DOI: https://doi.org/10.15276/hait.05.2022.15
UDC 004.85:539.3

Application of computational intelligence methods for the heterogeneous material stress state evaluation

Ruslan A. Babudzhan[1)]
ORCID: https://orcid.org/0000–0001–5765–9234; ruslanbabudzhan@gmail.com
Oleksii O. Vodka[1)]
ORCID: https://orcid.org/0000–0002–4462–9869, oleksii.vodka@gmail.com. Scopus Author ID: 56239259600
Mariia I. Shapovalova[1)]
ORCID: https://orcid.org/0000–0002–4771–7485; mishapovalova@gmail.com. Scopus Author ID: 57212000432
[1)] National Technical University "Kharkiv Polytechnic Institute", 2, Kyrpychova Str. Kharkiv, 61002,Ukraine

ABSTRACT

The use of surrogate models provides great advantages in working with computer-aided design and 3D modeling systems, which opens up new opportunities for designing complex systems. They also allow us to significantly rationalize the use of computing power in automated systems, for which response time and low energy consumption are critical. This work is devoted to the creation of a surrogate model for approximating the finite element solution of the problem of dispersion–strengthened composite plane sample deformation. An algorithm for constructing a parametric two–dimensional model of a composite is proposed. The calculation model is created using the ANSYS Mechanical computer-aided design and analysis program using the APDL scripting model builder. The parameters of the stress-strain state of the material microstructure are processed using a convolutional neural network. A neural network based on the U–Net architecture of the encoder-decoder type has been created to predict the distribution of equivalent stresses in the material according to the sample geometry and load values. A direct sequence of layers is taken from the specified architecture. To increase the speed and stability of training, the type of part of the convolutional layers has been changed. The architecture of the network consists of serially connected blocks, each of which combines layers such as convolution, normalization, activation, subsampling, and a latent space that connects the encoder and decoder and adds load data. To combine the load vector, such a neural network architecture as a concatenator is created, which additionally includes the Dense, Reshape and Concatenate layers. The model loss function is defined as the root mean square error over all points of the source matrix, which calculates the difference between the actual value of the target variable and the value generated by the surrogate model. Optimization of the loss function is performed using the first–order gradient local optimization method ADAM. The study of the model learning process is illustrated by plots of loss functions and additional metrics. There is a tendency for the indicators to coincide between the training and validation sets, which indicates the generalizing capability of the model. Analyzing the output of the model and the value of the metrics, a conclusion is made about the sufficient quality of the model. However, the values of the network weights after training are still not optimal in terms of minimizing the loss function. And also, to accurately reproduce the solution of the finite element method (FEM), the proposed model is quite simple and requires clarification. The speed comparison of obtaining results by the FEM and using the architecture of the neural network is proposed. The surrogate model is significantly ahead of the FEM and is used to speed up calculations and determine the overall quality of the approximation of problems of mechanics of this type.

Keywords: Convolutional neural network; stress–strain state; finite element method; surrogate model; U–Net; encoder–decoder

For citation**: Babudzhan R. A., Vodka O. O., Shapovalova M. I. Application of computational intelligence methods for the heterogeneous material stress state evaluation. *Herald of Advanced Information Technology*. 2022; Vol. 5 No. 3: 198–209. DOI: https://doi.org/10.15276/hait.05.2022.15**

INTRODUCTION

Machine learning is getting deeper into various industries and is widely used especially for pattern and image recognition, natural language processing, optimization of operations, data mining, and knowledge discovery. Systems built on machine learning algorithms are used to approximate solutions to differential or variational equations. Simplified solutions for finding the physical properties of structures (e.g. strains, stresses) have been used in industry for many years. However, such solutions are limited in the configuration of structural elements and do not have sufficient accuracy, forcing designers to be more conservative when choosing technological solutions.

At the same time, intelligent approximation methods based on data (surrogate models) [1] have unlimited possibilities for their complication and approximation to the original mathematical model, optimizing the similarity function between the approximated solution and the "classical" one, obtained, for example, using the finite element method (FEM). An appropriate model can replace the use of FEM with a certain accuracy, given enough training data.

1. ANALYSIS OF LITERARY DATA

There are many studies related to the creation of surrogate models for modeling structural mechanics with machine learning (ML) or deep learning (DL) approaches that confirm the feasibility of supervised learning models trained using FEM modeling [2, 3], [4, 5]. The authors of the study [6] propose to predict the reaction of a projectile after impact with steel armor using MLP, while their surrogate mode is trained with simulation data. The analysis of generalizing surrogate models for three-dimensional farms using MLP and FEM training data is considered in [7]. A different design of hot forged products with ML trained on FEM data obtained using the commercial DEFORM software is dealt with by the author in [8]. In [9], the authors use single-layer feed-forward neural networks instead of FEM as a metamodel in the Sequential Approximate Optimization (SAO) algorithm. Prediction of the velocity field and the location of the neutral point of cold flat rolling using MLP, trained on the results of modeling hard plastic finite element analysis, is described by the authors in [10]. In [11], the mechanical properties of a two–dimensional composite are estimated using a convolutional neural network (CNN) trained on the results of the FEM. The authors of [12] use Gaussian process regression (GPR) in their approach.

The disadvantages of surrogate models include, for example, the fact that generalization to hidden data is achieved only by discretization of the computational domain, exclusively in one use case [13]. Also, it may not be feasible to replicate published experiments because important parameters such as the number of finite elements, finite element type, discretization method, and ML model hyperparameters such as loss or activation functions are not reported.

Despite the disadvantages, surrogate models provide significant opportunities for computer-aided design and 3D modeling systems, provide new capabilities in the design of complex systems, and also contribute to a significant rationalization of the use of computing power in automated systems for which response time and low power consumption are critical.

2. THE PURPOSE AND OBJECTIVES OF THE RESEARCH

The main goal of the work is to create a surrogate model for approximating a complex finite element (FE) solution to the problem of determining the stress-strain state (SSS). The study is based on the approximation of a plane problem solution of elasticity theory for a composite material representative sample with a chaotic arrangement of inclusions.

To achieve the set goals, the following steps are proposed:

- to create a computational deformation model of a representative sample of a dispersion–reinforced composite material;
- to get a set of solutions for SSS of a certain number of arbitrary configurations of composite material samples and form a data set to create a surrogate model;
- to determine the architecture of a surrogate model based on a neural network to approximate the distribution of equivalent stresses using the finite element method;
- to train the neural network to obtain a surrogate model for approximating the results;
- to evaluate the error of the solutions by comparing the solutions obtained using the finite element method and the surrogate model.

A sample of a dispersion–reinforced composite material is shown in Fig. 1. In this case, the following boundary conditions are set: the element is fixed on the left and bottom faces; loads – in the form of displacements applied to the upper and right faces in the direction that varies. The size and number of inclusions vary, and the concentration of inclusions varies from 10 % to 30 %. The matrix material is isotropic, the inclusion material is orthotropic. The parameters of the matrix material and inclusions are given in Table 1 and Table 2, respectively.

