УДК 519.2 DOI: 10.20998/2078-9130.2026.1.357510

ВЛАДИСЛАВ ГАРКУША

ПОРІВНЯННЯ ПОХИБОК МАТЕМАТИЧНИХ МОДЕЛЕЙ І НЕЙРОННИХ МЕРЕЖ ДЛЯ ЗАДАЧ ВИЗНАЧЕННЯ ВУЗЛОВИХ ПЕРЕМІЩЕНЬ ТА ЕКВІВАЛЕНТНИХ НАПРУЖЕНЬ

Предметом дослідження є комп'ютерне моделювання вузлових переміщень та еквівалентних напружень балкового конструкційного елемента. Метою роботи є порівняння точності передбачень, отриманих за допомогою математичних моделей і нейронних мереж. Досліджуваний елемент характеризується геометричними параметрами, умовами закріплення та розподіленим статичним навантаженням, що визначають характер деформацій і напруженого стану. Вихідними даними слугують результати числових експериментів, отримані в системах автоматизованого проєктування на основі варіювання геометричних параметрів і значень прикладеного тиску. Це дає змогу сформувати навчальну вибірку для подальшого аналізу. Досягнення поставленої мети забезпечує можливість оцінити похибки моделей і встановити, який підхід забезпечує вищу точність передбачення вузлових переміщень і еквівалентних напружень за критерієм фон Мізеса. Для реалізації дослідження виконано підготовку даних числових експериментів, визначення ключових параметрів, формування метапараметрів, а також проєктування математичних моделей і архітектури нейронної мережі. Навчання здійснюється за принципом навчання з учителем: вхідними параметрами є геометричні характеристики та навантаження, а вихідними — максимальні значення переміщень і напружень. Передбачення в математичних моделях базується на алгоритмі лінійної регресії, оптимізація параметрів виконується із застосуванням алгоритму Adaptive Moment Estimation. Оцінювання точності здійснюється за допомогою функції середньоквадратичної похибки. Результатом є реалізація та навчання обох типів моделей і їх порівняння за точністю передбачень відносно еталонних даних числового моделювання. Отримані результати формують основу для подальших досліджень, зокрема визначення власних частот і аналізу динамічних характеристик конструкційних елементів.

Ключові слова: нейронні мережі, навчання з вчителем, лінійна регресія, статичний конструкційний міцнісний аналіз, метод скінченних елементів, математична модель.

VLADYSLAV HARKUSHA

A COMPARISON OF ERRORS IN MATHEMATICAL MODELS AND NEURAL NETWORKS FOR DETERMINING NODE DISPLACEMENTS AND EQUIVALENT STRESSES

The subject of this study is the computer simulation of nodal displacements and equivalent stresses in a beam structural element. The objective of this work is to compare the accuracy of predictions obtained using mathematical models and neural networks. The element under study is characterized by geometric parameters, fixing conditions, and a distributed static load, which determines the nature of deformations and the stress state. The input data consists of the results of numerical experiments obtained in computer-aided design systems based on variations in geometric parameters and applied pressure values. This allows for the formation of a training sample for further analysis. Achieving the set goal makes it possible to evaluate model errors and determine which approach provides higher accuracy in predicting nodal displacements and equivalent stresses according to the von Mises criterion. To carry out the study, data from numerical experiments were prepared, key parameters were determined, meta-parameters were formulated, and mathematical models and the neural network architecture were designed. Training is performed using supervised learning: the input parameters are geometric characteristics and loads, while the output parameters are the maximum values of displacements and stresses. Predictions in mathematical models are based on a linear regression algorithm, optimization of parameters is performed using the Adaptive Moment Estimation algorithm. Accuracy is evaluated using the mean squared error function. The result is the implementation and training of both types of models and their comparison in terms of prediction accuracy relative to reference data from numerical simulations. The obtained results form the basis for further research, in particular the determination of natural frequencies and the analysis of the dynamic characteristics of structural elements.

Keywords: neural networks, supervised learning, linear regression, static structural strength analysis, finite element method, mathematical model.

Introduction. Modern engineering design of structural elements is impossible without the use of established approaches to determining the stability and strength of structures based on the scientific principles of applied mechanics. The synergy of using classical mechanical approaches to determine the structural force parameters and the latest innovations in information technologies makes it possible to perform calculations for complex structural elements and determine the key strength characteristics of assembled mechanism assemblies both as separate independent modules as well as parts of complex machines. It should be noted that the complexity and speed of such calculations currently remain at a sufficiently acceptable level for their widespread use by design bureaus and companies manufacturing various machines and assemblies. The approach makes it possible to determine the limit values of loads and safety factors for each specific structure in accordance with its operational requirements within a relatively short period of time. Linear static structural strength analysis is based on the principles of elasticity theory, in which a body in equilibrium—taking into account its constraints—is subjected to time-independent concentrated or distributed loads. Under the action of these loads, the body (the object of study) undergoes deformation. According to the theory of elasticity, the sought-after quantities describing the deformed state of the body are the displacements of its points, strains, and stresses. The calculated values of the latter, reduced to an equivalent value according to a specific strength criterion (in particular, the von Mises criterion), when compared with the limit values for this material obtained experimentally, allow conclusions to be drawn regarding the strength of the structure and its ability to withstand loads.

Elasticity theory approaches are based on analytical relationships and consist of differential equations of equilibrium with respect to stress components, relationships between displacements and Cauchy deformations, and Hooke's law, which relates deformations and stresses. Boundary conditions are added to these equations, where the main challenge is the analytical description of the body's boundary, as well as initial conditions that describe the values of loads or other influences on the body at the start of the calculation or experiment. Therefore, numerical methods are used to simulate processes or states in complex structures or elements.

The most common numerical approach to solving problems of this kind is the finite element method (FEM). FEM is a numerical method used to approximate the solution of differential equations in complex geometric and engineering problems. The main principle of the method lies in dividing the body into simple geometric elements (finite elements), which are connected to each other only at a limited number of nodal points, and applying simple functions to approximate the sought quantities (displacements, deformations, stresses) within them. Thus, initially, based on the principle of minimizing the potential energy of the deformed body, a system of equations is formulated regarding nodal displacements, taking into account loads and constraints. Next, taking these values into account, displacement approximation polynomials are formed for each element, after which the distributions of deformations and stresses within each finite element of the mesh are determined using Cauchy's relations and Hooke's law. Thus, by not solving for the entire geometry at once, but rather performing the solution step by step for each element separately, the desired displacements, deformations, and stresses are obtained for all points of the object under study.[1]

This paper compares the errors in predicting maximum stresses according to the von Mises criterion and the values of nodal displacements under load in a structure with predefined fixations and a static load, obtained using a linear regression model (hereinafter RM) and neural networks (hereinafter NN). By linear regression model should interpreted a linear regression based mathematical model, optimized with ADAM algorithm, which will be proved and described below.

Analysis of recent studies and publications. Novel approaches to calculations based on machine learning models make it possible to adapt mechanical engineering problems for processing using polynomial, linear, and other types of regression algorithms, depending on the domain and nature of the problem facing the engineer. However, it should be noted that certain approaches and software tools have greater error or provide only approximate results in mechanics problems compared to others. Such tools include neural networks (NNs), which receive vectors of several key variables as input data, allowing for the identification of implicit dependencies between parameters using activation functions, such as ReLU or the linear activation function. In contrast, the use of mathematical models to make predictions based on input parameters presented as vectors, which include more obvious (linear) dependencies between key parameters, can provide predictions with lower error.

In [2], the authors present a new approach within the framework of probabilistic interpretation modeling, namely Profile-Wise Analysis (hereinafter PWA). This approach is proposed for use in mathematical models with full or partial uncertainty. The method involves the use of profile functions for forecasting without the need to fix key parameters.

This approach allows for predictions that are close to the results of a full probabilistic interpretation, but with lower time and computational costs. To demonstrate the effectiveness of this approach, the application of PWA in biological models with structural parameter uncertainty was investigated.

