

## Research Article

# ANN Architecture Specifications for Modelling of Open-Cell Aluminum under Compression

Marek Dudzik<sup>1</sup> and Anna Małgorzata Stręk<sup>2</sup>

<sup>1</sup>Cracow University of Technology, Faculty of Electrical and Computer Engineering, Department of Traction and Traffic Control, ul. Warszawska 24, 31-155 Kraków, Poland

<sup>2</sup>Cracow University of Technology, Faculty of Civil Engineering, Division of Structural Mechanics and Material Mechanics, ul. Warszawska 24, 31-155 Kraków, Poland

Correspondence should be addressed to Marek Dudzik; [marekdudzik@pk.edu.pl](mailto:marekdudzik@pk.edu.pl) and Anna Małgorzata Stręk; [anna.strek@pk.edu.pl](mailto:anna.strek@pk.edu.pl)

Received 22 October 2019; Revised 2 January 2020; Accepted 25 January 2020; Published 28 February 2020

Academic Editor: Marek Lefik

Copyright © 2020 Marek Dudzik and Anna Małgorzata Stręk. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The knowledge on strength properties of porous metals in compression is essential in tailored application design, as well as in elaboration of general material models. In this article, the authors propose specification details of the ANN architecture for adequate modelling of the phenomenon of compressive behaviour of open-cell aluminum. In the presented research, an algorithm was used to build different structures of artificial neural networks (ANNs), which approximated stress-strain relations of an aluminum sponge subjected to compression. Next, the quality of the built approximations was appraised. The mean absolute relative error (MARE), coefficient of determination between outputs and targets ( $R^2$ ), root mean square error (RMSE), and mean square error (MSE) were assumed as criterial measures for the assessment of the fitting quality. The studied neural networks (NNs) were two-layer feedforward networks with different numbers of neurons in the hidden layer. A set of experimental stress-strain data from quasistatic uniaxial compression tests of open-cell aluminum of various apparent densities was used as data for training of neural networks. Analysis was performed in two modes: in the first one, all samples were taken for training, and in the second case, one sample was left out during training in order to play the role of external data for testing the trained network later. The taken out samples were maximum and minimum density samples (for extrapolation) and one random from within the density interval. The results showed that good approximation on the engineering level ( $MARE < 5\%$ ) was reached for teaching networks with  $\geq 7$  neurons in the hidden layer for the first studied case and with  $\geq 8$  neurons for the second. Calculations on external data proved that 8 neurons are enough to actually obtain  $MARE < 10\%$ . Moreover, it was shown that the quality of approximation can be significantly improved to  $MARE \approx 7\%$  (tested on external data) if the initial region of the stress-strain relation is modelled by an additional network.

## 1. Introduction

Modern innovations cannot succeed without materials having features that comply with even very high needs. An example of such exceptional materials are porous metals, which due to their structure manifest valuable properties. These start with the most obvious lower apparent density and thus lower weight compared to solid metals, yet keeping a reasonable level of structural strength at the same time. Another trait beneficial in various application is capacity for dispersion of multiple energy forms (impact energy, sound

waves, etc.). Typically for metals, these materials show very good electric conductivity, which combined with large specific surface resulting from the porous structure makes them perfect candidates in catalyst industry applications.

Porous metals can be catalogued, depending on the production method [1], type of porosity [2], skeleton material, certain property level, or application area [3, 4]. For the purpose of this article, it is ample to mention shortly that there are three main production routes: liquid metallurgy (direct foaming and casting), sintering metallurgy (powders and semifinished products), and lastly coating. According to

the porosity type, we distinguish: closed-cell metallic materials, open-cell metals, sponges, gasars, and others. As for the skeleton, a range of alloys or pure metals can be used, e.g., copper, aluminum, titanium, gold, steel, and others [5–9].

In the present work, the authors investigate computational description of mechanical behaviour of an open-cell metal sponge with an aluminum skeleton, self-produced by investment casting. The article describes one of the stages of broader research, which includes among others study on manufacturing parameter calibration [10], experimental assessment of mechanical behaviour [11, 12] and modelling [13]. A preliminary positive verification of the possibility of usage of artificial neural networks in service for modelling the open-cell aluminum compressive behaviour was reported in [14]. The present study is a consequent development of the previous work, consisting in algorithmic creation and comparison and selection of a specific neural network structure to closely represent the material's stress-strain characteristics in compression.

Artificial neural networks constitute an interesting direction in numerical investigations in mechanics of materials and structures and in engineering [15–20]. A properly structured network trained on experimental data can become a remarkable asset in either building or verifying mechanical models or surrogate models. However, the following questions arise: How to appropriately choose the network architecture? What criterion to assume as the structure's quality estimator? and, on the contrary, how to pre- and postprocess data in relation to the chosen ANN? The discussion on neural network structural aspects, such as the optimization of weights, the number of inputs, the choice of learning parameters, the number of layers, the number of neurons, activation functions, and inclusion of the statistical approach, has been vigorously ongoing for recent decades, e.g., [21–25]. Many researchers agree, however, that either there exists no universal and explicit method for tuning these parameters or there is rather few guidance in this matter [23, 26, 27]. With the present study, the authors are aimed at addressing this lack for the field of mechanics of aluminum sponges. For this purpose, a number of ANNs were algorithmically built, trained on real experimental data from uniaxial tests, and evaluated. The evaluation led to formulation of guidelines regarding structural specifications of neural networks. Below in the remaining paragraphs of Introduction, there will be adduced general assumptions, reasoning outline, and contributions of the discussed research.

Our previous, preliminary study [14] had confirmed that the general network structure, which had been a two-layered feedforward network, was successful with regard to the considered phenomenon. Used activation function types were proven adequate: tansig in the first hidden layer and purelin in the second output layer [1]. It was then decided to sustain these general assumptions regarding the network architecture. In addition, they were in agreement with the advised general solution for fitting problems, which is “the multilayer perceptron, with tansig neurons in the hidden layers and linear neurons in the output layer” [28].