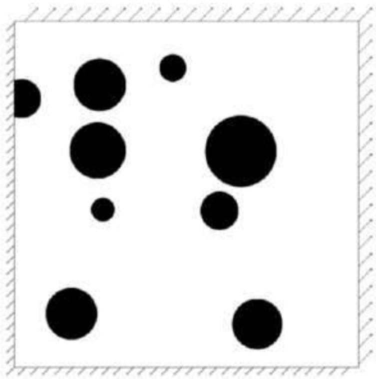


Fig. 1. **An example of a composite sample**
Source: **compiled by the authors**

Table 1. **Matrix parameters**

Material	Modulus of elasticity *E*, *GPa*	Poisson's ratio *ν*
Ferrite	180	0.35

Source: **compiled by the [14]**

Table 2. **Parameters of inclusions**

Material	c_{11}, c_{22}	c_{12}	c_{13}, c_{23}	c_{33}	c_{44}, c_{55}	c_{66}
	GPa					
Graphite	1060	290	109	46.6	2.3	385

Source: **compiled by the [14]**

3. CONSTRUCTION OF A SURROGATE MODEL

Creation of a dataset. Finite element model

In the first stage, a model of the composite material is creating (Fig. 1). The sample is a square two–dimensional plate with inclusions in the form of circles. To calculate the stress-strain state, the model is fixed on two adjacent edges, and a load is applied to two opposite edges. As a load, the displacements of the ribs along the X and Y axes are taken, while the magnitude of the displacements is a constant value.

The calculation model is created based on the work [14, 15], [16, 17], [18], using ANSYS Mechanical CAD and analysis tools [19], using the APDL scripting model building tool [20], and has the following steps:

- creating a model of matrix materials and inclusions;
- construction of circles, which are inclusions;
- construction of a frame according to the size of the sample;
- creation of the matrix area due to the extrusion of inclusions (Fig. 2);
- division of geometry into finite elements;
- application of boundary conditions to sample edges (Fig. 3);
- solving a system of algebraic equations.

The proposed sequence of actions allows you to create a calculation model and obtain a solution for the stress-strain state of the system in displacements for various configurations of inclusions and loads.

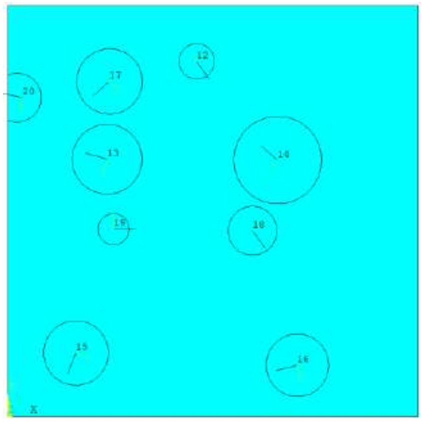


Fig. 2. **The geometry of the matrix and inclusions**
Source: **compiled by the authors**

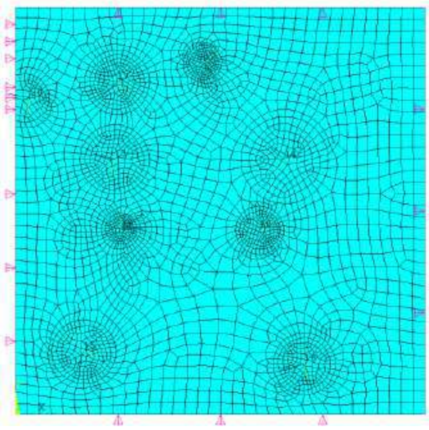


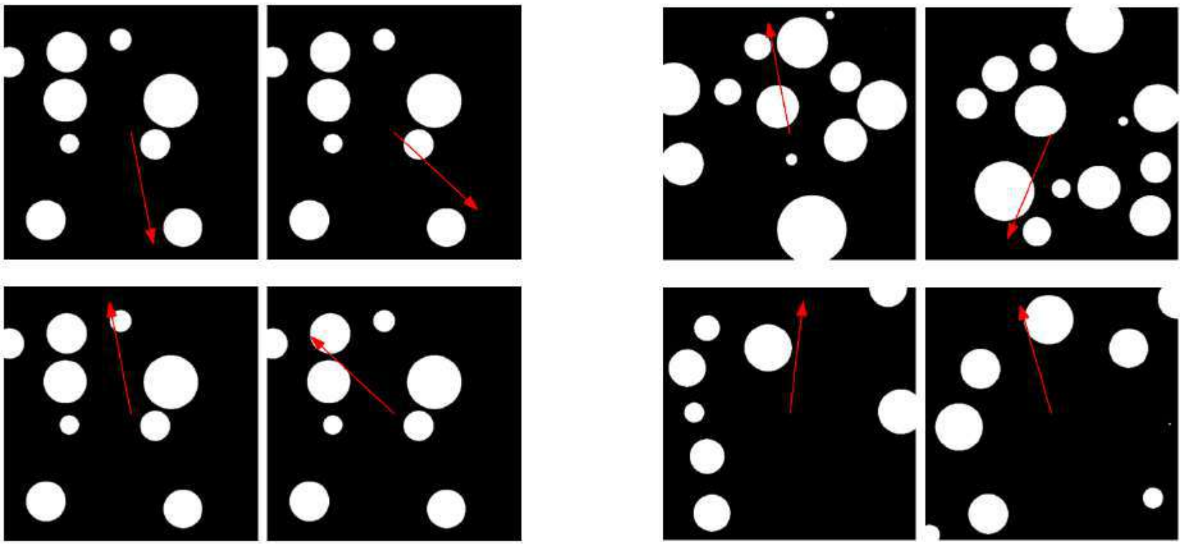
Fig. 3. **Finite element mesh and loads on the model**
Source: **compiled by the authors**

Generation of a data set

The collection and subsequent processing of the calculation results are carried out automatically based on the above sequence of operations. This allows the formation of a data set for further training of the neural network. The required number of examples for training is set to 10000. At the same time, it is known from the theory of the finite element method that the formation of the stiffness matrix is the most costly action in the process of system analysis. Therefore, 1000 different samples of composite material are formed in the work, and tension/compression is applied to each of them in 10 different directions. Such a scheme is implemented using the Load Step functionality in the Ansys Mechanical software package.

Load direction variations are shown in Fig. 4, where in case a – different directions of loads for one sample are shown, and on b – different directions of loads are shown for different samples.

The algorithm for collecting and post-processing the results is shown in Fig. 5 and Fig. 6. The construction of the composite geometry is based on the developed method for generating a statistically equivalent artificial microstructure of cast iron [14]. According to the paper, it is assumed that the size distribution of inclusions obeys the normal law, and the placement of inclusions on the plane is realized by the function of uniform distribution of the value and occurs randomly. This allows a sufficient number of samples to be generated for analysis. Here $|F|$ – is the displacement modulus, is a constant and is chosen according to [14]. The creation of a data set is implemented using ANSYS. At each overload step, a displacement



a b

Fig. 4. **Changing the direction of the load:**
a – for one sample; b – for different samples
***Source:* compiled by the authors**

vector is formed and applied to the corresponding finite element nodes. The initial bias is obtained from the uniform distribution. The SSS calculation is performed simultaneously for all load steps for one generated sample geometry.