In [3], an approach was developed for measuring error in mechanistic mathematical models within a single probabilistic framework. The method allows for the estimation of system parameters and error models via a likelihood function. As alternatives to the Gaussian noise model, multiplicative, Poisson, and log-normal models are proposed.

As in the previously discussed study, this work applies the PWA method to assess the information content of parameters and their impact on predictions. The predictions themselves are generated using the estimated parameters and a mechanistic mathematical model.

In [4], the authors developed a modified hyperelastic model to describe the behavior of HMPE and PET fibers under cyclic loading. The model is based on the use of a modified Yeoh function to predict the nonlinear behavior of materials using linear, quadratic, cubic, and exponential terms that depend on the strain invariants I_1 and I_2.

As a result, it was found that taking both invariants into account improves the accuracy of the hysteresis approximation. Errors of approximately 0.52% and 1.77% were achieved for HMPE and PET materials, respectively, ensuring high accuracy in modeling fibers under conditions of long-term cyclic operation.

In [5], the authors developed an extended friction model that accounts not only for the normal load but also for the sliding speed, contact area, and microstructure of the surfaces. The model is based on decomposing the normal force into external and internal components and describes the coefficient of friction as a function of load and speed.

Additionally, the model accounts for the stages of contact, namely: adhesion, sliding, and wear. This ensures a more accurate reproduction of the actual behavior of surfaces compared to Coulomb's classical law. As a result, the accuracy of predicting the friction force under conditions of varying loads and speeds is improved. Among the model's limitations, it is worth noting the lack of consideration of thermodynamic effects and the assumption of a constant contact area.

In publication [6], the authors examine the application of models in robotics. The review covers kinematic and dynamic models of manipulators, mobile robots, flexible structures, and complex systems. The shortcomings of using classical models, caused by simplifications and linearization that reduce accuracy under real operating conditions, are highlighted separately.

The paper also presents an approach to combining mathematical modeling with artificial intelligence (hereinafter referred to as AI) methods, in particular hybrid approaches that enhance the accuracy and adaptability of mechanisms and structures.

Thus, based on the works discussed above, it can be concluded that this study is relevant, scientifically valuable, and appropriate in terms of comparing mathematical models with machine learning methods within the framework of determining the static load of beam structural elements.

Study purpose and objectives. The objective of this paper is to conduct a comparative analysis of the prediction errors generated by RM and NN in order to determine which of the two approaches yields a smaller error magnitude within a series of validation experiments. The results of computer simulations of mechanical experiments in the ANSYS Mechanical Enterprise Academic Student automated engineering analysis system (hereinafter referred to as ANSYS) were selected as training and validation data. The initial data for the experiments are presented in the form of geometric parameters of the beam structural element under study, values of the distributed pressure acting on the structure, and the resulting values from the computer simulations — the maximum value of equivalent stresses according to the von Mises criterion and the maximum value of nodal displacements. The data for the experiments were generated using a series of engineering calculations, the input parameters for which were prepared using a pseudo-random algorithm and proportional correspondence of geometric configurations for beam structures.

The solution to the problem is focused on the design and preparation of regression models and neural network models, and on comparing the accuracy of their predictions with the initial data obtained through computer experiments solved using the FEM approach. The use of alternative approaches, as opposed to the FEM method, in determining structural characteristics is intended to accelerate the process of solving and determining the structural force parameters, namely, predicting the maximum values of equivalent stresses and the maximum values of nodal displacements. Additionaly prepared analitical solution regarding the general mathematical queations for beam calculation. Such approach will provide source of trouth regarding prepared dataset and trained models.

To achieve this goal, the following problems were solved:

- aggregation of the dataset;
- preparation of key parameters and meta-parameters based on them in the datasets;
- preparation and training of regression models;
- preparation and training of neural networks;
- preparation and formulation of analytical approach;
- analysis and comparison of the results obtained.

Mathematical formulation of problem. Mathematically, we have the following configuration for performing engineering calculations of the stress-strain state of a beam structure. We will create a beam structure with predefined geometric dimensions—width, length, and height—in accordance with the proportional values of the beam structure. The structural element is rigidly fixed at the side edge. The following length-to-height ratio has been selected as the proportion of the beam structural element under investigation:

$$\lambda = \frac{L}{h}, \qquad (1)$$

where: λ – the basic geometric proportion of a rectangular cross-section in a beam structure;

L – beam length;

h – beam height.

Parameter λ for a steel beam structure, the value must be within the range 20–30.

The beam structure is rigidly secured along its edge 0Y and is subjected to a static pressure on the face perpendicular to the fixed face.

The form of the matrix equation to be solved for static strength analysis using the FEM is given by:

$$[\boldsymbol{K}]\{\boldsymbol{u}\} = \{\boldsymbol{F}\}, \qquad (2)$$

where: $[\boldsymbol{K}]$ – system stiffness matrix;

$\{\boldsymbol{u}\}$ – column vector of nodal displacements;

$\{\boldsymbol{F}\}$ – a column vector of known components of the nodal forces.

In this case, Hooke's law, mentioned above, is expressed mathematically as follows:

$$\varepsilon = \boldsymbol{\Phi\sigma}, \qquad (3)$$

where: ε – column vector of deformations;

$\boldsymbol{\Phi}$ – a 6×6 matrix of contracted four-rank tensors;

$\boldsymbol{\sigma}$ – column stress vector.

To prepare the initial parameters for the experiments, a series of computer simulations was conducted to aggregate the required amount of data. Three iterations of calculating the stress-strain state of the structure were performed in ANSYS using the FEM. The variability of the simulation is due to the following constraints:

- variable geometric characteristics of the cross-section of the beam structural element;
- variable geometric characteristics of the cross-section and length of the beam structural element;
- variable geometric characteristics of the cross-section and length, as well as the pressure value to which the beam structural element is subjected.

The solution to the numerical experiment involves a calculation using FEM, which is based on the approach of dividing the body into a structural mesh of elements and finding the solution in each individual element using a simple approximation function. Figure 1 illustrates an example of a mesh applied to a structural element in the ANSYS CAE software.

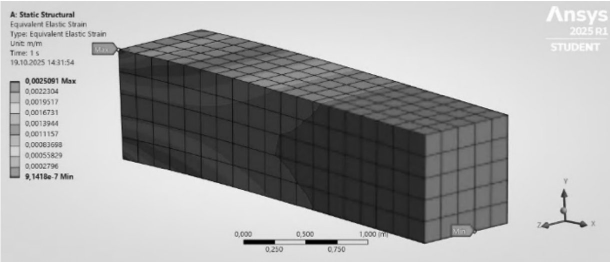

Fig. 1 – An example of calculating the stress-strain state and a specified FEM mesh as part of a computer simulation of a beam structural element.

The solution to this problem involves the development and calibration of finite element and finite volume methods capable of providing predictions, based on the input geometric and physical parameters of a beam structural element, in the form of values for maximum equivalent stresses and nodal displacements, and to compare the magnitude of the prediction error of both approaches for each specific variant of computer simulation data. The novelty lies in comparing and determining a more accurate approach for predicting the stress-strain states of a beam structural element. The practical value lies in identifying an approach capable of determining the maximum value of equivalent stresses in a beam structural element with correspondingly smaller error values.

Numerical experiments data. Conducting a comparative analysis between RM and NN approaches requires prepared datasets that will be used as input and output parameters for the models—that is, in their training and validation. To prepare the data, it is necessary to perform computer simulations, determine the parametric characteristics of the structure under study, and subsequently aggregate the parameters of the beam element. To implement this, chosed to use a prototype structural analysis approach within an automated design system, with typical parameters and subsequent translation into a high-level programming language script, specifically Python, and the corresponding modules of the ANSYS software suite—the gRPC server and the PyMAPDL translator. These modules are used to ensure iterative behavior and the stability of aggregating large amounts of training data. As a prototype of a numerical experiment, a typical static analysis of the stress-strain state of a beam element was performed, followed by the solution of a strength problem: determining the maximum nodal displacements and maximum values of equivalent stresses according to the von Mises criterion.