The investigated neural networks were trained on the data from previously conducted uniaxial tests [11, 12] and included stress, strain, and additionally apparent density of samples. As for data preparation for neural networks, a common way of conduct is to perform one of the following: normalisation, scaling, statistical standardisation, or other mathematical approach [29] in order to transform data into the interval  $\langle -1; 1 \rangle$  or  $\langle 0; 1 \rangle$ . However, the fact of performing the preprocessing, as well as its type choice, has impact on the whole process of neural computations. It is necessary to be aware that this impact might be positive, neutral, but even adverse. One of the undeniable benefits is, colloquially speaking, “equalling” of data significance among them, or, in other words, avoidance of favouring any data. On the contrary, data processing could also cause loss of some initial information, which definitely should be counteracted [30]; also, it might lengthen computations [31]. Interestingly, it was proven that, with data sets big enough, networks are able to compute nearly as good results without standardisation as with it [31]. In the present research, it was decided that normalisation would be performed, since the original data ranges were rather different (strain from 0% up to ca. 90%, stress from 0 MPa up to ca. 12.5 MPa, and apparent densities from 0.32 g/cm<sup>3</sup> up to 0.56 g/cm<sup>3</sup>).

Regarding the description of compressive behaviour, even before preliminary neural network computations, a general choice of variables and relation type between them had to be defined. One of the provisions had been the theoretical formula in the form of the dimensional law given as [32]

$$\frac{\text{Cellular material property}}{\text{Skeleton property}} = C \left( \frac{\rho}{\rho_s} \right)^n. \quad (1)$$

The above formula links a sought cellular material's property with the respective skeleton's property and both the cellular material's density  $\rho$  and the skeleton's density  $\rho_s$ . The two linking scaling parameters  $C$  and  $n$  should be determined experimentally for the given material. Using relation (1) allows one to calculate a few selected magnitudes [32], such as the plastic collapse stress ( $\sigma_{pl,c}$ ) and limit stress between densification and plastic collapse ( $\sigma^*$ ). Additionally, numerous experiments show that there exists a sort of correlation between stress-strain characteristics of cellular metals in compression and their apparent or relative density, e.g., [11, 32, 33]. On this basis, the authors decided to choose the following general relationship as the function to be approximated by the neural calculations:

$$\sigma = f(\varepsilon, \rho), \quad (2)$$

where  $\sigma$  is the stress,  $\varepsilon$  is the strain, and  $\rho$  is the apparent density.

The approach from the study [32], however, has some significant limitations. Only few characteristic stress values can be determined with its use. Also, it is necessary to calibrate the experimental constants  $C$  and  $n$ . Additionally, the knowledge on the skeleton's material parameters including its yield point, modulus, and density is necessary for calculations according to the cited approach. Taking this into

consideration, we decided to seek for an alternative method to determine compressive behaviour of open-cell aluminum. We decided to employ neural networks for this purpose. This path allows one to transfer the experimental constants' calibration to machine learning. It also frees the user from testing not only sponge samples but also the skeleton material, as it does not require the knowledge on skeleton properties. Finally, with such an approach, it is possible to find stresses for the whole range of strains and densities instead of only single values as in the referred above method.

The essence of the study laid in the subsequent stage which was algorithmic creation of 250 different NNs, training them on experimental data to fit the aforementioned relation (2) and assessment of the quality of approximations. For the evaluation, four criterial measures were selected [34]: mean absolute relative error (MARE), coefficient of determination between outputs and targets ( $R^2$ ), root mean square error (RMSE), and mean square error (MSE). For the MARE, there were two assumed threshold values: 10% and 5%; additionally, the minimum was sought. In case of the coefficient of determination, the threshold was assumed as 0.95. The root mean square error had the same unit as targets (MPa) and was used as a measure to complete the assessment done by the unitless MARE. The mean square error was set as the networks' internal performance function with the goal at 0.

The outcomes of the presented research met the prior set objective: determination of specifications of the neural network architecture, which could be used as practical guidance in modelling of the phenomenon of compression of open-cell aluminum. Significant contributions of the authors' work are as follows:

- (i) A proposition of an algorithm for the analysis of a range of neural networks and evaluation of the quality of obtained results (Section 2.2.3)
- (ii) Investigation of how exclusion of a random sample affects general effectiveness of the proposed algorithm (Section 3.1)
- (iii) Investigation of the extrapolation capabilities of obtained network architectures (Section 3.2)
- (iv) Verification of the optimal number of neurons in the hidden layer required for the description of the considered phenomenon (compression of open-cell aluminum) with the engineering accuracy, i.e., mean absolute relative error  $\leq 10\%$  and  $\leq 5\%$  (Sections 3.2 and 3.3)
- (v) Verification of models of considerable complexity networks with  $\geq 47$  neurons in the hidden layer (Sections 3.1–3.3)
- (vi) A proposition of improvement of the quality of approximation by introducing an additional network for the preplateau stress-strain relation and comparison of results obtained for testing the taught networks in two data ranges (Section 3.3)
- (vii) Finally, a proposition of neural network structure specifications for the use in modelling and/or

metamodelling (e.g., virtual experiments) of compression of aluminum sponges (Sections 3.1–3.3 and 4)

## 2. Materials and Methods

**2.1. Materials and Data from Uniaxial Compression Experiments.** The subject material was an open-cell metal sponge with an aluminum skeleton self-produced by investment casting. The authors limit the present description only to details necessary for clarity and lucidity of communication; more extensive information on production and experimental characteristics of the chosen material is given in separate publications [10–12].

Parameters of the manufacturing process were tuned in the course of the production, so two sample groups were obtained: a prototype lot denoted as “P” (exhibiting minor structural imperfections and larger apparent densities) and a regular lot denoted as “R” (with smaller apparent densities and without visible structural mistakes). The structural imperfections of the “P” samples included occasional semiclosed cells, swellings of struts, and form residuals sank inside. Exemplary samples of both types are shown in Figure 1. Samples' dimensions were fit to satisfy the condition of the minimal number of cells for the specimen's volume to be representative [35]. The dimensions were on average  $(53.0 \times 39.5 \times 39.0)$  mm and  $(62.8 \times 39.5 \times 38.0)$  mm, respectively, for the “P” and “R” lots. The pores per inch index was  $\text{PPI} = 5.4/6.2$  in both groups. Each sample had different apparent densities (the values are listed in Figure 2); apparent densities' variation was attributed mainly to the impact of structural imperfections within the “P” group but also to the stochastic distribution of cell dimensions in both groups (compare [36]).

Uniaxial compression experiments were performed using a Zwick 1455 20 kN machine and computer programme testXpert II. Assumed testing conditions were as follows: initial force 5 N, data acquisition frequency 100 Hz, and strain speed  $0.5\% h_0$  in mm/s, where  $h_0$  was the initial sample height. Figure 2 shows obtained stress-strain curves. Samples from the group “P” are drawn with solid lines, while samples from the other group are drawn with dashed lines. Along with the plot lines, values of apparent density  $\rho$  in  $\text{g/cm}^3$  for each sample are given. It can be clearly seen that the compressive behaviour is related to apparent density.

