percentage of a network's bandwidth that is currently being consumed by network traffic.

Let's consider neural network. Neural Network has two hidden layers (determined experimentally). The Input vector of neural network is

$$x_t\ (N,\ V,\ \tau,\ \Theta)$$

where $\tau = 0,1..n$; $\Theta = \{w_{ij}\}$, $i,j \in [1..k]$; t – timestep (discrete moments in time); the output vector of neural network is Ut.

The neuron of the neural network is a model of a nonlinear dynamic system in state space, that is, the time factor influences the behavior of the neuron, and the output signal U_t is determined by the input signals x_{t+1} and depends on the past states of the system x_t.

In this neural network, a delay element is included in the feedback circuit, which implements the backward shift operation $z^{-1}(x_{t+1}) = x_t$ and provides the neuron with the necessary dynamic properties over time. This can be seen in the inner circuit of the neural network. Thereby, the neural network allows to calculate the data network utilization parameter dynamically, at times t, $t+1$, and so on.

A dynamic neuron is described by the recurrent equation

$$x_{t+1} = \varphi(\sum w_{ij} * x_t + Q_j)$$

where $t = 0.1..$; $j = 1,2..n$; $i = 1,2..n$.

We will update the parameters of the neural network with learning algorithm until the error does not satisfy the result. If the error is satisfactory, then we will assume that we have found a solution to the system, that is, we have determined the network bandwidth utilization.

As a training algorithm, consider the classic gradient descent, which is an optimization algorithm used to minimize errors in a machine learning model.

Despite the fact that the gradient method of training a neural network and its modified version in the form of backpropagation are the most common algorithms in machine learning, the task of optimizing neural networks remains a rather difficult problem today.

The gradient descent method has a number of problems for training deep neural networks [6–8]:

- the gradient method gets stuck in a fairly deep local minimum (Fig. 10). There are solutions to sometimes work around such problems, such as momentum, which can carry optimizers through large lifts, or batch normalization, which smooths out the error space. But the root cause of many branching problems in neural networks is still a local minimum;

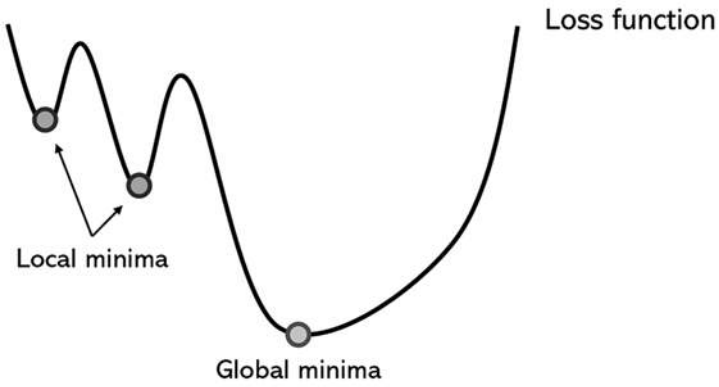


Fig. 10. Using the gradient descent method in the problem of searching for a global extremum

- high time spent on training. Gradient descent, due to its low convergence rate, is typically a time-consuming method, even with adaptation options for large data sets such as batch gradient descent.
- the gradient method is highly sensitive to optimizer initialization. For example, performance may be significantly better if the optimizer is initialized near the second local minimum rather than the first, but this is all arbitrary.
- the pace of learning determines the degree of confidence and riskiness of the optimizer. Setting the learning rate too high may result in missing the global minimum, while setting the learning rate too low will increase execution time. To solve this problem, the learning rate is reduced gradually, but choosing the rate of decrease, taking into account many other variables that determine it, is quite difficult.
- gradient descent requires gradients. This means that it is vulnerable to inherent problems like damping or exploding gradients, in addition to its inability to handle non-differentiable functions.

Based on the analysis of gradient descent problems, it is proposed to use a complex optimizer to train a neural network.