In accordance with the aforementioned data preparation requirements, the engineering design environment in ANSYS was configured with the Structural Analysis setup to determine the strength characteristics of the beam structure. A separate step involves initializing the software environment, starting the recording of actions within the design system, and translating them into Python directives, which will subsequently be used to iteratively perform model analysis with different parameters and collect the results of the calculations. A similar operation is also performed during the specification of engineering data, the creation of a geometric prototype, and the analysis of the structural element itself.

Regarding the engineering values, the material properties corresponding to the parameters of structural steel were specified, namely Poisson’s ratio, density, and Young’s modulus, after first enabling the recording of actions in the environment. The properties of the surrounding environment are not critical in this study, so they were set to standard values. These include atmospheric pressure of one atmosphere and a room temperature of 22 degrees of Celsius.

The next step was to create a geometric model of the beam structure in accordance with the above-mentioned research conditions, which included the explicit specification of the cross-section and length as mandatory conditions for the geometric shape of the body.

For the analysis itself, an FEM with a mesh consisting of hexahedral quadrilateral elements (hex20) was used, which have 3 degrees of freedom in the longitudinal displacement directions of the coordinate axes—UX, UY, UZ. This element also supports elastic-plastic and visco-elastic nonlinearities and is oriented toward a brick-shaped mesh of elements. A uniform division of the body into such elements was performed, and a mesh network was created.

In accordance with the physical conditions, the boundary conditions for the beam element are specified: the plane of the side face is fixed with zero structural mobility along the axes, and the opposite face is free. The initial conditions were set to zero, i.e., the structure has no prior loading conditions. A uniformly distributed pressure across the entire plane of the upper face of the beam structure was specified as the load.

The next step involves specifying the criteria for analyzing the structure, namely determining the total deformation and equivalent average stresses (according to von Mises) of the element. The results of the analysis are presented in figure 2.

The accuracy of the calculations and, accordingly, the quality of the mesh division of the body into a FEM grid must be verified using the relative difference between the dominant stress components on the element and its limit, or between the equivalent stress and its limit in the local region of high values, the maximum deviation of which should not exceed 7 %:

$$\frac{(\boldsymbol{SMXB}_{\mathrm{SEQV}} - \boldsymbol{SMX}_{\mathrm{SEQV}})}{\boldsymbol{SMX}_{\mathrm{SEQV}}} * 100\%, \quad (4)$$

where: $\boldsymbol{SMXB_{SEQV}}$ – the upper limit of the equivalent stress according to the von Mises criterion, calculated taking into account the stress values from adjacent elements at a specific mesh node;

$\boldsymbol{SMX_{SEQV}}$ – the maximum value of the equivalent stress according to the von Mises criterion in the localregion of high stress values [7].

Based on the formula described above, calculations were performed and a result of 5.77% was obtained, which satisfies the condition for the accuracy of the mesh division of the body into elements, i.e., its sufficient discretization in the regions of highest stress in the beam structure.

Regarding the suggested approach of aggregating data from numeric experiments based of FEM via aggregation script decided to control results based on convergency value in region. Prepared aggregation script will automatically ignore result of experiment convergency values more than 7%, according to Equation 4.

The results of a common calculation for a beam element are placed in Table 1.

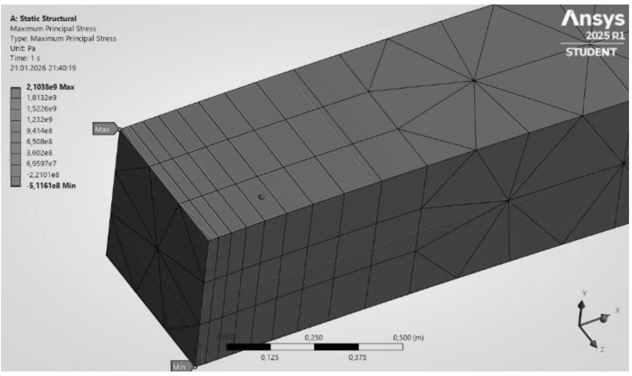

Fig. 2. – Example of a solution to a static structural analysis of a beam structure showing the distribution of equivalent stresses

Table 1 – Values of typical parametric characteristics and results of the numerical experiment

Parameter	Value
Width	1,2 m
Height	0,06 m
Length	0,06 m
Load	9E+6 Pa
Elasticity module	2E+11 N/m2
Poisson's ratio	0,27
Density	7850 kg/m3
The average maximum value of equivalent stresses according to the von Mises criterion	2,1355E+10 Pa
Maximum nodal displacements along the OY axis	4,7819E-5 m

Conducting multi-parameter computational experiments with varying parameters makes it possible to identify key aspects for developing specialized software code designed for repeated iteration of the experiment, with the aim of generating a dataset.

Based on the results of a typical calculation, basic parametric characteristics were identified that are essential for model training and, accordingly, vary from one iteration of the experiment to another.

The following design parameters were identified:

1. Width;
2. Height;
3. Length;
4. Load;
5. Maximum value of equivalent stresses values in the structure by the von Mises criterion;
6. Maximum value of nodal displacements;

Data aggregation and preparation. After performing a standard calculation in the computer-aided design system, all actions carried out during the main stages of the analysis were saved, namely: setting the engineering parameters of the structural material, creating a geometric prototype, and the analysis of the structural element itself, including the definition of the FEM mesh conditions, the initial and boundary conditions of the experiment, and so on. Accordingly, the actions were recorded in the design system and translated into Python directives using PyAPDL.

However, the directives generated in ANSYS are not sufficient to perform experimental iterations in a standard Python execution environment; so, it is brought necessity to install additional libraries that provide access to the APDL API for proper interaction with the aforementioned directives and the design system as a whole. The ANSYS Python library [8] was installed.

The ANSYS library provides an API for interacting with ANSYS Mechanical to perform direct analysis and with ANSYS Geometry Service to create geometric design prototypes. All interaction will be based on the gRPC protocol, which is included in the ANSYS suite and is responsible for performing so-called headless computations.

During the preparation of the data collection program, the number of experiment iterations was calculated to aggregate the number of records using the following formula:

$$N \geq C \frac{P}{\varepsilon}, \quad (5)$$

where: N – the number of data points (vectors) required to train the model;

P – the number of trainable parameters;

ε – relative value of the total error;

C – control constant, inductive bias.

Accordingly, we must calculate the parameter using the given formula:

$$P = \sum_{l=1}^{L} (n_{l-1} + 1)\, n_l, \quad (6)$$

where: L – model layers number;

n – number of neurons per model layer;

Based on the formulas listed above, for training the model with an input vector containing 15 parameters (including meta-parameters), the optimal amount of data is 1000 records (vectors), respectively.

Based on the aforementioned mathematical conditions of the study, three cycles of data aggregation were performed according to three design analysis scenarios.

Important part of study is tracking amount of time that system spent for preparing and aggregating data for alterantive metods models. According to emperical observations was defined that single iteration of simulation takes about 7-10 seconds, which extrapolated to numer of records shows time frame from 116 to 166 minutes per single dataset preparation, which necessary unique for each of 3 experiment conditions. In total was spent from 348 to 498 minutes to aggreate necessary amount of data.

After collecting a sufficient amount of data and forming arrays in accordance with the specified conditions, it is necessary to create meta-parameters, which are derived values from the values aggregated directly during the experiments. To facilitate faster formation of meta-parameters, without complicating data types and program code, and to enable convenient work with large amounts of data, an analysis of ready-made solutions for working with aggregated data was conducted.