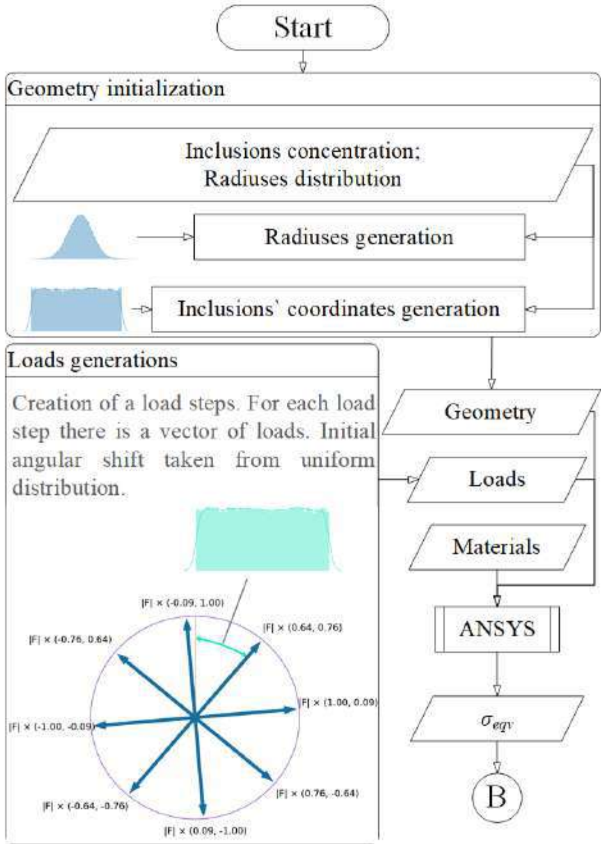


Fig. 5. **Creation of a data set.**
Scheme A – obtaining a solution in the ANSYS
***Source:* compiled by the authors**

After the automatic solution of the equations system, the stresses inside the finite elements are calculated from the obtained displacements of each node of the structure. The calculation results for 10 load steps of one sample are shown in Fig. 7.

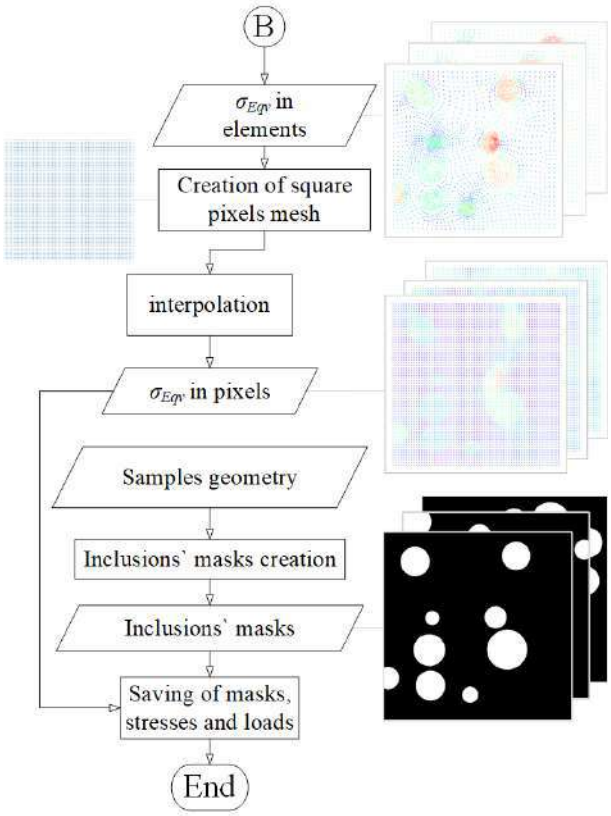


Fig. 6. **Receiving a data set.**
Scheme B – further data processing
***Source:* compiled by the authors**

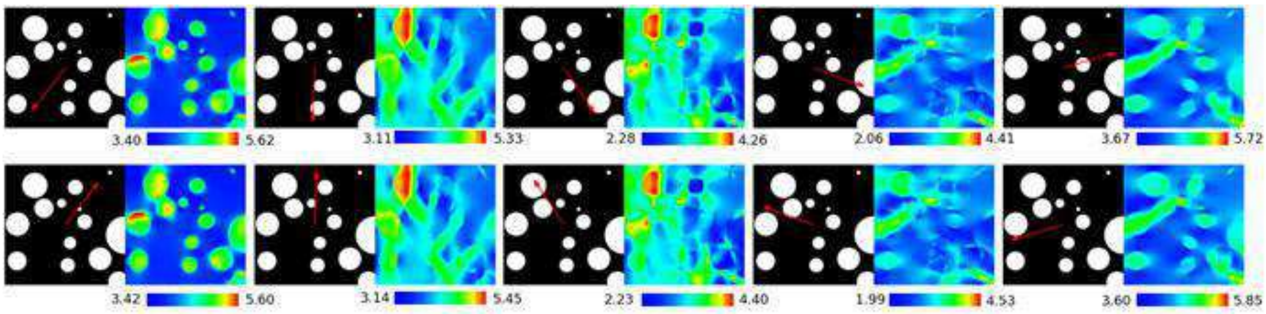


Fig. 7. **Calculations of the stress state for 10 loading steps of one sample. Equivalent stresses, *GPa***
***Source:* compiled by the authors**

The formation of an algorithm for constructing a computational model and subsequent processing of the results is carried out using the Python programming language, which provides opportunities for creating software tools and analyzing data (Fig. 6). The convolutional neural network algorithm takes structured matrices as input, so stress data imported into Python requires further processing. Suitable datasets are restructured by cubic spline interpolation [21]. At the output, stresses are obtained at the centers of the pixels.

When working with neural networks, the geometry of each sample needs to be rasterized to get a binary mask that is applied to the input of the neural network. The load vector data array corresponding to each load distribution is concatenated with the latent space of the model.

Neural network creation

A neural network for predicting the von Mises equivalent stress distribution during tension/compression of a two-dimensional composite sample is implemented in the Python programming language using the TensorFlow open-source deep learning library [22].

To achieve the set goals, an encoder-decoder type model is chosen, where the encoder “compresses” the input data array into a latent (limited in size) space; the decoder – "reveals" the feature vector compressed into the latent space to obtain a modified data array at the output The serial connection of the encoder-decoder model in combination with the algorithm for simultaneously optimizing the weights of the neurons of both networks makes it possible to obtain a neural network, which is a surrogate model for searching for the SSS of a composite sample.

The model architecture used in this work is based on the U-NET architecture [23]. One of the features of U-NET is the duplication of the feature map of each block of layers. In this case, one copy goes “deeper” through the network, and the other one joins the symmetrical decoder block, bypassing the deeper blocks of the network. This feature is not implemented in this work due to the impossibility of concatenating these loads into large feature maps without a significant increase in the calculation time. Thus, a direct sequence of layers is taken from the named architecture. Among the changes made to the architecture, it should be noted a change in the type of convolutional layers to increase the speed and stability of training.