The results from the cited compression tests as well as specimens' characteristics (apparent density and indicator of the lot) were used as data for training of neural networks.

**2.2. Method: The ANN Algorithm.** Computations with neural networks consisted in building a number of neural networks which varied in structure. Each such network will be referred in this article as a unit neural network. The efficiency of unit networks was assessed by comparing the mean absolute relative error (MARE), coefficient of determination between outputs and targets ( $R^2$ ), root mean square error (RMSE), and mean square error (MSE) for each of them. All computations were implemented as an algorithm. The algorithm

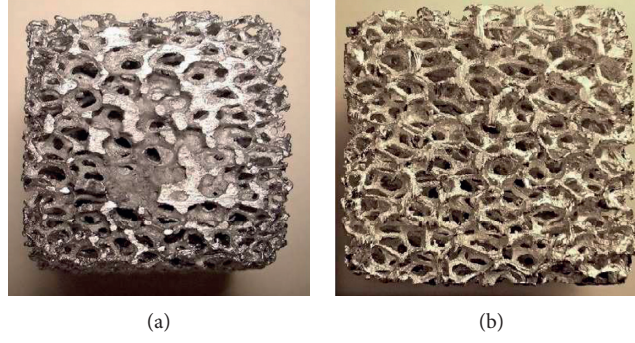


FIGURE 1: Samples of the open-cell aluminum: (a) a sample from the prototype group; (b) a sample from the regular group.

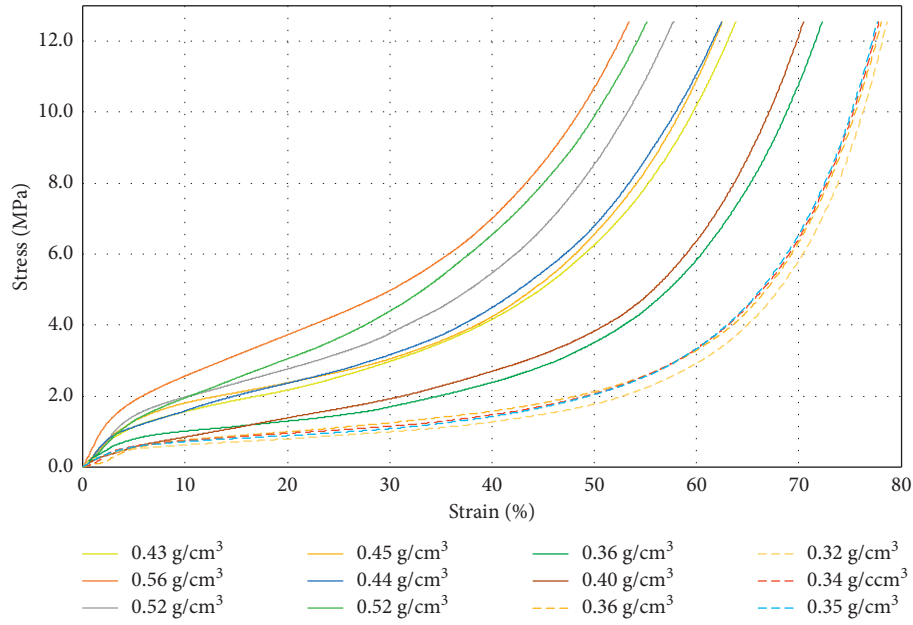


FIGURE 2: Stress-strain plots from the compression test; the numbers by plots are apparent densities  $\rho$  in g/cm<sup>3</sup>. Solid lines represent the prototype group, and dashed lines are for regular samples.

was performed in two modes: the first mode for all data and the second mode with one specimen excluded so that it can be used later as testing data. Calculations were performed using Matlab R2019A version.

In order to make the method's description more comprehensible, it will be divided into four subsections. First, preparation and processing of data will be discussed (Section 2.2.1); then, the unit network is built (Section 2.2.2); next, the algorithm is built, which compares the unit NNs (Section 2.2.3); and finally, the quality assessment is done (Section 2.2.4).

**2.2.1. Data Preparation and Pre- and Postprocessing.** Before entering the network, data obtained in uniaxial compression tests and specimens' characteristics were prepared as follows:

- (1) 12 open-cell aluminum samples were taken into consideration; for each sample, 1,000 experimental strain and respective stress values were taken. The

chosen strain values were equally dispersed among the original strain data range for each sample. Additionally, the sample's apparent density and two parameters:  $l_P$  and  $l_R$ , defining to which of the two lots ("P" or "R") the sample belonged were included. The parameters were equal to 1 if the sample belonged to the given lot or 0 in the opposite case. Thus, there were  $n = 12000$  produced data sets:  $\{\epsilon_i, \sigma_i, \rho_i, l_P, l_R\}$ , with  $i = 1, 2, \dots, n$ .

- (2) In coordination with the assumed sought relation form  $\sigma = f(\epsilon, \rho)$  (equation (2)), arguments entering the neural network were set into  $n$  vectors:  $A_i = [\epsilon_i, \rho_i, l_P, l_R]^T$ , with  $i = 1, 2, \dots, n$ . The respective value of experimental stress from the  $i$ -th data set was assumed as the corresponding target for the network:  $t_i = \sigma_i$ , with  $i = 1, 2, \dots, n$ .
- (3) Two modes of calculations were performed. In the first mode, all 12 samples were used in the process of network teaching,  $n = 12000$ . In the second mode, one aluminum sample was taken out of the input



data for teaching the networks ( $n = 11000$ ); next, this sample was used to test the quality of networks taught in the previous step.