An optimizer is an algorithm for achieving the best, most accurate results while increasing the learning rate. In other words, it is an algorithm used to change parameters such as weights of neural network and learning rates slightly so that the model is adequate and can produce accurate results quickly. To do this, the optimizer algorithm adjusts the neural network connection weights [9,10].

We proposed to use the popular optimization algorithm in the field of deep learning – optimizer ADAM. The name is derived from Adaptive Moment Estimation.

The algorithm has the following advantages compared to other optimizers:

- the algorithm works effectively with online and streaming data, works well with non-stationary data, for example, it allows to optimize noise [11,12];
- the algorithm allows to very accurately approximate a set of points with a linear (Fig. 11) or nonlinear function (Fig. 12);
- the algorithm has a simple implementation and computational efficiency, provides good accuracy compared to other optimizers, and is not demanding on computer resources [13,14];

- an important advantage of ADAM is that updating the w_t parameter is completely invariant to gradient scaling, the algorithm converges even if the objective function changes over the time or the objective function is nonlinear and contains several local extrema, which is relevant for the data transmission over a network (Fig. 13).

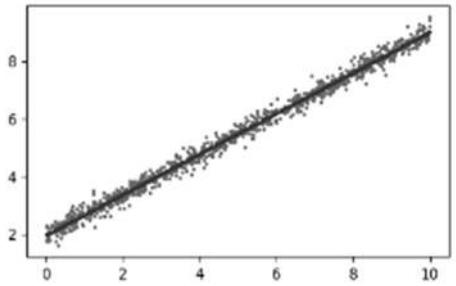


Fig. 11. Linear approximation function

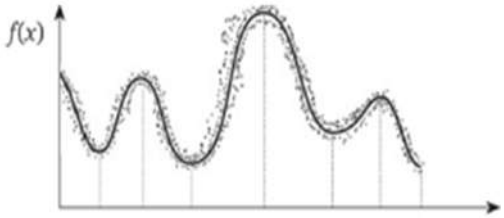


Fig. 12. Nonlinear approximation function

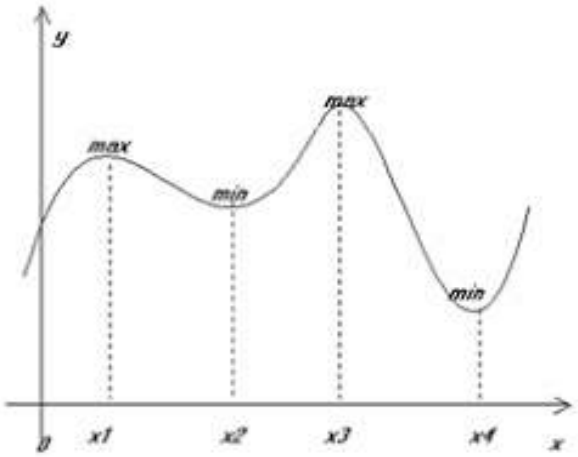


Fig. 13. Function with several local extrema

Let's consider the ADAM algorithm for optimizing network traffic.

The optimizer is called Adaptive Moment Estimation because it uses estimation of the first moment of the gradient, m_1 (the mean), and estimation of the second moment of the gradient, m_2 (the uncentered variance) to adapt the learning rate for each weight of the neural network.

The algorithm calculates the exponential moving average of the gradient p_t and the squared gradient q_t, and the parameters m_1 and m_2 control the decay rate of these moving averages.

The initial value of the moving averages p_t, $_{qt}$ and values of m_1 and m_2 close to 1 (which are recommended for this algorithm, since the algorithm is stable to the initial values of the learning rate and damping coefficient), you can see this on command initialize.

This leads to a shift in the moment estimates $\widehat{p_t}\widehat{q_t}$, towards zero. This displacement is overcome by correcting the 1st and 2nd order moments. Next, the weight parameter w_t is updated.

The algorithmic model can be presented in the following form.