The encoder and decoder architectures are shown in Fig. 8 and Fig. 9 respectively.

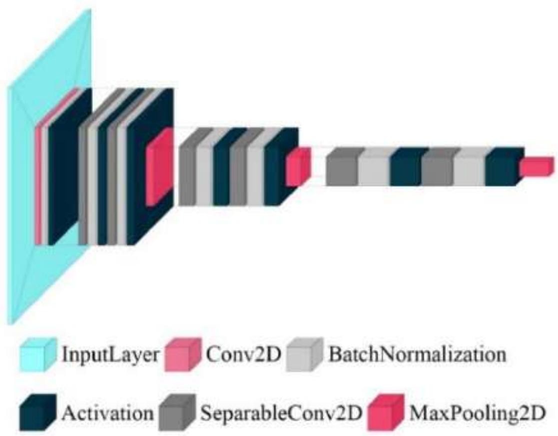


Fig. 8. **Architecture of the neural network. Encoder**
***Source:* compiled by the authors**

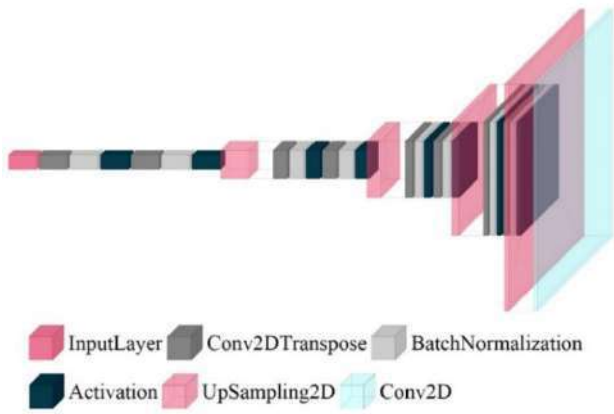


Fig. 9. **Architecture of the neural network. Decoder**
***Source:* compiled by the authors**

ISSN 2663–0176 (Print)
ISSN 2663–7731 (Online)

The network architecture consists of a series–of connected blocks, each of which contains the following layers:

1) convolution (Conv2D, SeparableConv2D, Conv2DTranspose). These layers are different implementations of the idea of convolutional layers. Conv2D is a regular convolutional layer, SeparableConv2D is a layer that implements the convolution function by superimposing two 1D convolution kernels instead of one 2D kernel. Conv2DTranspose – a convolution layer with a dilation factor of more than 1 (implements a “sweep”);
2) batch normalization;
3) activation. Activation function overlays;
4) subsampling (MaxPooling2D, UpSampling2D). Layers that implement feature map scaling (2x reduction in case of MaxPooling2D and increase in case of UpSampling2D).

At the input to the neural network – InputLayer (Fig. 8), a mask representing the geometry of the composite sample is supplied. The input block receives an image of 256×256 pixels, which has one channel, while the encoder dimension is (256, 256, 1). The initial dimension of the decoder is respectively (256, 256, 1) – at the output, the decoder creates a matrix of equivalent stresses 256×256.

The last element of the network architecture is the latent space connecting the encoder and decoder and adding load data. The load vector has the dimension (2, 1) and is represented by the vector of node displacements of the specimen edges under the action of the load.

The output of the encoder, as well as the input of the decoder, have the dimension (16, 16, 256), that is, 256 feature maps with a size of 16×16 pixels. To combine the load vector of the dimension

of the latent space, a concatenator is created (Fig. 10). The input images are 256×256, so there are 16×16 in the latent space of the feature map. The architecture of the concatenator is explained by the need to create geometry feature maps and load feature maps of the same size. Such an element of the network architecture, in addition to the already indicated layers, also has Dense, Reshape and Concatenate layers. At the same time, the first block of the developed architecture expands the input vector into one feature map. Dense is an ordinary fully connected layer, the input to which is a vector of dimension (2, 1), and the output has dimension (256, 1). Reshape – used to change the dimension of the space to (16, 16, 1).

In the next step, the first block is combined with the convolutional block (without sub-describing). The output of the last layer has the dimension (16, 16, 32), subsequent concatenation with the output of the encoder changes the dimension to (16, 16, 288). With the help of Conv2DTranspose, the layers are folded into a latent space of dimensions (16, 16, 256), into which the input data is encoded.

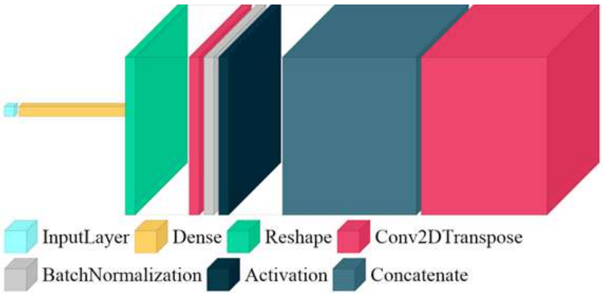


Fig. 10. **Architecture of the neural network. Latent space concatenation**
Source: **compiled by the authors**

After the initialization of the neural network, the loss function and target metrics are determined. The loss function is a function that calculates the difference between the actual value of the target variable (in our case, the stress distribution calculated using ANSYS) and the value generated by the surrogate model.

For the loss function, the root–mean–square error (RMSE) is selected for all points of the original matrix (1):

$$RMSE = \sqrt{\frac{1}{256\times256}\sum_{i=1}^{256}\sum_{j=1}^{256}\left(y_{ij}-\widehat{y_{ij}}\right)^2}. \quad (1)$$

Thus, neural network training occurs by minimizing the mean square error. At each iteration of network training, the values of the functions are also calculated: the root-mean-square difference ($RMSE_{max}$) of the maxima of equivalent stresses (2) and the root-mean-square error at 80 % ($RMSE_{80\%}$) of the quantile (3).

The considered metrics are defined as the difference between the actual and model-generated stress values:

$$RMSE_{max} = \sqrt{\left(y_{max}-\widehat{y_{max}}\right)^2}, \quad (2)$$

$$RMSE_{80\%} = \sqrt{\frac{1}{256\times256}\sum_{i=1}^{256}\sum_{j=1}^{256}\left[\left(y_{ij}-\widehat{y_{ij}}\right)^2\times\theta\left(y_{ij}-Q_{80}(y)\right)\right]}, \quad (3)$$

where $Q_{80}(y)$ – 80 % quantile of the actual stress distribution; θ – the Heaviside function (4) (Fig. 11):

$$\theta(x-a) = \begin{cases} 0, x < a \\ 1, x \geq a \end{cases}. \quad (4)$$

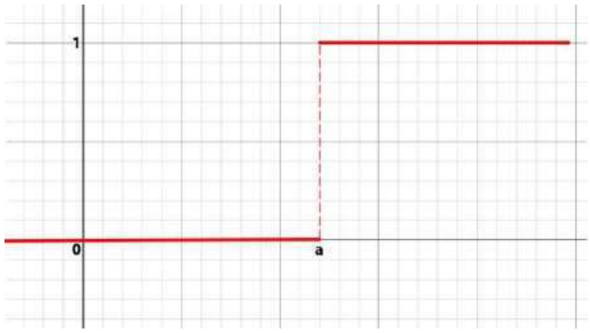


Fig. 11. **Heaviside function**
***Source:* compiled by the authors**

Using the first-order gradient local optimization method (ADAM) [24], the cost function is optimized. At each iteration of the optimization of the network weights, this method takes into account the exponential damping of the gradient (5) and the square of the gradient (6) in previous iterations.