The Pandas [9] and Keras [10] libraries were selected for comparison. The comparison focused on the libraries' ability to work with tabular data, performance, and compatibility with ML and ML systems. Pandas offers a wide range of capabilities for working with tabular data, specifically filtering and sorting functions, as well as performing certain types of mathematical transformations on dataset vectors. It is a specialized tool for more advanced use in research and feature engineering. One of its drawbacks is its use of an in-memory data storage approach, meaning it stores data in RAM, which can impose certain limitations when working with large datasets. Keras, on the other hand, has fewer functions and capabilities for working with data compared to the first library, and offers poorer functionality for reproducing results, which is important when conducting various experiments and can complicate the program code. However, it is worth noting that Keras offers native integration capabilities when working with data and directly training models. But this feature is not key, since Keras is specifically designed for working with neural networks [11]. Based on the analysis, comparison, and characteristics of the research subjects, the Pandas library was selected for data processing and the formulation of meta-parameters.

Table 2 presents an example of aggregated data for the first dataset (variable beam cross-section parameters).

Table 2 – Aggregated data examples for problem No. 1

Block_cutting, m2	Max_VMIS, Pa	Max_Delta, m
7,6176	5,42E+08	0,0146765
6,5536	1,48E+09	0,0692422
7,4529	8,98E+07	0,000751757
4,8841	2,91E+08	0,0046644
1,8225	3,55E+08	0,00576457
3,1684	1,15E+08	0,00101172
2,6244	9,24E+08	0,0313842
1,9321	6,97E+08	0,0180675
4,7961	1,21E+08	0,00112887

Explanation for table 2:

- Block_cutting – the cross-sectional area of the beam;
- Max_VMIS – values of maximum equivalent stresses using the von Mises criterion;
- Max_Delta – the value of maximum node displacement along the OY axis;

In accordance with the mathematical problem at hand, the data presented in Table 2 pertain to Problem No. 1, i.e., the cross-sectional parameters are variables, while the length and pressure remain constant.

The next step in data preparation involves the process of so-called feature engineering [12]. The idea behind this procedure is to identify key parameters in the dataset that will be crucial for training the models. A separate aspect of this process is the creation of meta-parameters—derived data—to improve the identification of key parameters in the dataset. The key data in this case are those that will have a clear dependence on the initial parameters of equivalent stresses and nodal displacements.

After performing the data analysis, we can summarize the following list of parameters available at the start of the feature engineering process:

- Length – beam length;
- Width – beam width;
- Height – beam height;
- Pressure – the value of the distributed pressure on the beam
- Horizontal_section – the plane of the top face OX.

Based on the listed parameters, meta-parameters have been defined to enable a more comprehensive analysis of the relationships between parameters. List of meta-parameters:

- Aspect_ratio – the ratio of the length to the height of the structural element;
- Inertia – the moment of inertia;
- Total_force – the total applied force to the area of the top face of the structure;
- Vertical_section – the cross-sectional area of the beam element;
- Moment_max – the maximum value of the bending moment
- Theta – structural stability under bending;
- Sigma – stress intensity under bending.
- The initial parameters are listed below:
- Max_VMIS – maximum value of equivalent stresses according to the von Mises criterion;
- Max_Delta – maximum value of nodal displacements along the OY axis.

After finalizing the dataset, an analysis was performed of the relationships between parameter values and key (input) parameters. It should be noted that, given the presence of several key parameters in the problem conditions, it is necessary to identify dependencies for all of the listed parameters.

Key dependencies were found for the dataset with variable cross-sections:

- between the maximum equivalent stress and the cross-sectional areas of the side faces of the beam structure in a dataset with variable cross-sections. A graphical representation of this relationship is shown in figure 3.

- between the cross-sectional areas and the maximum nodal displacement. A graphical representation of this relationship is shown in figure 4.

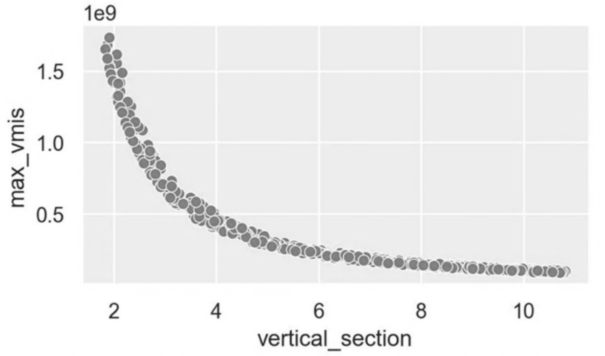

Fig. 3 – Example of the relationship between the area of the top face (x-axis) and the maximum equivalent stresses according to the von Mises criterion (y-axis) for a variable cross-section

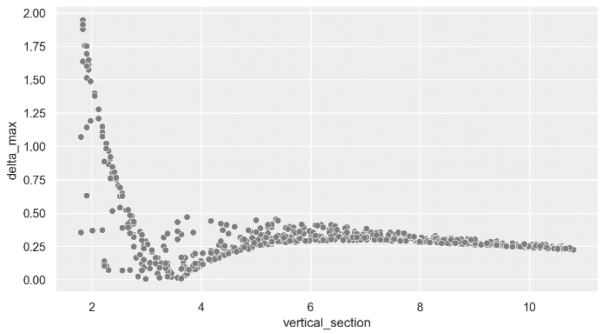

Fig. 4 – Example of the relationship between the area of the lateral face (x-axis) and the maximum nodal displacements (y-axis) for a variable cross-section

In addition, dependencies were identified for datasets with varying cross-sectional areas and lengths:

- the relationship between the length-to-height ratio and the maximum equivalent stress. The relationship is shown in figure 5;
- the relationship between the values of bending stress intensity and maximum nodal displacement.

The relationship is shown in figure 6.

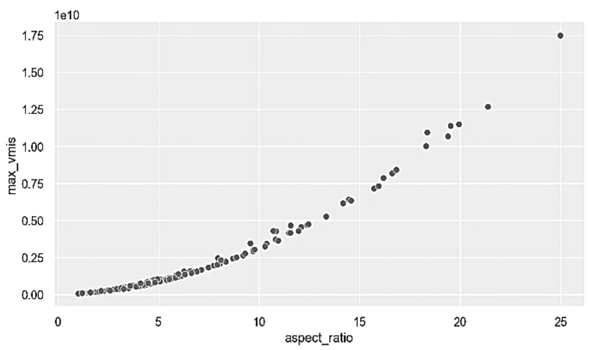

Fig. 5 – Example of the relationship between length and height (x-axis) and maximum equivalent stress (y-axis)

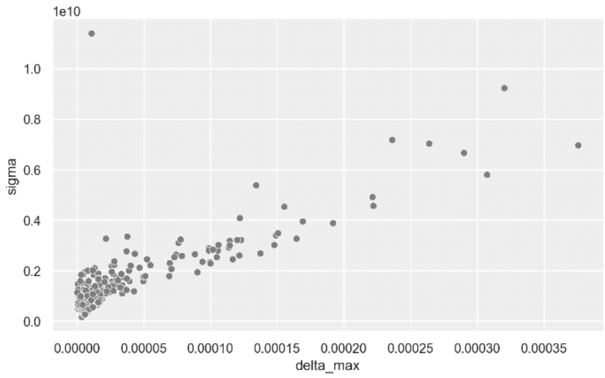

Fig. 6 – Example of the relationship between bending stress intensity (y-axis) and maximum nodal displacement (x-axis)

Based on a dataset obtained by varying the parameters of pressure, cross-section, and length, the following relationships were identified:

- the relationship between bending stress intensity and maximum equivalent stresses. Shown in figure 7;
- the relationship between bending stress intensity values and maximum nodal displacement. The relationship is shown in figure 8.

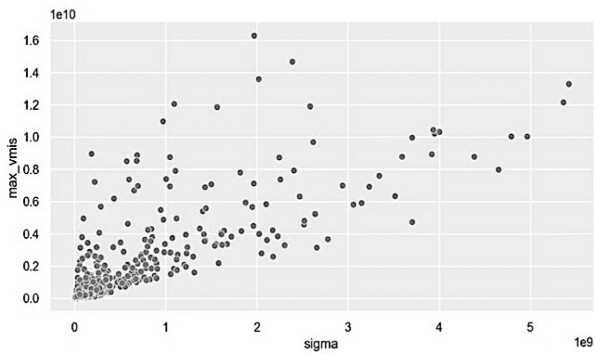

Fig. 7 – Example of the relationship between stress intensity values during bending (x-axis) and the maximum value of equivalent stresses (y-axis).