Initialize t=0, m_1=0.9, m_2=0.999, ϵ=10-8, α=0.001

Step 1: while w_t do not converges

do{

t=t+1

Step 2: Calculate gradient $g_t = \frac{\partial f(x,w)}{\partial w}$

Step 3: Calculate $p_t = m_1 \cdot p_{t-1} + (1 - m_1) \cdot g_t$

Step 4: Calculate $q_t = m_2 \cdot q_{t-1} + (1 - m_2) \cdot g_t^2$

Step 5: Calculate $\widehat{p_t} = p_t / (1 - m_1^t)$

Step 6: Calculate $\widehat{q_t} = q_t / (1 - m_2^t)$

Step 7: Update the parameter $w_t = w_{t-1} - \alpha \cdot \widehat{p_t} / \left(\sqrt{\widehat{q_t}} + \epsilon\right)$

}

Step 8: return w_t

where *f(w)* — nonlinear stochastic objective function with parameter *w* containing local extremes; *g(t)* — gradient at time t along w, allows to get the direction to move towards local extrema; *t* — timestep, t=1, 2..; m_1 — the first moment vector (mean), the proposed value for ADAM is m_1=0.9; m_2 — the second moment vector (uncertained variance), the proposed value for ADAM is m_2=0.999; p_t — bias of the first order moment; q_t — bias of the second order moment; $\widehat{p_t}$ — bias correction of the first order moment; $\widehat{q_t}$ — bias correction of the second order moment; α — initial learning rate, the proposed value for ADAM is α=0.001; ϵ — parameter preventing division by zero, does not affect learning, the proposed value for ADAM is $\epsilon = 10^{-8}$.

The implementation of this algorithm allows adaptive adjustment of the optimal learning speed of each model parameter, which leads to highly accurate results, but storing the model parameters consumes memory twice as much as the model itself, which is an obstacle when training large models. In practice, to support such an algorithm with high memory consumption, it is necessary to use unloading to the CPU, which increases the delay and slows down the learning process.

RESULTS OF COMPUTATIONAL EXPERIMENTS

This section presents a description of the experimental learning and obtained results. A practical implementation of PINN learning with the ADAM algorithm using Python you can see on the Figs. 14–16.

Fig. 14 shows an example of a fragment of a network signal.

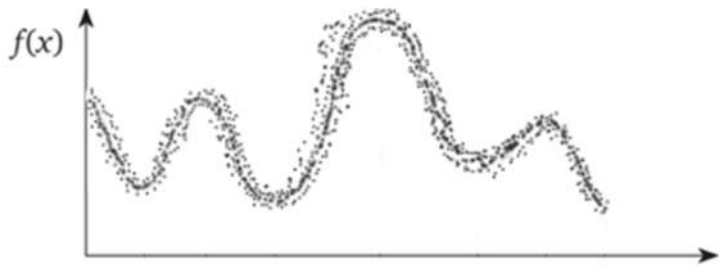


Fig. 14. An example of a fragment of a network signal

The Fig. 15 shows the restored at these points the initial nonlinear function *f*, in our case this is the optimized network signal. From Fig. 15 it follows that we quite accurately approximated the set of points with a nonlinear function. This was the purpose of learning — finding unknown parameters $(\widehat{p_t}, \widehat{q_t})$ to minimize a given loss function.

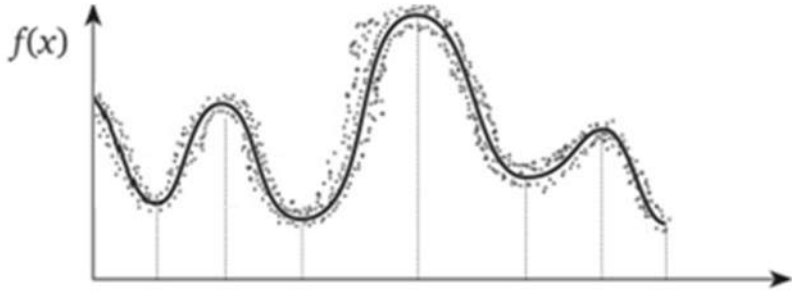


Fig. 15. The restored at points the initial nonlinear function *f*

Fig. 16 shows the network signal function directly without noise.