Before the training, data were normalised. Since network implementation was done in the Matlab environment, the inbuilt function `mapminmax` was used [37–39]. This function is a linear transformation into the interval of given boundaries:

$$V' = \frac{V - V_{\min}}{V_{\max} - V_{\min}} \cdot (V'_{\max} - V'_{\min}) + V'_{\min}, \quad (3)$$

where  $V$  is the original value;  $V'$  is the transformed value;  $V_{\max}$  and  $V_{\min}$  are original interval boundaries; and  $V'_{\max}$  and  $V'_{\min}$  are desired range boundaries, whose values are  $-1$  and  $1$  here. Vectors  $A_i$  were normalised, respectively, into the following input vectors:  $X_i = [x_{1,i} = \varepsilon_{\text{norm},i}, x_{2,i} = \rho_{\text{norm},i}, x_{3,i} = l_{\text{norm},P}, x_{4,i} = l_{\text{norm},R}]^T$ , with  $i = 1, 2, \dots, n$ . On the other end, the output of the network which was supposed to correspond to the sought stress was also normalised:  $y_i = \sigma_{\text{appr},\text{norm},i}$ , so the reverse action was needed in order to obtain the desired stress approximation:  $y_i = \sigma_{\text{appr},\text{norm},i} \xrightarrow{\text{post-processing}} y_i = \sigma_{\text{appr},i}$ .

**2.2.2. Learning Parameters and the Unit Neural Network.** The unit NN's learning consisted of three stages: training, validation, and testing. Data from experiments were divided between these steps in the following proportions: 60% for training, 20% for validation, and 20% for testing. It was specified that the same sets of input vectors were assigned to respective steps for all unit networks. The selection was performed in the manner which is now going to be shortly described. Data from uniaxial experiments were a uniformly dense data point sequence with 12000 elements. Every fourth element was assigned to validation, every fifth to testing, and the rest to training. Such grouping enabled equivalence of data fractions used in all learning steps. Also, since data choice was the same for all unit networks, it did not affect learning, and thus, the quality of the networks themselves could be compared.

The assumed training algorithm was the Levenberg–Marquardt algorithm [40] and the mean square error (MSE) was chosen as the performance function. Other assumed training parameters are set in Table 1. Two parameters were changed in comparison with an earlier, preliminary work [14]; namely, the maximum validation failures were risen from 6 to 12 and the maximum number of epochs was from 1000 to 100000. The alternation aimed at reassuring better accuracy and enabling better training.

The chosen general network structure type and activation function types were assumed in accordance with what is advised in case of nonlinear function approximation in the literature [28] and with previously investigated examples [14]. The unit neural network was assumed as a feedforward network with two layers: one hidden layer denoted as  $\{1\}$  and one output layer denoted as  $\{2\}$ . A detailed scheme of the unit NN is shown in Figure 3 and will be explained now. In mathematical expressions (4)–(11), the index  $i = 1, 2, \dots, n$ ,

TABLE 1: Learning parameters for a unit NN.

Learning parameter	Value
Performance function goal	0
Minimum performance gradient	$10^{-10}$
Maximum validation failures	12
Learning rate	0.01
Momentum	0.9
Maximum number of epochs to train	100000

indicating the numbering of a given data set, is going to be omitted, since it is obvious that one cycle of processing by the unit network concerned one and the same set of data and that eventually all data (all  $i$ -s) were exhausted and all once, however, not in the sequence of numbering but randomly.

The vector  $\mathbf{X}$  entered the hidden layer  $\{1\}$  as the input. The number of neurons in the hidden layer was varying and was from the range  $s^{(1)} = (1/50)^{(1)}$ . The activation (transfer) function was tansig-hyperbolic tangent sigmoid, mathematically equivalent to  $\tanh$  [37]:

$$f^{(1)}_{\text{activ}}(\arg^{(1)}) = \frac{e^{2 \cdot \arg^{(1)}} - 1}{e^{2 \cdot \arg^{(1)}} + 1} = \tanh(\arg^{(1)}), \quad (4)$$

where  $\arg^{(1)}$  is the argument of the transfer function given by

$$\arg^{(1)} = \mathbf{W}^{(1)} \cdot \mathbf{X} + \mathbf{B}^{(1)}, \quad (5)$$

with  $\mathbf{X}$  being the input column vector,  $\mathbf{B}^{(1)}$  being the column vector of biases for the hidden layer, and  $\mathbf{W}^{(1)}$  being the matrix of weights of input arguments for the hidden layer:

$$\mathbf{X} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad \mathbf{B}^{(1)} = \begin{bmatrix} b_{1^{(1)}} \\ b_{2^{(1)}} \\ \vdots \\ b_{p-1^{(1)}} \\ b_{p^{(1)}} \\ b_{p+1^{(1)}} \\ \vdots \\ b_{s^{(1)}} \end{bmatrix}, \quad (6)$$

$$\mathbf{W}^{(1)} = \begin{bmatrix} w_{1,1}^{(1)} & w_{1,2}^{(1)} & w_{1,3}^{(1)} & w_{1,4}^{(1)} \\ w_{2,1}^{(1)} & w_{2,2}^{(1)} & w_{2,3}^{(1)} & w_{2,4}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ w_{p-1,1}^{(1)} & w_{p-1,2}^{(1)} & w_{p-1,3}^{(1)} & w_{p-1,4}^{(1)} \\ w_{p,1}^{(1)} & w_{p,2}^{(1)} & w_{p,3}^{(1)} & w_{p,4}^{(1)} \\ w_{p+1,1}^{(1)} & w_{p+1,2}^{(1)} & w_{p+1,3}^{(1)} & w_{p+1,4}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ w_{s^{(1)},1}^{(1)} & w_{s^{(1)},2}^{(1)} & w_{s^{(1)},3}^{(1)} & w_{s^{(1)},4}^{(1)} \end{bmatrix}.$$

As a result of computations in the hidden layer  $\{1\}$ , the column vector of the hidden layer  $\{1\}$  outputs  $\mathbf{Y}^{(1)}$  was produced:

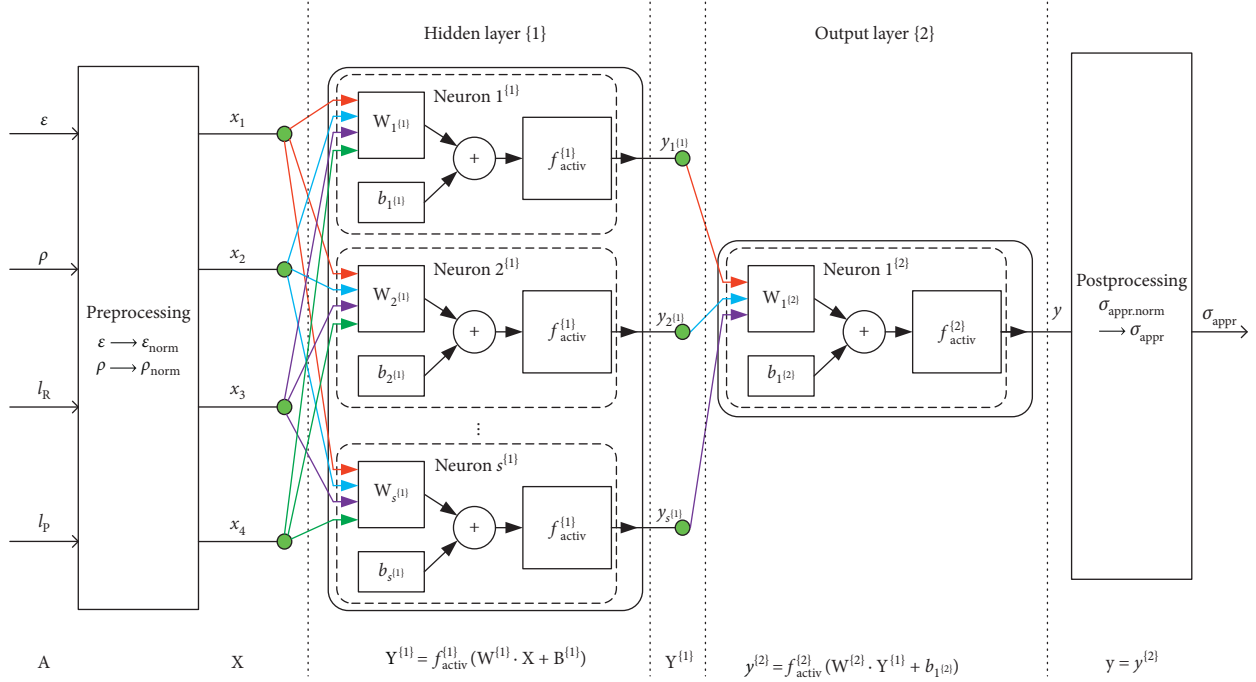


FIGURE 3: Structure of a unit neural network used in this study. All symbols are explained in the text (Section 2.2.2).

$$\mathbf{Y}^{[1]} = \begin{bmatrix} y_{1^{[1]}} \\ y_{2^{[1]}} \\ \vdots \\ y_{p-1^{[1]}} \\ y_{p^{[1]}} \\ y_{p+1^{[1]}} \\ \vdots \\ y_{s^{[1]}} \end{bmatrix}. \quad (7)$$

Hidden layer {1} outputs entered the second output layer {2}. In this layer, the number of neurons was constant and equal to  $s^{[2]} = 1^{[2]}$  in accordance with a single variable output [29] and the transfer function was purelin [37]:

$$\mathbf{f}^{[2]}_{\text{activ}} = a \cdot \arg^{[2]}, \quad (8)$$

where  $a = 1$  is the directional coefficient and  $\arg^{[2]}$  is the argument of the transfer function given by

$$\arg^{[2]} = \mathbf{W}^{[2]} \cdot \mathbf{Y}^{[1]} + b_{1^{[2]}}, \quad (9)$$

with  $\mathbf{Y}^{[1]}$  as in (7),  $b_{1^{[2]}}$  being the bias for the output layer, and  $\mathbf{W}^{[2]}$  being the row vector of weights of input arguments for the output layer:

$$\mathbf{W}^{[2]} = [w_{1,1}^{[2]} \ w_{1,2}^{[2]} \ \dots \ w_{1,j-1}^{[2]} \ w_{1,j}^{[2]} \ w_{1,j+1}^{[2]} \ \dots \ w_{1,s^{[1]}}^{[2]}]. \quad (10)$$

Finally, after computations in the output layer, the output of the unit network,  $y$ , was returned [41]:

$$y = f_{\text{activ}}^{[2]}(\mathbf{W}^{[2]} \cdot (f_{\text{activ}}^{[1]}(\mathbf{W}^{[1]} \cdot \mathbf{X} + \mathbf{B}^{[1]})) + b_{1^{[2]}}). \quad (11)$$

**2.2.3. The Algorithm for Unit Neural Network Building.** The general purpose for implementation of the algorithm was to execute a number of networks which differed in structure and

determine criterial measures—MSE, MARE, RMSE, and  $R^2$ —for each of them (the three measures are explained in the next section). The algorithm was performed in two modes: the first mode for all data and the second mode with 11 samples as the input. Now the structure of the algorithm is going to be described in detail, and an auxiliary scheme is shown in Figure 4.

The algorithm consisted of two procedures: the parent procedure P1 and the procedure P2, which was nested in P1. The P1 procedure's overall purpose was to provide varying unit network architecture parameters by attributing a given number of neurons from the range  $s^{[1]} = (1/50)^{[1]}$  to hidden layers of unit NNs. The procedure started with learning data index assignment for training, validation, and testing. Next, two loops for the above-mentioned objectives were repeated until  $s^{[1]} = 50^{[1]}$ . In the first loop, experimental data were loaded and the P2 procedure was executed. After finishing both loops, the algorithm stopped.

In short, the procedure P2 aimed at building unit networks with structures defined before by the P1 procedure. The procedure P2 started with loading the unit network characteristics attributed in P1 ( $s^{[1]}; s^{[2]} = 1 = \text{const}$ ), loading original data, and learning settings. Then, computations took place: data assignment to learning steps according to earlier defined indexation and network learning together with pre- and postprocessing of data. Results were saved. After five reiterations, the procedure P2 ended and exited back to P1.

Several (five) repetitions of calculations with the same number of neurons were implemented in the P2 procedure intentionally, as the initial attribution of weights in training is random [37]. In consequence, a single execution of the process does not guarantee obtaining the best result from the network with the given number of neurons, whereas several reiterations significantly increase the probability of obtaining the best result for the chosen network architecture.