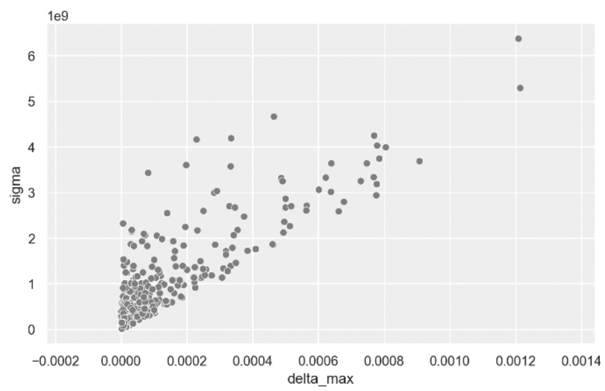

Fig. 8 – Example of the relationship between stress intensity values during bending (y-axis) and maximum nodal displacements (x-axis).

Based on the mentions above, drew the following conclusions. A specialized program for ANSYS was developed using the APDL API for data aggregation. After collecting the data and performing all necessary data processing, and having identified key dependencies between the parameters and key variables, three datasets were formed, each containing a sufficient amount of data to achieve the established research objective, namely the preparation of RM and NN for their subsequent comparison and tracking amount of time spent for aggregating necessary amount of data.

Model preparation. Based on the mathematical formulation of the problem, we designed two models each, in accordance with the research objective—to compare the prediction errors of RM and NN. The comparisons were conducted on progressively more complex data, grouped into datasets:

- a dataset with variable cross-sectional area characteristics of a beam element;
- a dataset with variable characteristics of the cross-sectional area and length of the beam element;
- a dataset with variable characteristics of the cross-sectional area, length of the beam element, and uniformly distributed pressure supported by the beam element.

The next step was to determine which approach would be used in the abstractions under study: networks or regression models. In the practice of designing artificial intelligence models, two distinct problems can be identified that the models are intended to solve:

- linear regression problems – predicting a scalar based on input parameters;
- logistic regression problems – classifying vectors according to input parameters, or the so-called Bernoulli probability.

Given the nature of the engineering problem, we can conclude that this problem falls under the category of linear regression problems—using the input parameters to predict the scalar nodal displacement and the maximum value of equivalent stresses. This type of problem has established solution approaches in both model variants—RM and NN.

Given the need for models to predict not just one but several values, it is necessary to create not simple NN and RM models, but so-called parametric surrogate models, which will include two independent generalized linear models for predicting each individual structural parameter.

Additionally, based on the input data—specifically the values of nodal displacements and maximum equivalent stresses—it is advisable to use special approaches for scaling the values. This approach aims to approximate the values to the origin of the coordinate system, which will accordingly decrease or increase the values of the dataset parameters. This scheme allows for a significant increase in the accuracy of the model's predictions if the values have a substantial range of variation, i.e., values that are too large or too small [13] The StandartScaler function from the sklearn library is used as the actor in this aspect [14]. Additionally, for node displacement values, an approach using logarithmic transformation, the so-called z-index (7), is employed.

$$z = \frac{y-\mu}{\sigma}, \tag{7}$$

where: y – the actual initial value of the vector;

μ – the mean of the logarithmic shift in the dataset;

σ – standard deviation of the logarithmic shift;

When creating a mathematical model in accordance with the conditions defined above—specifically, a linear regression model trained using a teacher—it is necessary to use a formal definition of the training and optimization of this model. The formal definition is described in Equation 8.

$$\theta \in \arg\min_{\theta} \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_\theta(x_i)) + \lambda\Omega(\theta), \tag{8}$$

where: $x_i \in \mathbb{R}^d$ – vector of parameters (input values) for the dataset;

y_i – a vector of key features (initial values) of the dataset;

L – loss function;

Ω – control function;

λ – regulation sensitivity coefficient.

The function for training the model using linear regression is described by the formula 9

$$f(x) = \omega^T x + b, \tag{9}$$

where: ω^T – weight vector;

x – vector of input parameters;

b – shift in values.

Creating a neural network requires the approach described in Equation 10. That is, summing the results of the activation functions across layers based on the input data and without directly modifying the activation function itself.y.

$$f_\theta(x) = f_L(f_{L-1}(\ldots f_1(x))),$$
$$f_l(h) = \sigma_l(W_l h + b_l), \tag{10}$$

where: W_l – weight matrix;

σ_l – layer activation function;

b_l – vector of value shifts;

$\theta = \{W_l, b_l\}_{l=1}^{L}$ – training settings.

Based on the formal definition of the model, it is necessary to select a software implementation for RM and NN to facilitate their accelerated and unified development and training. For the RM, the Python library sklearn was selected. This library provides a wide range of ready-made interfaces for interacting with various types of models, predefined optimization functions, error estimation functions, and more.

For creating neural networks, the PyTorch[15] and TensorFlow[16] frameworks were selected for comparison. The key characteristics identified include the ability to reuse model training and optimization descriptions, the formality of neural network specification, and the availability of an ecosystem of ready-made solutions for implementation. PyTorch offers a wide range of capabilities and flexible solutions, but provides more tools for low-level optimization for various types of network tasks and is used specifically for neural network research. TensorFlow, on the other hand, offers capabilities for working with high-level abstractions and a wide range of ready-to-implement solutions, supporting various types of neural networks. Given the above, TensorFlow is the optimal choice for creating neural networks.

The design and description of the RM, in accordance with Equation 6, also requires a loss function, which is intended to evaluate the accuracy of predictions regarding the input values and the output during training, testing, and direct application in the RM. The mean squared error function is used as the loss function for the RM. A formal description of the function is given in Equation 11.

$$L(w,b) = \frac{1}{N}\sum_{i=1}^{N}(y_i - (w^T x_i + b))^2, \qquad (11)$$

where: w^T – weight vector;

x_i – vector of input parameters;

y_i – the value of the output parameter (actual);

b – bias values.

The development of a neural network requires the definition of a specific architecture consisting of sequentially connected layers, where each layer contains a corresponding number of neurons and activation functions used for processing input data and generating output predictions. In addition, the training process requires the selection of both an optimization algorithm and a loss function.

In this study, the Mean Squared Error (MSE) function was selected as the loss function due to its suitability for regression-based engineering problems involving continuous output parameters. The formal definition of the MSE function is presented in Equation 12.

Several optimization approaches were analyzed in accordance with the characteristics of the investigated problem. In particular, the ADAM optimization algorithm and the least squares method were considered and compared. The selection criteria included the ability of the optimization method to efficiently explore the solution space, provide stable convergence during the training process, and ensure applicability for both regression models and neural network architectures. Such an approach allows a consistent optimization framework to be applied across all investigated model types.

Based on the obtained results, the ADAM optimization algorithm was selected due to its faster convergence rate compared to the least squares method, as well as its ability to achieve lower prediction error values during the training process. The mathematical formulation of the selected optimization algorithm is presented in Equation 13.

$$L_{MSE}(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_{fact} - y_{predict})^2, \qquad (12)$$

where: N – the number of input data vectors in the dataset;

y_{fact} – the value of the output parameter (actual);

$y_{predict}$ – the value of the predicted output parameter.

$$g_t = \nabla_\theta L(\theta_t),$$
$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t, \qquad (13)$$
$$\upsilon_t = \beta_2 \upsilon_{t-1} + (1-\beta_2){g_t}^2,$$

where: g_t – backpropagation gradient descent function;

m_t – exponential sliding average of the gradient;

$\beta_1, \beta_2 \in [0,1)$ – pulse decay time;

υ_t – root mean square exponential sliding of the gradient;`

Now that we have prepared the necessary components of the neural network, we will move on to using the dataset in accordance with the research objectives. Prior to beginning the model training process, the task datasets were divided into appropriate parts:

- Training set – data used directly to train the models (70% of the total);
- Test set – data used to verify the accuracy of the model's predictions (30% of the total data);

This approach, which involves splitting the data in this way, aims to create competitive conditions for training the model. In other words, the training and test sets are used directly during the creation and optimization of the model to verify its accuracy.