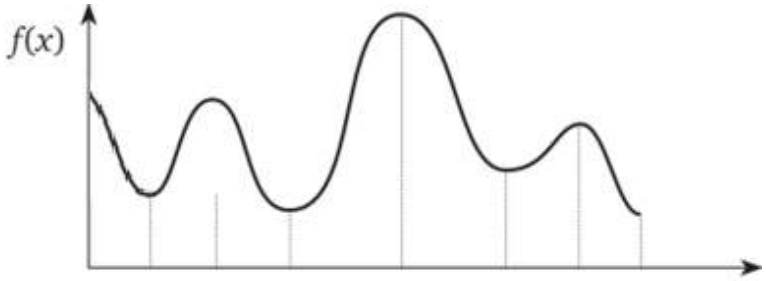


Fig. 16. The network signal function directly without noise

Let's consider the practical example. For example, we tried to send packages to Tfl.gov.uk from our network in Kharkiv, Ukraine. Let's complete the following steps:

- trace using the command tracert in Windows Operation system to determine the number of nodes;
- execute a command ping to send packets without using PINN;
- calculate network bandwidth utilization for data transfer without using PINN;
- execute a command ping to send packets using PINN;
- calculate network bandwidth utilization for data transfer using PINN.

Calculate the result of sending 4 packet of 4096 B to Tfl.gov.uk without using PINN.

$$U1 = \frac{4 \times 4096\ B}{(27 + 193 + 61 + 49)\ ms} \times 100\% = \frac{0{,}016384\ MB}{0{,}327\ s} \times 100\% =$$

$$= 0{,}05 \frac{MB}{s} \times 100\% = 5{,}0\%$$

And testing data transmission using the PINN trained with the ADAM algorithm we obtained the following results:

$$U2 = \frac{4 \times 4096\ B}{(17 + 12 + 15 + 35)\ ms} \times 100\% = \frac{0{,}016384\ MB}{0{,}079\ s} \times 100\% =$$

$$= 0{,}207 \frac{MB}{s} \times 100\% = 20{,}7\%$$

So, using the PINN trained with the ADAM algorithm to trasnsmit data the efficiency has increased. Thanks to this method, it is possible to obtain a low noise signal in practice, due to which network traffic is optimized.

DISCUSSION

Based on the research results presented in this article, the following main conclusions can be drawn.

Network traffic optimization can be improved at the protocol level and at the application level.

At the protocol level you can increase the number of transmitted packets. The more packets are transmitted, the greater efficiency of data transmission will be, but taking into account the channel capacity and packet loss in the network.

At the application level you can use Physics-Informed Neural Network for optimization of network bandwidth utilization. While high network utilization indicates the network is busy, low network utilization indicates the network is idle. Using the PINN trained with the ADAM algorithm to trasnsmit data the efficiency is 15.7 %.

CONCLUSION

At the application level it is proposed to use also another neural network architecture – Learning Vector Quantization (LVQ) with Encoder-Decoder architecture, which is suitable for optimizing of network noises, texts and images over the network.

ACKNOWLEDGMENTS

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.

REFERENCES

1. C. Jiang, H. Zhang, Y. Ren, Z. Han, K. Chen, L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Communications*, vol. 24, no. 2, pp. 98–105, 2017. doi: https://doi.org/ 10.1109/MWC.2016.1500356WC
2. X. Zhang, T. Wang, "Elastic and reliable bandwidth reservation based on distributed traffic monitoring and control," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 12, pp. 4563–4580, Aug. 2022. doi: https://doi.org/10.1109/TPDS.2022.3196840
3. V. Paxson, F. Floyd, "Wide area traffic: The failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, Jun. 1995. doi: https://doi.org/10.1109/90.392383
4. S. Huang, P. Wei, B. Hualaitu, "Bandwidth optimization of information application system under fine integral method of fuzzy fractional order ordinary differential equations," *Alexandria Engineering Journal*, vol. 59, issue 4, pp. 2793–2801, 2020. doi: https://doi.org/10.1016/j.aej.2020.06.015