The change in weights occurs according to (7):

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \quad (5)$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) {g_t}^2, \quad (6)$$

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t}} m_t, \quad (7)$$

where β_1 and β_2 – attenuation coefficients for calculating the average gradient and the average square of the gradient, respectively; η – gradient descent step; w_t – the value of the network weights at the current iteration; g_t – gradient over the current iteration.

Attenuation coefficients are algorithm hyperparameters that can be optimized, but for this algorithm, the standard values defined by its developers usually work at a sufficient level.

To train the neural network, a sample of 10,000 examples is divided into three sub-samples: training for direct network training (6000 els.), validation – for controlling the cost function and metrics (2000 els.), test sub-sample – for the final assessment of the quality of the model (2000 els.).

Network training takes place over 30 epochs, which corresponds to 30 passes of the test dataset through the network. Each epoch takes an average of 270 *seconds*, and network weight optimization takes 135 *minutes*.

4. INVESTIGATION OF THE MODEL LEARNING PROCESS

The course of training a surrogate neural network model for determining the equivalent stresses of a flat composite sample is illustrated in the graphs of the dependence of loss functions and additional metrics (Fig. 12, Fig. 13 and Fig. 14).

Fig. 12. **The loss function on the training and validation samples**
***Source:* compiled by the authors**

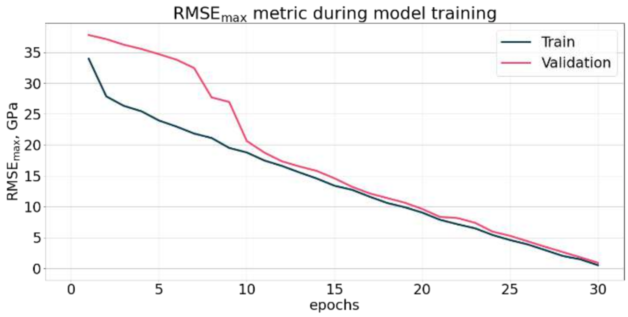


Fig. 13. **RMSE$_{max}$ on the training and validation samples**
***Source:* compiled by the authors**

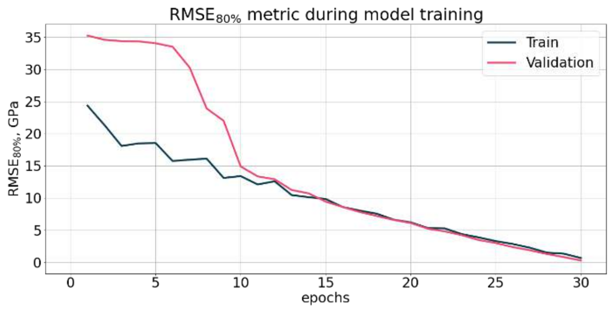


Fig. 14. $\boldsymbol{RMSE_{80\%}}$ **on the training and validation samples**
***Source:* compiled by the authors**

There is a tendency for the indicators to coincide between the training and validation samples. According to the theory of deep learning, this behavior of indicators indicates the generalizing the capability of the model. Therefore, prediction on unknown data (on the validation set) occurs with the same quality as on known data (on the training set).

The obtained results (Fig. 12) indicate that the intensity of the decrease in the loss function is constant. This can serve as an indicator that the gradient method has not reached a local minimum and further optimization of the network weights can

ISSN 2663–0176 (Print)
ISSN 2663–7731 (Online)

improve the results. Approaching a local minimum is displayed on the graph of the loss function as a decrease in the rate of change of the function on the training sample. If this does not happen, it is necessary to increase the number of epochs.

It is a predicted field of equivalent stresses on the training set. It also helps to determine whether the developed model architecture is suitable for solving the main goal of the problem. Comparing the output of the model and the value of the metrics with acceptable ones, it is concluded that the quality of the model is sufficient.

5. THE RESULTS OF THE MODEL

A graphic representation of the geometry of the samples, the direction of tension/compression, as well as the actual and predicted stress distribution for the training set are shown in Fig.15. The data for the validation and test sets are shown in Fig.16 and Fig.17.

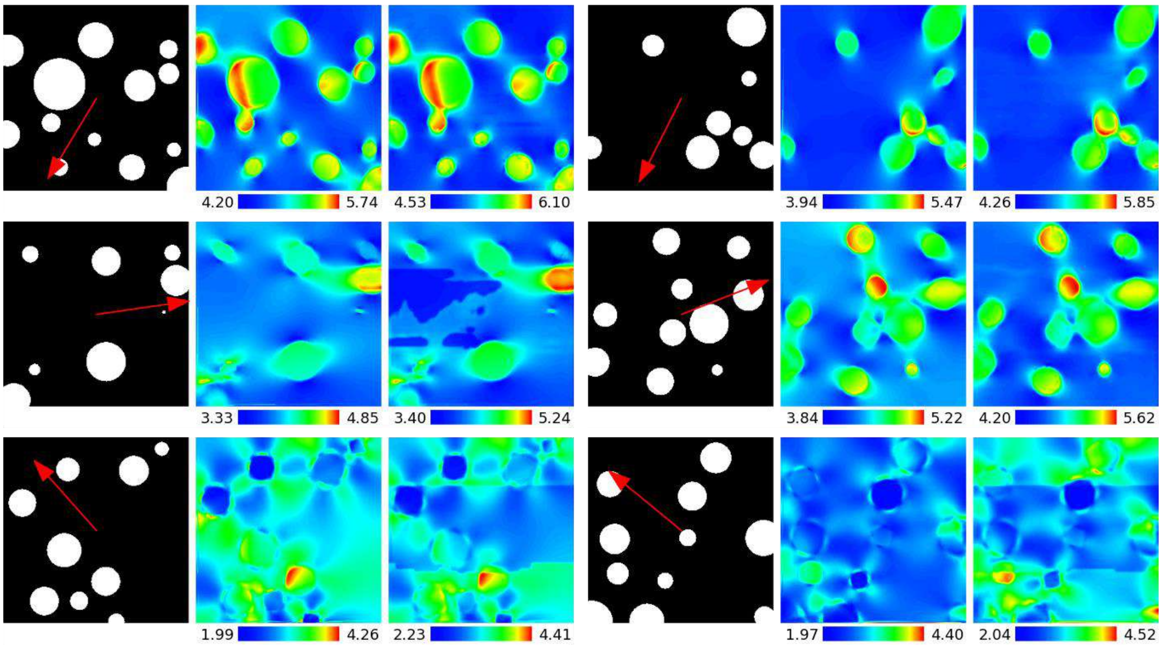


Fig. 15. **Sample geometry, MCE solution, and surrogate model output. Training sample, *GPa***
***Source:* compiled by the authors**