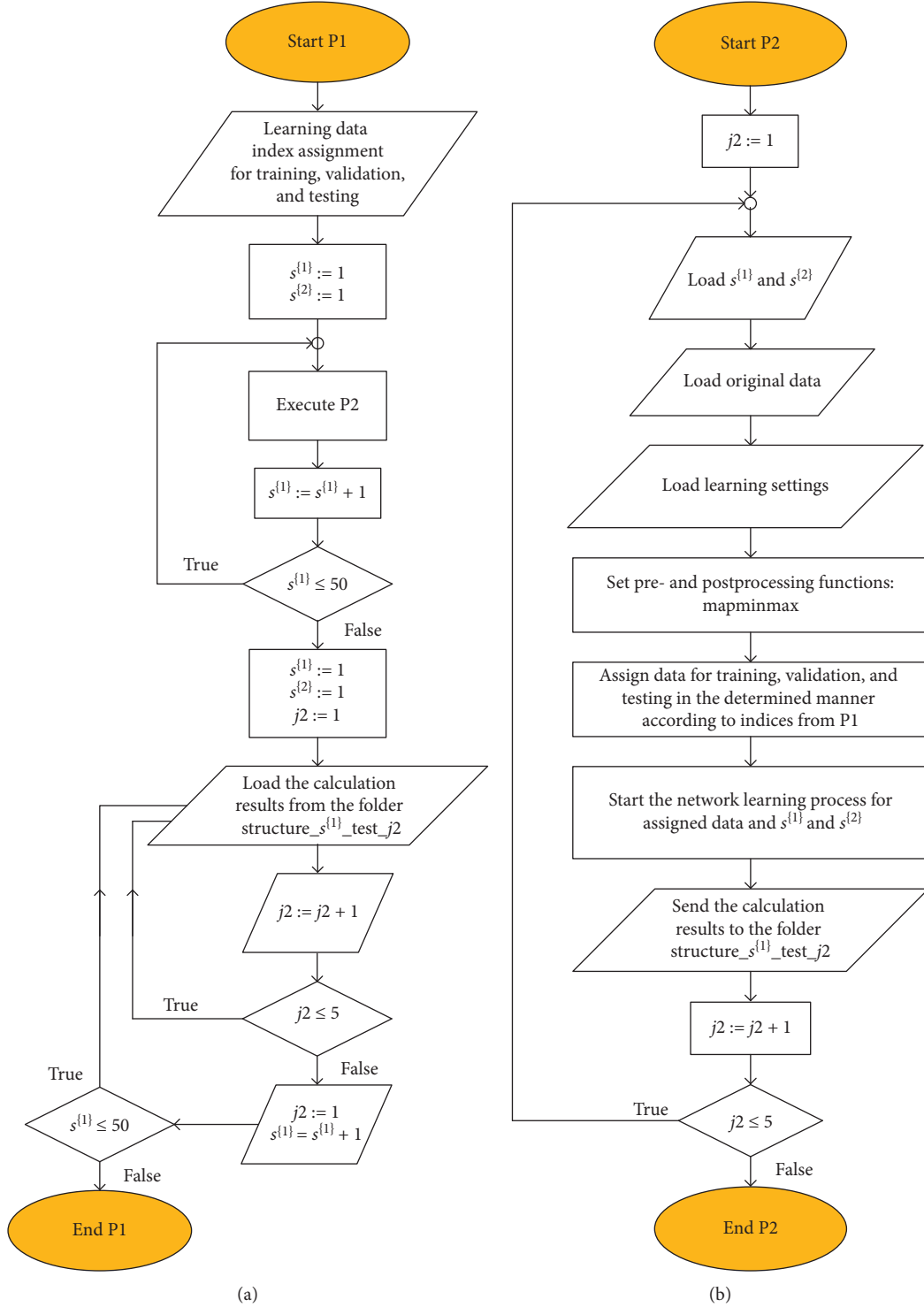


FIGURE 4: Algorithm for choosing the best unit network: (a) parent procedure P1; (b) nested procedure P2.

**2.2.4. Evaluation Criteria.** As has been said in Section 2.2.2, the mean square error (MSE) was chosen as the internal network quality measure, that is, the performance function. Assessment based on this measure was performed in the validation stage. The criterion for MSE was minimal, which in practice means that its target value was desired as 0. The mean square error was defined in computations as follows:

$$\text{MSE} = \frac{\sum_{i=1}^n (t_i - \sigma_{\text{appr},i})^2}{n} = \frac{\sum_{i=1}^n (e_i)^2}{n}, \quad (12a)$$

where  $n = 12000$  or  $n = 11000$  is the number of experimental data sets:  $\{\varepsilon_i, \sigma_i, \rho_i\}$ ,  $i = 1, 2, \dots, n$ ;  $t_i = \sigma_i$  is a target for the network (experimental stress),  $i = 1, 2, \dots, n$ ; and  $\sigma_{\text{appr},i}$  is the final output of the network respective to the  $i$ -th target

$t_i = \sigma_i$  (approximated stress),  $i = 1, 2, \dots, n$ ; implicitly, the error was also defined as

$$e_i = t_i - \sigma_{\text{appr},i}. \quad (12b)$$

It should be noted that data normalisation and the reverse procedure were performed before calculation of errors and MSEs; hence, the units of errors and MSEs were the unit of stresses and the squared unit of stresses, respectively (MPa and MPa<sup>2</sup> in this case).

Externally, the measure of the mean absolute relative error (MARE) was used as one of the quality indices. This measure was the quality indicator for values calculated in the test stage. Two threshold values were determined as desired engineering accuracy levels: MARE = 5% and MARE = 10%. Apart from these, the minimum was sought. The understanding of the mean absolute relative error in our calculations is explained by the following formula:

$$\text{MARE} = \frac{\sum_{i=1}^n |t_i - \sigma_{\text{appr},i}|/t_i}{n} = \frac{\sum_{i=1}^n |\text{RE}_i|}{n} = \frac{\sum_{i=1}^n \text{ARE}_i}{n}, \quad (13a)$$

with  $n$ ,  $t_i$ ,  $\sigma_{\text{appr},i}$ , and  $i$  as mentioned above and with additional definitions of the relative error and absolute relative error as

$$\text{RE}_i = \frac{t_i - \sigma_{\text{appr},i}}{t_i}, \quad (13b)$$

$$\text{ARE}_i = |\text{RE}_i| = \left| \frac{t_i - \sigma_{\text{appr},i}}{t_i} \right|. \quad (13c)$$

Another measure for the evaluation in the testing stage was the root mean square error (RMSE):

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n (t_i - \sigma_{\text{appr},i})^2}{n}} = \sqrt{\frac{\sum_{i=1}^n (e_i)^2}{n}}, \quad (14)$$

with variables, indices, and errors assumed as in (12a) and (12b).