5. G. Karniadakis, I. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, L. Yang, "Physics-informed machine learning," *Nature Reviews Physics*, vol. 3, no. 6, pp. 422–440, 2021. doi: https://doi.org/10.1038/s42254-021-00314-5
6. L. Von Rueden et al., "Informed machine learning – A taxonomy and survey of integrating prior knowledge into learning systems," *IEEE Transactions on Knowledge and Data Engineering*, 2023, pp. 614–633. doi: https:// doi.org/10.1109/TKDE.2021.3079836
7. E. Bodyanskyi, O. Rudenko, *Artificial neural networks: Architecture, training, application.* Kharkiv: Teletech, 2004, 369 p.
8. O. Rudenko, E. Bodiansky, *Artificial neural networks.* Kharkiv: Smith Company, 2006, 390 p.
9. V. Filatov, O. Zolotukhin, A. Yerokhin, M. Kudryavtseva, "The methods for the prediction of climate control indicators in the Internet of Things systems," *CEUR Workshop Proceedings*, pp. 391–400, 2021. doi: https://doi.org/10.5281/zenodo.14526027
10. I.E. Lagaris, A. Likas, D.I. Fotiadis, "Artificial neural networks for solving ordinary and partial differential equations," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, pp. 987–1000, Sept. 1998. doi: https://doi.org/ 10.1109/72.712178
11. Y. Bodyanskiy, O. Zolotukhin, A. Yerokhin, M. Kudryavtseva, M. Yerokhin, "Fast stacking neuro-neo-fuzzy system for inverse modeling in online mode," *International Journal of Computing*, vol. 24, no. 4, pp. 661–667, 2025. doi: https://doi.org/10.47839/ijc.24.4.4330
12. O. Zolotukhin, V. Filatov, A. Yerokhin, M. Kudryavtseva, and V. Semenets, "An approach to the selection of behavior patterns autonomous intelligent mobile systems," in *IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*, *Kharkiv, Ukraine, 2021*, pp. 349–352. doi: https://doi.org/10.1109/PICST54195.2021.9772110
13. V. Filatov, O. Zolotukhin, M. Kudryavtseva, "Intellectual data analysis in relational information and analytical systems," *Innovative Technologies and Scientific Solutions for Industries*, no. 4, pp. 101–111, 2025. doi: https://doi.org/10.30837/2522-9818.2025.4.101
14. J. Uduagbomen, S. Lakshminarayana, M. Leeson, T. Xu, "Physics-informed neural network modeling of solution pulses in optical communication systems," *2022 IEEE Photonics Society Summer Topicals Meeting Series (SUM), Cabo San Lucas, Mexico, 2022*, pp. 1–2. doi: https://doi.org/10.1109/ SUM53465.2022.9858244

Received 16.01.2025

INFORMATION ON THE ARTICLE

Oleg V. Zolotukhin, ORCID: 0000-0002-0152-7600, Kharkiv National University of Radio Electronics, Ukraine, e-mail: oleg.zolotukhin@nure.ua
Maryna S. Kudryavtseva, ORCID: 0000-0003-0524-5528, Kharkiv National University of Radio Electronics, Ukraine, e-mail: maryna.kudryavtseva@nure.ua
Yevgeniy V. Bodyanskiy, ORCID: 0000-0001-5418-2143, Kharkiv National University of Radio Electronics, Ukraine, e-mail: yevgeniy.bodyanskiy@nure.ua
Valentin O. Filatov, ORCID: 0000-0002-3718-2077, Kharkiv National University of Radio Electronics, Ukraine, e-mail: valentin.filatov@nure.ua
Andriy V. Antilikatorov, Kharkiv National University of Radio Electronics, Ukraine, e-mail: andrii.antilikatorov@nure.ua
Denys V. Kalinin, Kharkiv National University of Radio Electronics, Ukraine, e-mail: denis.kalinin@teamdev.com