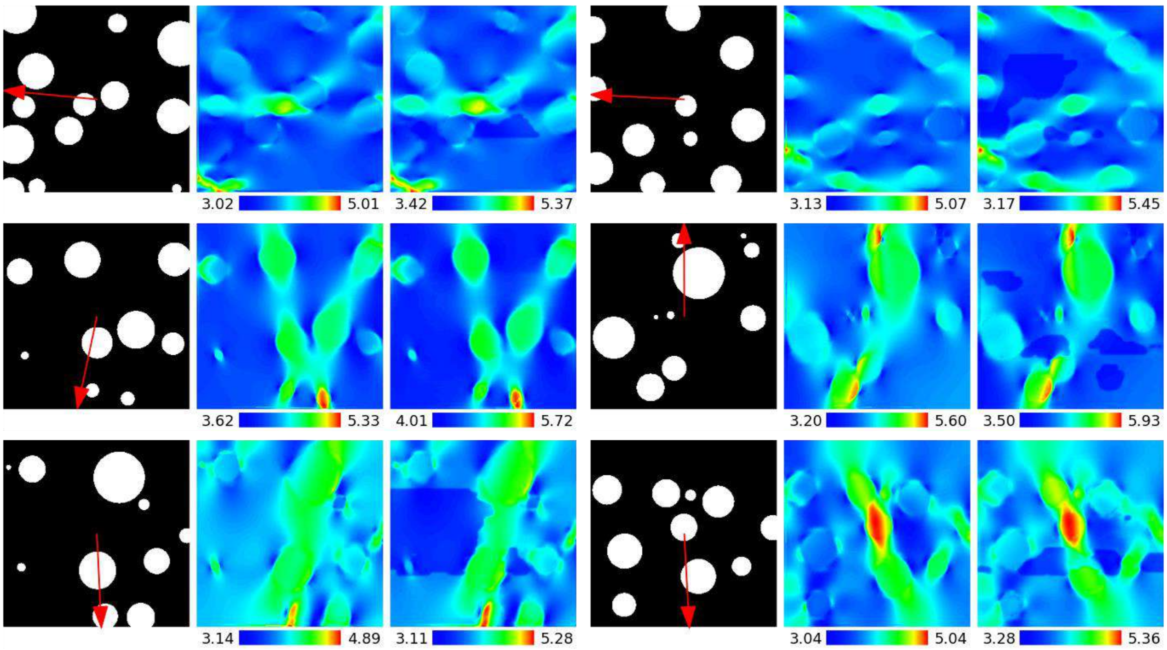


Fig. 16. **Sample geometry, MCE solution, and surrogate model output. Validation sample, *GPa***
***Source:* compiled by the authors**

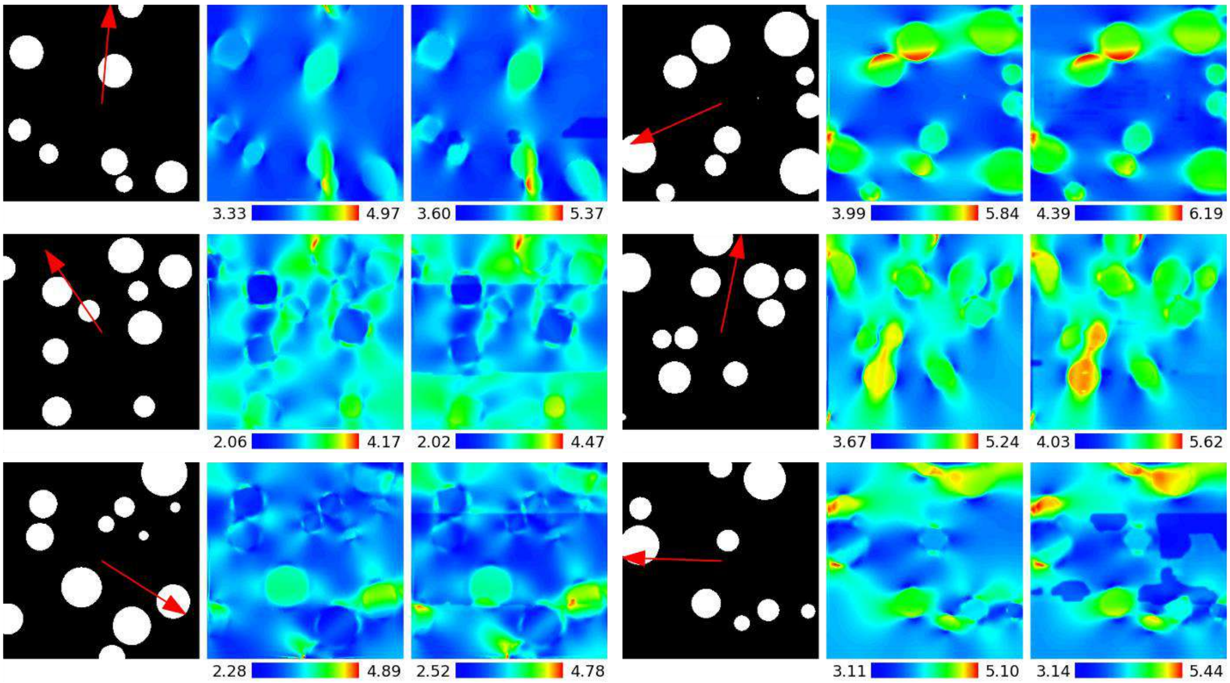


Fig. 17. **Sample geometry, MCE solution, and surrogate model output. The test sample, *GPa*.**

***Source:* compiled by the authors**

The values of the loss function and metrics for all three samples are shown in Table 3.

Table 3. **Loss function results on subsamples**

Sample	RMSE, *GPa*	Metrics $RMSE_{80\%}$, *GPa*	Metrics $RMSE_{max}$, *GPa*
Training	0.339	0.386	0.474
Validation	0.386	0.460	0.521
Test	0.412	0.478	0.540

Source: compiled by the authors

CONCLUSIONS

In this work, a surrogate model based on a convolutional neural network was constructed to approximate the finite element solution of the dispersion-strengthened composite plane sample deformation to accelerate the material microstructure stress-strain state calculations. And also the overall quality of approximations for typical problems is determined. To train the neural network, 10.000 variants of the SSS of the parameterized calculation model of a composite material sample are analyzed. A neural network based on the U–Net architecture of the encoder-decoder type is created to predict the distribution of equivalent stresses in the material according to the sample geometry and load values.

Analysis of the results showed that the mean squared error (MSE) for maximum stresses is about 540 *MPa*, and the average for the entire stress distribution is 412 *MPa*. The average maximum stresses in the samples are 5310 *MPa*. Evaluation of the results shows that: the values of the network weights after training are still not optimal in terms of minimizing the cost function; and also that the model is simple enough to accurately reproduce the solution of the finite element method.