Having outlined the approach to splitting data for training and validating models, the next step is to set the key parameters according to the conditions of the problem under study. These conditions are the same for both types of models under study. The approach involves investigating the prediction accuracy of both model types under essentially similar key parameters and tracking their behavior with increasingly complex (less explicit) data. The following parameters were selected as input data for the models:

- aspect_ratio;
- inertia;
- width;
- height;
- length;
- total_force.

After running the training, the models produced the following predictions. Tables 3 and 4 present the results of the mathematical model's predictions for each of the datasets described above (Model No. 1 corresponds to Dataset No. 1, and so on).

Additionaly was tracked time alocation to learn regression model. It takes less than 5 sec. to learn single model which is preatty fast but regarding to moder laptop resources.

Table 3 – Accuracy of regression models in determining maximum node displacements

№ RM	Error value in RRMSE (Training set), %	Error value in RRMSE (Test set), %
Model № 1	>0,01	0,43
Model № 2	0,09	1,89
Model № 3	0,29	7,24

Table 4 – Accuracy of regression models in determining the maximum equivalent stress

№ RM	Meaning of RRMSE (Training set), %	Error value in RRMSE (Test set), %
Model № 1	0,01	0,62
Model № 2	>0,01	0,60
Model № 3	0,08	6,17

After training and testing the RM, similar steps were performed for the NN.

To design the network, we will refer to the data aggregation principle described above, which determines the number of layers and neurons in the NN. Based on the aggregated data, we have 4 hidden layers with 64 neurons in each layer. Based on this description, an MLP perceptron is formed. The design of the MLP is based on the idea of creating a surrogate model, which involves specifying two competing networks for the simultaneous determination of equivalent stresses and nodal displacements. In this case, the input layer will be common to both circuits, while the output layers will be defined separately.

After designing the MLP, activation functions for the hidden layers were selected. These functions influence the input parameters received from the previous layers of the network, along with the neuron's weight coefficients in a specific layer. Thus, a cascading effect occurs on the input data through the network layers. The ReLU function, described in Equation 14, was chosen as the activation function for the layers of the feedforward networks. For the output layer, the Linear activation function was used.

$$\mathrm{ReLU}(z) = \max(0, z),$$
$$z < 0 \rightarrow \mathrm{ReLU}(z) = 0, \quad (14)$$
$$z \geq 0 \rightarrow \mathrm{ReLU}(z) = z,$$

where: z – input parameters that are passed to the layer;

The epoch-based approach was chosen as the training method for the models. The idea behind epochs is to gradually optimize the model after the data has been fed into the model and the optimization function has been executed at the end of the epoch, with the optimization results carried over to the next epoch. Based on the complexity of the task and the volume of data, we set the number of epochs to 300.

From learning time perspective been spent about 18 sec. to learn single NN with modern laptop resources, which is not so important from process perspective.

Table 5 – Accuracy of neural networks at determining maximum node displacements

№ NN	Value RRMSE (Training set), %	Value RRMSE (Test set), %
Model № 1	3,51	3,24
Model № 2	7,94	8,85
Model № 3	14,98	15,02

Table 6 – Accuracy of neural networks in determining maximum equivalent stresses

№ NN	RRMSE Error value (Training set), %	RRMSE Error value (Test set), %
Model № 1	13.60	13,47
Model № 2	14,29	15,24
Model № 3	13,23	13,46

Additionally, as a part of verification process been prepared analytical solutions based on Cantilever beam under distributed pressure theory, which fits for FEM verification and works by Euler—Bernoulli bending beam approach for verification maximum equivalent stress by von Misses criterion which described at equation 15. For nodal displacement been taken Euler—Bernoulli bending beam theory with Timoshenko shear correction for rectangle beams described at equation 16.

$$\sigma_{max} = \frac{M_{max} * y}{I} = \frac{3\rho L^2}{h^2}, \quad (15)$$

where: M_{max} – bending moment produced by a uniformly distributed load over a beam,

I – second moment of area for a rectangular cross-section about its neutral axis.

$$\delta_b = \frac{3\rho L^4}{2Eh^2},$$
$$V(x) = q(L - x),$$
$$\delta_s = \frac{6\rho L^2(1+\nu)}{5Eh}, \quad (16)$$
$$\delta_{total} = \delta_s + \delta_b,$$

As results of this section of the study include the design of NN and RM models, the identification of the main mathematical functions used in the models for optimization, and the verification of prediction error values. It was determined how the input data will be divided prior to model training and which input parameters will be used during model training. Additionally, the accuracy of different types of models (NN and RM) was trained and investigated using prepared data, in accordance with the conditions of the tasks and the datasets associated with them. The prediction error of RM and NN on the training dataset was identified when determining the maximum nodal displacements and maximum equivalent stresses of the beam structure. Prepared analytical solution to verify obtained results from both models.

Verification of model predictions. After preparing the necessary datasets for numerical experiments, designing and training models in accordance with the tasks described above for determining the structural parameters, it is advisable to verify the accuracy of the models' predictions and compare the RM approaches themselves on the NN, namely, the accuracy of their predictions relative to the values of calculations performed in the ANSYS system, calculated by FEM numerical method and to compare it with analytical approach values which will indicate any kind of anomalies in prepared data and trained models. To ensure impartiality in the verification and to prevent issues related to pre-learned model responses, additional data collection based on prepared aggregation script, which calculates elements via FEM. Procedures were performed for data that the models had not previously encountered, corresponding to each of the investigated loading conditions and structural element fixations.

Figure 9 shows: the predicted values of maximum nodal displacements and maximum equivalent stresses for Problem No. 1 for regression models (square, rubin), neural networks (circle, triangle) and analytical approach (cross, plus) based on MSE. DISP – values of maximum nodal displacements, VMIS – values of maximum equivalent stresses.

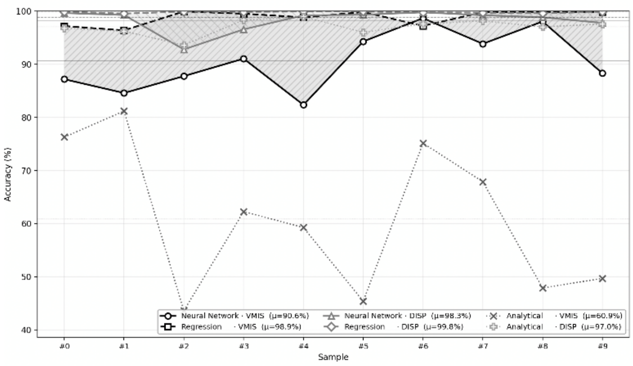

Fig. 9 – Graph of models predictions for Problem No.1

Figures 10 and 11 show the graphs of the predicted values from regression models, neural networks and analitical approach for problems No. 2 and No. 3, respectively.