The last quality criterion referred to the results of linear regression between the network's outputs and targets ( $\sigma_{\text{appr},i} \xleftrightarrow{l.\text{ regr.}} t_i = \sigma_i$ ) for all three stages of training, validation, and testing ( $t_i$ ,  $\sigma_i$ ,  $\sigma_{\text{appr},i}$ , and  $i$  are as above). Initially, the Pearson correlation coefficient  $R$  was calculated. Then, its square—the coefficient of determination ( $R^2$ )—was computed. The condition assumed for the best approximation was  $R^2 \rightarrow \max$ , which is equivalent to  $R^2 \rightarrow 1$ , and the condition to regard the approximation as successful was  $R^2 = 0.95$ . The coefficient of determination was calculated for the testing stage.

### 3. Results and Discussion

The outcome of the study regarding modelling the relationship between stress, strain, and apparent density of cellular aluminum in compression showed that it is possible to determine the general instruction for structural aspects of

neural networks such that the approximation attains a specified degree of accuracy.

Calculations with the algorithm were performed in two modes: firstly on data from all 12 specimens and secondly on data from 11 specimens with one sample excluded. The purpose of the first mode was to assess how the quality of approximation changed with respect to the number of neurons in the hidden layer and to roughly determine the required hidden layer size for the assumed accuracy. This is described in Section 3.1. The exclusion mode was performed in order to teach networks similarly as in the first mode, but saving the eliminated sample data for later testing of the networks' quality as on external data. One randomly taken out specimen, no. 8, was excluded with the purpose of checking the approximation within the density interval. The lowest density sample, no. 1, and the greatest density sample, no. 12, were omitted with the purpose of checking the extrapolation capability.

Section 3.2 presents results of interpolate and extrapolate testing of chosen networks taught on the 11-sample input. The results showed that only networks with 7 or 8 neurons produced approximation at the acceptable level.

Since some of the results showed that the initial region of the stress-strain curves was somewhat problematic for the ANNs to approximate, the algorithm was performed once more only for up to 200 initial experimental data for each aluminum sponge. This was done again with exclusion of sample no. 8. Teaching and testing of these networks are described in Section 3.3, and the results show that the used solution allowed to improve the results of the actual MARE in testing to about 7%.

For the purpose of the discussion of results, four additional measures were introduced:

- (i)  $\text{MSE}_{[5]}$ : the mean value of mean square errors from five individual tests for networks with the same number of neurons in the hidden layer
- (ii)  $\text{MARE}_{[5]}$ : the mean value of mean absolute relative errors from five individual tests for networks with the same number of neurons in the hidden layer
- (iii)  $\text{RMSE}_{[5]}$ : the mean value of root mean square errors from five individual tests for networks with the same number of neurons in the hidden layer
- (iv)  $R^2_{[5]}$ : the mean value of determination coefficients from five individual tests for networks with the same number of neurons in the hidden layer

All essential results and figures are presented below in detail. However, for the clarity of the discourse within the main article body, some of the figures are set as Supplementary Materials. Whenever illustrations are transferred to Appendix, the information about it is provided in the text.

**3.1. Results for Networks with 12-Sample and 11-Sample Inputs.** Figure 5 shows MSE and  $\text{MSE}_{[5]}$  for validation stages for all taught networks built for sets of 12 samples. It can be seen that, in both cases, MSE of up to 12 neurons for



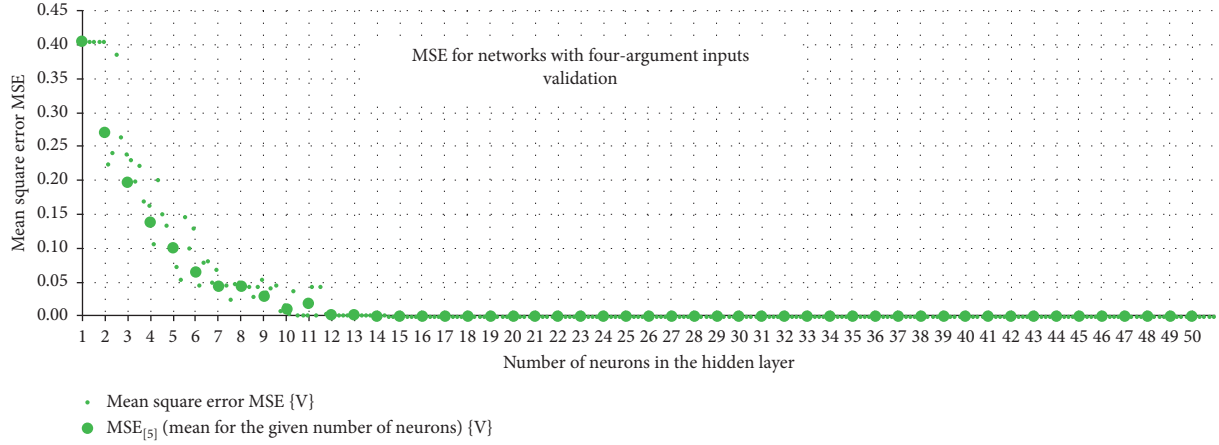


FIGURE 5: Values of MSE and  $MSE_{[5]}$  in function of the number of neurons in the hidden layer for networks taught on full experimental data.

individual networks was slightly scattered around  $MSE_{[5]}$ . On the contrary, from 12 neurons on, the mean square error was almost equal to 0 in all instances. Similar characteristics were obtained for networks taught on 11-sample inputs. The boundary number of neurons was 12 for networks taught with excluded sample nos. 1 and 12 and 13 for networks with sample no. 8 eliminated. Analogous graphical representations of MSE and  $MSE_{[5]}$  for all networks trained on 11-sample inputs are depicted in Figures A.1–A.3 in Supplementary Materials.

Figure 6 depicts MARE,  $MARE_{[5]}$ , RMSE, and  $RMSE_{[5]}$  for networks with the 12-sample input. Characteristic obtained values are summarised in Table 2. As for the networks with the 11-sample input, characteristic obtained values are set in Tables 3–5, while graphs showing MARE,  $MARE_{[5]}$ , RMSE, and  $RMSE_{[5]}$  are in Figures A.4–A.6 in Supplementary Materials.