A comparison of the calculation speed showed that the neural network is significantly ahead of the finite element method (FEM). The calculation of the FEM for 10,000 configurations on a stationary PC takes 430 *minutes*. Formation of the stiffness matrix and calculation of one loading step – 19 *seconds*. The training of the neural network on these samples takes 135 *minutes*, and the resulting surrogate model generates a matrix of equivalent stresses simultaneously for 32 samples in 9 *seconds*. Thus, the time advantage when using an already trained model is up to 70 times.

ACKNOWLEDGMENT

This work has been supported by the Ministry of Education and Science of Ukraine in the framework of the realization of the research project “Development of computational intelligence methods in the tasks of synthesizing the characteristics of critical elements, improving the reliability and efficiency of innovative technology” (State Reg. Num. 0121U100730).

REFERENCES

1. Yang, X. S., Koziel, S. & Leifsson, L. "Computational optimization, modelling and simulation: Recent trends and challenges". *Procedia Computer Science*. 2013; 18: 855–860. DOI: https://doi.org/10.1016/j.procs.2013.05.250.

2. Li, H., Shi, M., Liu, X. & Shi, Y. "Uncertainty optimization of dental implant based on finite element method, global sensitivity analysis and support vector regression". *Proceedings of the Institution of Mechanical Engineers. Part H: Journal of Engineering in Medicine*. 2019; 233(2): 232–243. DOI: https://doi.org/10.1177/0954411918819116.

3. Hu, F. & Li, D. "Modelling and simulation of milling forces using an arbitrary Lagrangian–Eulerian finite element method and support vector regression". *Journal of Optimization Theory and Applications*. 2012; 153(2): 461–484. DOI: https://doi.org/10.1007/s10957–011–9927–y.

4. Martinez–Martinez, F., Ruperez, M. & Martinez–Sober, M. "A finite element–based machine learning approach for modeling the mechanical behavior of the breast tissues under compression in real–time". *Computers in biology and medicine*. 2017; 90: 116–124. DOI: https://doi.org/10.1016/j.compbiomed.2017.09.019.

5. Zhang, W., Zhang, Z., Wu, C. & Goh, A. "Assessment of basal heave stability for braced excavations in anisotropic clay using extreme gradient boosting and random forest regression". *Underground Space*. 2020; 1: 1–12. DOI: https://doi.org/10.1016/j.undsp.2020.03.001.

6. Garcia-Crespo, A., Ruiz-Mezcua, B., Fernandez-Fdz, D. & Zaera, R. "Prediction of the response under impact of steel armours using a multilayer perceptron". *Neural Computing and Applications*. 2007; 16(2): 147–154. DOI: https://doi.org/10.1007/s00521-006-0050-1.

7. Nourbakhsh, M., Irizarry, J., Haymaker, J. "Generalizable surrogate model features to approximate stress in 3D trusses". *Engineering Applications of Artificial Intelligence*. 2018; 71: 15–27. DOI: https://doi.org/10.1016/j.engappai.2018.01.006.

8. Chan, W. L., Fu, M. W. & Lu, J. "An integrated FEM and ANN methodology for metal-formed product design". *Engineering Applications of Artificial Intelligence*. 2008; 21(8): 1170–1181. DOI: https://doi.org/10.1016/j.engappai.2008.04.001.

9. D'Addona, D. M. & Antonelli, D. "Neural network multiobjective optimization of hot forging". *Procedia CIRP*. 2018; 7: 498–503. DOI: https://doi.org/10.1016/j.procir.2017.12.251.

10. Gudur, P. P. & Dixit, U. S. "A neural network-assisted finite element analysis of cold flat rolling". *Engineering Applications of Artificial Intelligence*. 2008. 21 (1): 43–52. DOI: https://doi.org/10.1016/j.engappai.2006.10.001.

11. Abueidda, D., Almasri, M., Ammourah, R. & Ravaioli, U. "Prediction and optimization of mechanical properties of composites using convolutional neural networks". *Composite Structures*. 2019; 227: 111–264. DOI: https://doi.org/10.1016/j.compstruct.2019.111264.

12. Roberts, S., Kusiak, J., Liu, Y. L., Forcellese, A. & Withers, P. J. "Prediction of damage evolution in forged aluminium metal matrix composites using a neural network approach". *Journal of Materials Processing Technology*. 1998; 80: 507–512. DOI: https://doi.org/10.1016/S0924–0136(98)00153–8.

13. Haghighat, E., Raissi, M., Moure, A., Gomez, H. & Juanes, R. "A physics–informed deep learning framework for inversion and surrogate modeling in solid mechanics". *Computer Methods in Applied Mechanics and Engineering*. 2021; 379: 113–141. DOI: https://doi.org/10.1016/j.cma.2021.113741.

14 Shapovalova, M. I. Evaluation Assessment of the limiting state of a two-component material with spherical inclusions and predicting the reliability of the structure by methods of computer and mathematical modeling [Electronic resource]: dissertation. Doctor of Philosophy. National Technical University "Kharkiv Polytechnic Institute". Kharkiv: 2021. 156 p. http://repository.kpi.kharkov.ua/handle/KhPI-Press/54867.

15. Shapovalova, M. I. & Vodka, O. O. "Computer methods for constructing parametric statistically equivalent models of high–strength cast iron microstructure to analyze it's elastic characteristics" (in Ukraine). *In: Notes of the Tavrida National Univ. V. I.Vernadsky. Series: Technical Sciences.* 2019; 6(1): 179–187. DOI: https://doi.org/10.32838/2663–5941/2019.6–1/33.

16. Shapovalova, M. & Vodka, O. "Application of data–driven yield surface to prediction of failure probability for centrifugal pump". *Mechanics of Complex Structures. Advanced Structured Materials, Springer, Cham*. 2021; 157: 295–309. DOI: https://doi.org/10.1007/978–3–030–75890–5_17.

17. Shapovalova, M. & Vodka, O. "Image processing technology to determine the parameters of the internal structure of composite materials". *IEEE 2nd KhPI Week on Advanced Technology (KhPIWeek),*

Kharkiv, September 13 – 17. 2021. p. 539–543. DOI: https://doi.org/10.1109 KhPIWeek53812.2021.9570099.

18. Shapovalova, M. & Vodka, O. “Computer method of determining the yield surface of variable structure of heterogeneous materials based on the statistical evaluation of their elastic characteristics”. *Nonstationary Systems: Theory and Applications. WNSTA 2021. Applied Condition Monitoring, Springer. Cham.* 2022; 18: 378–392. DOI: https://doi.org/10.1007/978–3–030–82110–4_21.

19. Al-Khafaji, H. M., Habeeb, L. & Alkhafaji, A. J. “Introduction to simulation with ANSYS mechanical APDL”. *LAP Lambert Academic Publishing*. 2016. 956 p. ISBN: 978–3–659–95856–4.