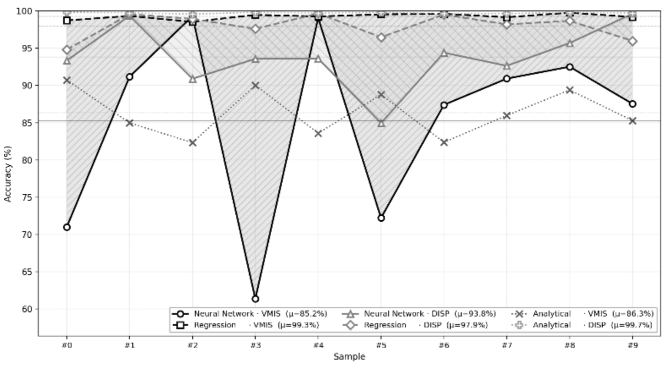

Fig. 10 – Graph of model predictions for Problem No. 2

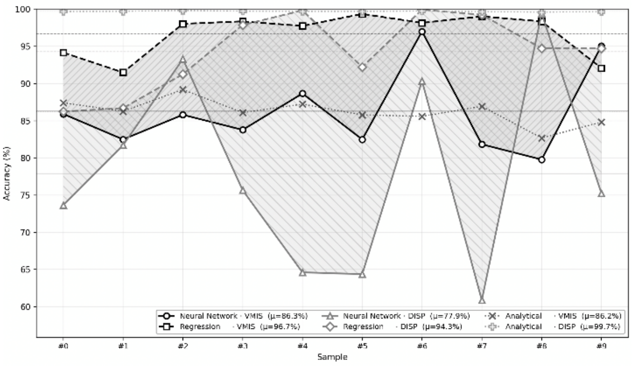

Fig. 11 – Graph of model predictions for Problem No. 3

Results analysis. After conducting a series of validation experiments on the RM and NN models, will draw conclusions regarding the magnitude of the prediction errors for these models.

The values average accuracy of regression model, neural networs and analitical solution for determing maximal equvaltent stress values by von Misses criteria presented at Figure 12. The accuracy average values of regression model, neural networs and analitical solution for determing maximal nodal diplacement values presented at Figure 13.

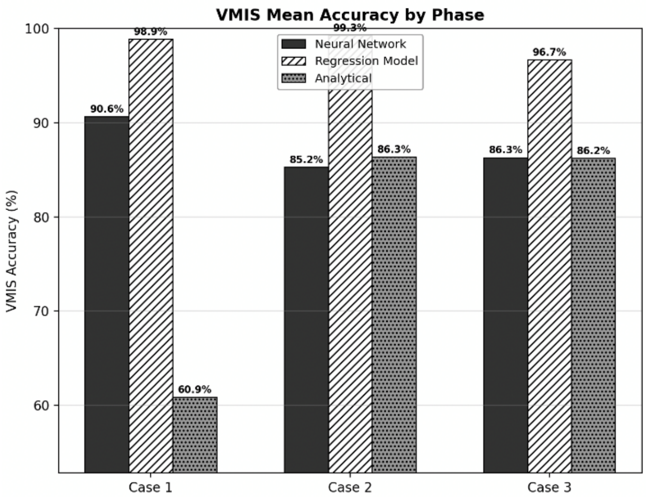

Fig. 12 – Charts of model's accuracy for von Misses criterion

Based on the data presented in charts few conclusions detected. Variety and compexity of data don't affect predictions by models dramatically. From accuracy perspective for problem No. 1 best fit provided by regression model, but from analitical model perspective might be determed as a overfiting for von Mises criteria values. Other point is that analitical solution showed 60% of accuracy on FEM prepared data which might inform about not clear or consistent data from beam proportions perspective for this specific problem.

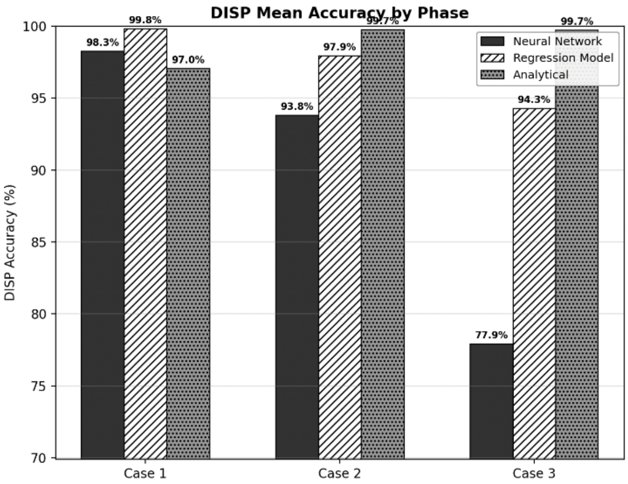

Fig. 13 – Charts of model's accuracy for nodal displacement

Although, for problem No. 2 and No. 3 obtained convergency between neural network model percent of accuracy and analitical solution, which describes that NN model does not trained with overfiting as a regression model which showed more than 95% accuracy for von Mises criteria.

For nodal displacement values, for problem №1 regression model shows overfiting, analitical approach indicates about inconsitent data. But for problem №2 and №3 obtained correct values and from this perspective regression model showed better accuracy than NN.

The obtained results indicate that further improvement in prediction accuracy requires a substantial increase in dataset size and greater variability of the training data. Based on the presented results, it can be concluded that NN models provide a more accurate approach for determination of maximum equivalent stresses according to the von Mises criteria structural analysis problems, but for maximum nodal displacements determination fit regression model approach. This picture shows additional impovements of aggregated data that necessary to prepare. Also need to mention about time spent on data preparation. From this perspective to generate datasets for all mentioned problem been spent from 6 to 8 hours which is not so effective and need to be improved as workflow. Other important part is the training time for regression model requiers less aboumt of time than a neural network

Conclusions. The objective of the presented study was to evaluate the applicability of artificial intelligence methods for predicting the strength characteristics of a structural beam element based on its parametric properties, including geometric parameters, material properties, and mechanical loading conditions. The investigated output parameters included the maximum equivalent stress according to the von Mises criterion and the maximum nodal displacement.

Within the scope of the research, six predictive models were developed, including three regression models (RM) and three neural network models (NN), in accordance with the defined research objectives. Numerical simulations were performed to determine the investigated structural characteristics, which enabled the implementation of a data aggregation procedure in ANSYS for the generation of the training dataset. After obtaining a sufficient amount of numerical data, the developed models were trained to solve the considered engineering problems using identical input parameters. The ADAM optimization algorithm was selected as the optimization method for model training, while the Mean Squared Error (MSE) function was applied as the loss function. The obtained results were additionally compared with analytical solution approaches.

After achieving satisfactory performance on the test dataset, an additional validation dataset was generated for further verification and comparative analysis of the developed models. Based on the obtained results, the predictive capabilities and accuracy of the investigated approaches were evaluated. The conducted analysis demonstrated that the regression models provided more accurate predictions of the maximum equivalent stress according to the von Mises criterion and the maximum nodal displacement compared with the investigated neural network models. Furthermore, considering the computational efficiency, training time allocation, and the indications obtained from the analytical solutions, the application of regression models optimized using the ADAM algorithm appears to be a more effective approach for solving the investigated class of structural mechanics problems than the considered neural network architectures.

The obtained results indicate that a перспективний напрямок further research is the investigation of structural elements characterized by pronounced nonlinear behavior under mechanical loading, particularly rotating shafts and similar engineering systems. Such problems may require the application of specialized neural network architectures incorporating physics-based loss functions instead of conventional mathematical formulations derived from classical shaft equations. In this context, Physics-Informed Neural Networks (PINNs) represent a promising approach, since they integrate governing physical laws directly into the training process and therefore may provide improved prediction accuracy for complex engineering systems and parametric structural analyses

Information about datasets. Datasets used for this article were prepared by authors for this study. For accessing to datasets contact garkusha2580@gmail.com.

Statement on Funding and Acknowledgments. This study did not receive any external funding.

Authors would like to thank Gennadii Martynenko, Doctor of Technical Sciences and professor at National Technical University "Kharkiv Polytechnic Institute" for his expert advice on applied mechanics.

Statement on Conflict of Interest. The authors declare that there is no conflict of interest.

Statement on the Use of Artificial Intelligence Tools In preparing this paper, the authors used ChatGPT 5.2 to edit the text of the article. All generated content has been reviewed and edited by the authors. The authors are solely responsible for the content of this publication.

References

[1] О. Зєнкевич, Р. Тейлор, Дж. Чжу. Метод скінченних елементів: основи та фундаментальні принципи. 7-ме вид. Волтем, США: Butterworth-Heinemann, 2013. 714 p. doi: https://doi.org/10.1016/C2009-0-24909-9.

[2] М. Дж. Сімпсон, О. Дж. Макларен. Прогнозування з використанням недостатньо ідентифікованих математичних моделей, *Бюлетень математичної біології*. 2024. Vol. 86. Art. no. 80. doi: https://doi.org/10.1007/s11538-024-01294-0.