НЕЙРОННА МЕРЕЖА З ФІЗИЧНОЮ ІНФОРМАЦІЄЮ ДЛЯ ОБРОБЛЕННЯ СИГНАЛІВ ТА ЗВ'ЯЗКУ МЕРЕЖЕВОГО ТРАФІКУ/ О.В. Золотухін, М.С. Кудрявцева, Є.В. Бодянський, В.О. Філатов, А.В. Антілікаторов, Д.В. Калінін

Анотація. Проблема оброблення сигналів та оптимізації мережевого трафіку вирішується на апаратному рівні і є цікавою, сучасною та актуальною з точки зору прикладного рівня. Необхідно запропонувати підхід, який поєднує методи машинного навчання із завданнями пропускної здатності мережі та трафіком по мережі. Для вирішення цієї задачі запропоновано використовувати концепцію інформованого машинного навчання (IML), а саме таксономію IML, принципи побудови систем глибокого машинного навчання на основі інформації про фізичні властивості досліджуваної мережі передавання даних. Платформою для розроблення є моделі глибокого машинного навчання з використанням нейронних мереж з фізичною інформацією (PINN). Нейронна мережа представляє клас алгоритмів глибокого навчання, які можуть інтегрувати дані з описом фізичних процесів або без них. Як алгоритм запропоновано використовувати популярний алгоритм у галузі глибокого навчання – ADAM для оптимізації мережевого трафіку. Використання PINN, навченого за алгоритмом ADAM, для передавання даних підвищило ефективність. Завдяки такому методу на практиці вдається отримати малошумний сигнал, внаслідок чого оптимізується мережевий трафік.

Ключові слова: оцінка адаптивного моменту, штучний інтелект, штучна нейронна мережа, візуалізація даних, динамічний нейрон, градієнтне зниження, інформоване машинне навчання, алгоритм навчання, використання пропускної здатності мережі, оптимізація мережевого трафіку, нейро-нечітка логіка, Python.

ВІДОМОСТІ ПРО АВТОРІВ

Алєксєйчук Богдан Михайлович,
аспірант кафедри судноводіння і морської безпеки Одеського національного морського університету, Україна, Одеса

Антілікаторов Андрій Володимирович,
аспірант Харківського національного університету радіоелектроніки, Україна, Харків

Бодянський Євгеній Володимирович,
професор, доктор технічних наук, професор кафедри штучного інтелекту Харківського національного університету радіоелектроніки, Україна, Харків

Бойко Юрій Володимирович,
доцент, кандидат фізико-математичних наук, завідувач кафедри комп'ютерної інженерії факультету радіофізики, електроніки та комп'ютерних систем Київського національного університету імені Тараса Шевченка, Україна, Київ

Волянський Сергій Михайлович,
доцент, кандидат технічних наук, доцент кафедри судноводіння і морської безпеки Одеського національного морського університету, Україна, Одеса

Горчаков Олексій Олександрович,
аспірант Відділу математичних проблем механіки та теорії керування Інституту математики НАН України, Київ

Гурський Олександр Олександрович,
кандидат технічних наук, доцент кафедри автоматизації технологічних, електромеханічних і робототехнічних систем Одеського національного технологічного університету, Україна, Одеса

Денисенко Андрій Володимирович,
кандидат технічних наук, доцент кафедри інженерії програмного забезпечення Національного університету «Одеська політехніка», Україна, Одеса

Дифучин Антон Юрійович,
доктор філософії, старший викладач кафедри інформатики та програмної інженерії КПІ ім. Ігоря Сікорського, Україна, Київ

Дубна Сергій Михайлович,
старший викладач кафедри автоматизації технологічних, електромеханічних і робототехнічних систем Одеського національного технологічного університету, Україна, Одеса

Зарічковий Олександр Анатолійович,
доктор філософії, асистент кафедри інформатики та програмної інженерії КПІ ім. Ігоря Сікорського, Україна, Київ

Золотухін Олег Вікторович,
доцент, кандидат технічних наук, декан факультету комп'ютерних наук Харківського національного університету радіоелектроніки, Україна, Харків