20. Thompson, M. K. & Thompson, J. M. “ANSYS mechanical APDL for finite element analysis.” *Butterworth–Heinemann.* 2017. 1204 p. ISBN 978–0–12–812981–4.

21. Qinyang, Li Q., Wang, N. & Yi, D. “Numerical Analysis”. *Huazhong University of Science and Technology Press*. 2001. ISBN 7–302–04561–5.

22. Abadi, M. et. al. “TensorFlow: A system for large–scale machine learning”. *12th USENIX Symposium on Operating Systems Design and Implementation USENIX Association*. 2016. arXiv:abs/1605.08695.

23. Ronneberger, O., Fischer, P. & Brox, T. “U–net: Convolutional networks for biomedical image segmentation*”. International Conference on Medical image computing and computer–assisted intervention. Springer*. *Cham*. 2015. p. 234–241. DOI: https://doi.org/10.48550/arXiv.1505.04597.

24. Kingma D. P. & Ba J. “Adam: A method for stochastic optimization”. *The 3rd International Conference for Learning Representations.* San Diego. 2015. DOI: https://doi.org/10.48550/arXiv.1412.6980.

Conflicts of Interest:the authors declare no conflict of interest

Received 26.08.2022
Received after revision 03.10.2022
Accepted 19.10.2022

DOI: https://doi.org/10.15276/hait.05.2022.15
УДК 004.85:539.3

Застосування методів інтелектуальних обчислень для оцінки напруженого стану гетерогенного матеріалу

Бабуджан Руслан Андрійович[1)]
ORCID: https://orcid.org/0000–0001–5765–9234; ruslanbabudzhan@gmail.com
Водка Олексій Олександрович[1)]
ORCID: https://orcid.org/0000–0002–4462–9869; oleksii.vodka@gmail.com. Scopus Author ID: 56239259600
Шаповалова Марія Ігорівна[1)]
ORCID: https://orcid.org/0000–0002–4771–7485; mishapovalova@gmail.com. Scopus Author ID: 57212000432
[1)] Національний технічний університет «Харківський політехнічний інститут», вул. Кирпичова, 2. Харків, 61002,Україна

АНОТАЦІЯ

Використання сурогатних моделей дає великі переваги у роботі з системами автоматизованого проектування та 3D–моделювання, що відкриває нові можливості у проектуванні складних систем. Також вони дозволяють значно раціоналізувати використання обчислювальних потужностей в автоматизованих системах, для яких критичними є час відгуку та невисоке споживання енергії. Дана робота присвячена створенню сурогатної моделі для апроксимації скінчено–елементного рішення задачі деформування плоского зразку дисперсійно–зміцненого композиту. Запропоновано алгоритм побудови параметричної двовимірної моделі композиту. Розрахункова модель створюється за допомогою засобів автоматизованого проектування та аналізу ANSYS Mechanical, використовуючи скриптовий засіб побудови моделей APDL. Обробка параметрів напружено–деформованого стану мікроструктури матеріалу відбувається за допомогою згорткової нейронної мережі. Створена нейронна мережа на основі архітектури U–Net типу енкодер–декодер, для передбачення розподілу еквівалентних напружень у матеріалі за геометрією зразка та значеннями навантажень. Від названої архітектури береться пряма послідовність шарів. Для збільшення швидкості та стабільності навчання змінено тип частини згорткових шарів. Архітектура мережі складається із послідовно з'єднаних блоків, кожен з яких об'єднує такі шари, як згортки, побатчевої нормалізації, активації, субдискретизації, та латентний простір, що сполучує енкодер та декодер, додаючи данні про навантаження. Для об'єднання вектору навантаження створюється така архітектура нейронної мережі як конкатенатор, що додатково включає шари Dense, Reshape та Concatenate. Функція втрат моделі визначається, як середньоквадратична похибка за усіма точками вихідної матриці, що розраховує різницю між дійсним значенням цільової змінної та значенням, згенерованим сурогатною моделлю. Оптимізація функції витрат проводиться за допомогою градієнтного методу локальної оптимізації першого порядку ADAM. Дослідження процесу навчання моделі проілюстровано на графіках залежності функцій втрат та додаткових метрик. Спостерігається тенденція співпадіння показників між тренувальною та валідаційною підвибірками, що свідчить про узагальнюючу можливість моделі. Аналізуючи вихід моделі та значення метрик робиться висновок про достатню якість моделі. Проте значення ваг мережі після навчання все ще не є оптимальними у сенсі мінімізації функції витрат. А також, для точного відтворення рішення методу скінченних елементів запропонована модель є досить простою, та потребує уточнення. Проведено порівняння швидкості отримання результатів методом скінченних елементів та за допомогою запропонованої архітектури нейронної мережі. Сурогатна модель суттєво випереджує метод скінчених елементів, та використовується для прискорення розрахунків і визначення загальної якості апроксимації задач механіки такого типу.

Ключові слова: згорточна нейронна мережа; напружено–деформований стан; метод скінченних елементів; сурогатна модель; U–Net; кодер–декодер

ABOUT THE AUTHORS

Ruslan. A. Babudzhan – PhD student of the department of Dynamics and Strength of Machines. National Technical University "Kharkiv Polytechnic Institute", 2, Kyrpychova Str. Kharkiv, 62001, Ukraine.
ORCID: https://orcid.org/0000–0001–5765–9234; ruslanbabudzhan@gmail.com
Research field: machine learning, reliability and lifetime, mathematical modeling in mechanics

Бабуджан Руслан Андрійович – аспірант кафедри Динаміки та міцності машин. Національний технічний університет «Харківський політехнічний інститут», вул. Кирпичова, 2. Харків, 61002, Україна.

Oleksii O. Vodka – PhD, Associate Professor, Head of the Department of Dynamics and Strength of Machines. National Technical University "Kharkiv Polytechnic Institute", 2, Kyrpychova Str. Kharkiv, 62001, Ukraine.
ORCID: https://orcid.org/0000–0002–4462–9869; oleksii.vodka@gmail.com. Scopus Author ID: 56239259600
Research field: Vibrations; reliability and lifetime prediction of mechanical systems

Водка Олексій Олександрович – кандидат технічних наук, доцент, завідувач кафедри динаміки та міцності машин. Національний технічний університет «Харківський політехнічний інститут», вул. Кирпичова, 2. Харків, 61002, Україна.

Mariia I. Shapovalova – PhD, senior lector of the department of Dynamics and Strength of Machines. National Technical University "Kharkiv Polytechnic Institute", 2, Kyrpychova Str. Kharkiv, 62001, Ukraine.
ORCID: https://orcid.org/0000–0002–4771–7485, mishapovalova@gmail.com. Scopus Author ID: 57212000432
Research field: Information technologies; machine learning and computer vision; mathematical modeling in mechanics; reliability of complex systems and technical condition diagnostics

Шаповалова Марія Ігорівна – кандидат технічних наук, старший викладач кафедри динаміки та міцності машин. Національний технічний університет «Харківський політехнічний інститут», вул. Кирпичова, 2. Харків, 61002, Україна.