[3] Р. Дж. Мерфі, О. Дж. Макларен, М. Дж. Сімпсон. Реалізація моделей похибок вимірювання з механістичними математичними моделями в межах правдоподібнісного підходу для оцінювання, аналізу ідентифікованості та прогнозування в науках про життяф, *Журнал інтерфейсу Королівського товариства*. 2024. Vol. 21, no. 210. Art. no. 20230402. doi: https://doi.org/10.1098/rsif.2023.0402.

[4] Д. Круз, І. Бастос, А. Сілва, Дж. Манфредіні Вассолер, Ф. Штумпф, А. Круз Джуніор, Т. Карвалью Мартінс, Ф. Клайн, К. Гільєрме. Моделювання динамічних циклів у морських швартових волокнах: модифіковані математичні моделі та залежність від інваріантів механіки суцільного середовища, *Вектор – журнал точних та інженерних наук*. 2025. Vol. 35. Art. no. e18256. doi: https://doi.org/10.63595/vetor.v35i1.18256.

[5] М. Юс, Л. Бартошова. Математична модель тертя, *Ключові інженерні матеріали*. 2025. Vol. 1017. P. 11–17. doi: https://doi.org/10.4028/p-5CEhr1.

[6] П. С. Дасанаяке, В. Баранаускас, Г. Дервініс, Л. Баласевічюс. Огляд математичних моделей у робототехніці, *Прикладні науки*. 2025. Vol. 15, no. 14. Art. no. 8093. doi: https://doi.org/10.3390/app15148093.

[7] Г. Ю. Мартиненко, Л. В. Розова. Комп’ютерне моделювання елементів конструкцій та визначення їх міцності при статичних навантаженнях: навчальний посібник. Харків: НТУ «ХПІ», 2021. 242 с.

[8] PyAnsys. PyMAPDL, дата звернення: 24.02.2026. [Електронний ресурс]. Режим доступу: https://mapdl.docs.pyansys.com.

[9] Д. Й. Чен. Pandas для всіх. Бостон, США: Addison-Wesley Professional, 2022. 512 p.

[10] Дж. Мулайїл. Вивчення Keras для глибоких нейронних мереж: швидкий підхід до сучасного глибокого навчання з Python. 2-ге вид. Нью-Йорк, США: Springer, 2019. P.192, doi: https://doi.org/10.1007/978-1-4842-4240-7.

[11] М. Кун, К. Джонсон. Інженерія ознак та їх відбір: практичний підхід до прогнозних моделей. Бока-Ратон, США: Chapman & Hall/CRC Press, 2019. p.310, doi: https://doi.org/10.1201/9781315108230..

[12] С. Оздемір. Практикум з інженерії ознак. Шелтер-Айленд, США: Manning Publications, 2022. 272 p.

[13] Р. Ф. Де Мелло, М. А. Понті. Машинне навчання: практичний підхід до теорії статистичного навчання. Шам, Швейцарія: Springer, 2018. 362 p.

[14] Scikit-learn. Документація StandardScaler, дата звернення: 24.02.2026, [Електронний ресурс]. Режим доступу: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[15] Е. Стівенс, Л. Антіга, В. Томас. Глибоке навчання з PyTorch. Шелтер-Айленд, США: Manning Publications Co., 2020. 491 p.

[16] А. Сенгупта, Б. Зоф, Дж. Шленс. Програмування з TensorFlow: рішення для застосувань периферійних обчислень. Шам, Швейцарія: Springer, 2021. 190 p.

References (transliterated)

[1] O. Zienkiewicz, R. Taylor, and J. Zhu, The Finite Element Method: Its Basis and Fundamentals, 7th ed. Waltham, MA, USA: Butterworth-Heinemann, 2013, p. 714, doi: https://doi.org/10.1016/C2009-0-24909-9.

[2] M. J. Simpson and O. J. Maclaren, “Making predictions using poorly identified mathematical models,” Bull. Math. Biol., vol. 86, art. no. 80, 2024, doi: https://doi.org/10.1007/s11538-024-01294-0.

[3] R. J. Murphy, O. J. Maclaren, and M. J. Simpson, “Implementing measurement error models with mechanistic mathematical models in a likelihood-based framework for estimation, identifiability analysis and prediction in the life sciences,” J. R. Soc. Interface, vol. 21, no. 210, art. no. 20230402, 2024, doi: https://doi.org/10.1098/rsif.2023.0402.

[4] D. Cruz, I. Bastos, A. Silva, J. Manfredini Vassoler, F. Stumpf, A. Cruz Junior, T. Carvalho Martins, F. Clain, and C. Guilherme, “Dynamic cycle simulation in offshore mooring fibers: Modified mathematical models and dependence on invariants of continuum mechanics,” Vetor – Rev. Cienc. Exatas Eng., vol. 35, art. no. e18256, 2025, doi: https://doi.org/10.63595/vetor.v35i1.18256.

[5] M. Jus and L. Bartošová, “Mathematical model of friction,” Key Eng. Mater., vol. 1017, pp. 11–17, 2025, doi: https://doi.org/10.4028/p-5CEhr1.

[6] P. S. Dasanayake, V. Baranauskas, G. Dervinis, and L. Balasevicius, “A review of mathematical models in robotics,” Appl. Sci., vol. 15, no. 14, art. no. 8093, 2025, doi: https://doi.org/10.3390/app15148093.

[7] G. Y. Martynenko, L. V. Rozova., Kompiuterne modeliuvannia elementiv konstruktsii ta vyznachennia yikh mitsnosti pry statychnykh navantazhenniakh: navchalnyi posibnyk [Computer modeling of structural elements and determination of their strength under static loads: a textbook]. Kharkiv, Ukraine: NTU “KHPI” Publ., 2021, p. 242.

[8] PyAnsys: PyMAPDL, accessed: Feb. 24, 2026. [Online]. Available: https://mapdl.docs.pyansys.com.

[9] D. Y. Chen, Pandas for Everyone. Boston, MA, USA: Addison-Wesley Professional, 2022, p. 512.

[10] J. Moolayil, Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python, 2nd ed. New York, NY, USA: Springer, 2019, p. 192, doi: https://doi.org/10.1007/978-1-4842-4240-7.

[11] M. Kuhn and K. Johnson, Feature Engineering and Selection: A Practical Approach for Predictive Models. Boca Raton, FL, USA: Chapman and Hall/CRC Press, 2019, p. 310, doi: https://doi.org/10.1201/9781315108230.

[12] S. Ozdemir, Feature Engineering Bookcamp. Shelter Island, NY, USA: Manning Publications, 2022, p. 272.

[13] R. F. De Mello and M. A. Ponti, Machine Learning: A Practical Approach on the Statistical Learning Theory. Cham, Switzerland: Springer, 2018, p. 362.

[14] Scikit-learn StandardScaler Documentation, accessed: Feb. 24, 2026 [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html.

[15] E. Stevens, L. Antiga, and V. Thomas, Deep Learning with PyTorch. Shelter Island, NY, USA: Manning Publications Co., 2020, p. 491.

[16] Sengupta, B. Zoph, and J. Shlens, Programming with TensorFlow: Solutions for Edge Computing Applications. Cham, Switzerland: Springer, 2021, p. 190.p.

Надійшла (received) 14.04.2026

Прийнята до друку (accepted) 28.05.2026

Опублікована (published) 29.05.2026

Відомості про авторів та їх внесок / About The Authors And Their Contributions

Владислав Гаркуша (Vladyslav Harkusha) – Національний технічний університет «Харківський політехнічний інститут», аспірант кафедри математичного моделювання та інтелектуальних обчислень в інженерії; м. Харків, Україна; ORCID: https://orcid.org/0000-0001-5980-4802; email: Vladyslav.Harkusha@infiz.khpi.edu.ua

Автор ознайомився з остаточною версією рукопису та погодився з її публікацією.