Іващенко Олександр Вікторович,
старший викладач кафедри інформаційних систем і технологій Національного транспортного університету, Україна, Київ

Калінін Денис Володимирович,
аспірант Харківського національного університету радіоелектроніки, Україна, Харків

Клімов Сергій Васильович,
доцент, кандидат технічних наук, доцент кафедри гідротехнічного будівництва та гідравліки Національного університету водного господарства та природокористування, Україна, Рівне

Ковальов Юрій Едуардович,
аспірант кафедри комп'ютерної інженерії факультету радіофізики, електроніки та комп'ютерних систем Київського національного університету імені Тараса Шевченка, Україна, Київ

Корнага Ярослав Ігорович,
професор, доктор технічних наук, професор кафедри інформаційних систем та технологій КПІ ім. Ігоря Сікорського, Україна, Київ

Крашений Ігор Едуардович,
кандидат технічних наук, старший викладач факультету прикладних наук Українського католицького університету, Україна, Львів

Кудрявцева Марина Сергіївна,
доцент, кандидат технічних наук, професор кафедри штучного інтелекту Харківського національного університету радіоелектроніки, Україна, Харків

Курдюк Сергій Вікторович,
старший дослідник, доктор філософії, Інститут Військово-Морських Сил Національного університету «Одеська морська академія», Україна, Одеса

Матвієнко Сергій Миколайович,
асистент, кандидат технічних наук, асистент кафедри комп'ютерно-інтегрованих технологій виробництва приладів факультету робототехніки та приладобудування КПІ ім. Ігоря Сікорського, Україна, Київ

Мельник Олексій Миколайович,
професор, доктор технічних наук, завідувач кафедри судноводіння і морської безпеки Одеського національного морського університету, Україна, Одеса

Онищенко Олег Анатолійович,
професор, доктор технічних наук, професор кафедри управління судном Національного університету «Одеська морська академія», Україна, Одеса

Сидорський Володимир Сергійович,
аспірант кафедри математичних методів системного аналізу НН ІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Смирнов Сергій Анатолійович,
старший науковий співробітник, кандидат-фізико-математичних наук, доцент кафедри інформаційної безпеки НН ФТІ КПІ ім. Ігоря Сікорського

Старовойт Тетяна Василівна,
аспірантка кафедри системного проектування НН ІПСА КПІ ім. Ігоря Сікорського, Україна, Київ

Стельмах Олександр Петрович,
доктор філософії, старший викладач кафедри інформатики та програмної інженерії КПІ ім. Ігоря Сікорського, Україна, Київ

Стеценко Інна Вячеславівна,
професор, доктор технічних наук, професор кафедри інформатики та програмної інженерії КПІ ім. Ігоря Сікорського, Україна, Київ

Терещенко Іван Миколайович,
кандидат фізико-математичних наук, доцент кафедри математичного моделювання та аналізу даних НН ФТІ КПІ ім. Ігоря Сікорського

Тимчик Григорій Семенович,
професор, доктор технічних наук, професор кафедри комп'ютерно-інтегрованих технологій виробництва приладів факультету робототехніки та приладобудування КПІ ім. Ігоря Сікорського, Україна, Київ

Федін Сергій Сергійович,
професор, доктор технічних наук, професор кафедри інформаційних систем і технологій Національного транспортного університету, Україна, Київ

Філатов Валентин Олександрович,
професор, доктор технічних наук, професор кафедри штучного інтелекту Харківського національного університету радіоелектроніки, Україна, Харків

Швець Олександр Юрійович,
професор, доктор фізико-математичних наук, професор кафедри математичної фізики та диференціальних рівнянь фізико-математичного факультету КПІ ім. Ігоря Сікорського, Україна, Київ

Шевченко Валерій Анатолійович,
професор, доктор технічних наук, завідувач кафедри технічної експлуатації суден Національного університету «Одеська морська академія», Україна, Одеса

Якубенко Олексій Петрович,
технічний директор компанії "Pleso", Україна, Київ