For the complete set of samples, the threshold values 10% and 5% of the mean absolute relative error were first obtained for individual networks with 4 and 7 neurons, respectively. This fact implies that, with a larger number of repetitions of tests for such architectures, one could have a chance of achieving a learned network providing satisfactory results. However, the presented research showed that it is more reliable to choose networks with at least 6 and 10 neurons, since for such structures mean values of the critical measure reached the boundary  $MARE_{[5]} \leq 10\%$  and  $MARE_{[5]} \leq 5\%$ , respectively. If one would require to be almost sure that the set levels 10% and 5% would not be exceeded, they should select a hidden layer with at least 7 and 11 neurons, respectively. The lowest value of the mean absolute relative error was achieved in the case with 47 neurons and was equal to  $MARE_{\min} = 0.239\%$ . Root mean square errors and coefficients of determination for all above instances are in Table 2. As for the set of samples with a specimen excluded, analogous reasoning can be performed but with respective numbers of neurons, as shown in Tables 3–5. It can be seen that the respective sizes of hidden layers for the given assumed accuracies are the same or different at

most by only 1 neuron compared to the 12-sample input mode. This observation leads to a general conclusion that, with 5 neurons, one can reach  $MARE < 10\%$ , and with 8,  $MARE < 5\%$ . Also, the aforementioned fact proves that the assumed data set was large enough, since exclusion of about 8% of inputs (with targets) did not affect the results significantly.

In case of the coefficient of determination both for its mean and for consideration of individual instances, it turned out that 2 neurons guarantee the output-to-target relation at the assumed level of accuracy in both modes.

Now a detailed insight into network teaching quality will be presented. It will be shown on the example of the network taught on the 12-sample input for which  $MARE \leq 5\%$  individually for the first time (7 neurons). Results for the network with the first obtained accuracy below 10% (4 neurons) and for the best accuracy network (47 neurons) are presented in Supplementary Materials. Such a choice was made since the threshold  $MARE = 10\%$  turned out to produce underfitting, while the high number of neurons leads to overfitting. Also, because of the fact that respective results obtained for 11-sample input networks had very similar general characteristics, the detailed information on them is presented in Supplementary Materials.

Figure 7 shows network outputs, i.e., approximate stresses  $\sigma_{\text{appr},i}$ , plotted against targets, i.e., experimental stresses  $t_i = \sigma_i$ , for all learning stages separately (training, validation, and testing) as well as combined. A linear regression between the two magnitudes was performed. The obtained linear fits are depicted left to each of the graphs, and the Pearson coefficients are above them. The ideal correlation would be  $\sigma_{\text{appr},i} = 1 \cdot \sigma_i + 0$  and  $R = 1$ . Based on Pearson's coefficient, the ideal coefficient of determination would be  $R^2 = 1^2 = 1$ . Analogous figures for remaining networks are in Supplementary Materials (Figures A.7–A.17).

As for the errors, histograms are depicted in Figure 8. Errors are understood here in accordance with the definition

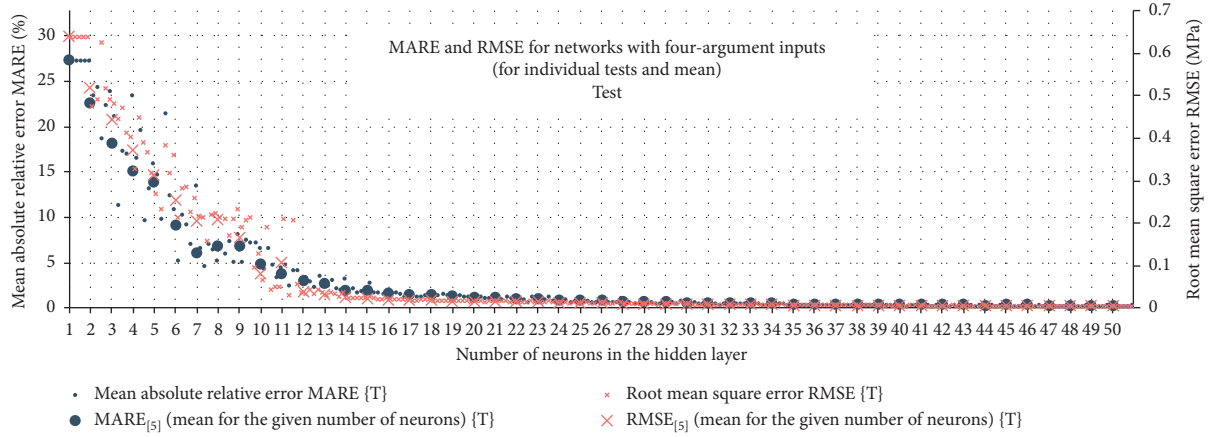


FIGURE 6: Values of MARE, MARE<sub>[5]</sub>, RMSE, and RMSE<sub>[5]</sub> for networks with the 12-sample input in function of the number of neurons in the hidden layer.

TABLE 2: Summary results of ANN computations using the 12-sample input in the testing stage.

Condition	Number of neurons in the hidden layer for which the condition is fulfilled	RMSE or RMSE <sub>[5]</sub> (MPa)	MARE or MARE <sub>[5]</sub> (%)	R <sup>2</sup> or R <sup>2</sup> <sub>[5]</sub>
MARE <sub>[5]</sub> ≤ 10%	≥ 6	0.25318	9.139	0.99291
MARE ≤ 10%	≥ 7	—	—	—
MARE ≤ 10% individually for the first time	4	0.39031	9.826	0.98338
MARE <sub>[5]</sub> ≤ 5%	≥ 10	0.08025	4.787	0.99895
MARE ≤ 5%	≥ 11	—	—	—
MARE ≤ 5% individually for the first time	7	0.21221	4.763	0.99509
Minimal MARE <sub>min</sub>	47	0.00397	0.239	1.00000

TABLE 3: Summary results of ANN computations using the 11-sample input in the testing stage; data for sample no. 1 were excluded from inputs.

Condition	Number of neurons in the hidden layer for which the condition is fulfilled	RMSE or RMSE <sub>[5]</sub> (MPa)	MARE or MARE <sub>[5]</sub> (%)	R <sup>2</sup> or R <sup>2</sup> <sub>[5]</sub>
MARE <sub>[5]</sub> ≤ 10%	≥ 6	0.23761	9.313	0.99379
MARE ≤ 10%	≥ 7	—	—	—
MARE ≤ 10% individually for the first time	5	0.23809	7.895	0.99385
MARE <sub>[5]</sub> ≤ 5%	≥ 9	0.16126	4.888	0.99687
MARE ≤ 5%	≥ 12	—	—	—
MARE ≤ 5% individually for the first time	8	0.22305	4.843	0.99460
Minimal MARE <sub>min</sub>	47	0.00381	0.212	1.00000

(12b). In this picture, there are included results for the networks taught on the 12-specimen set with 7 neurons in the hidden layer. Analogous figures for remaining networks are in Supplementary Materials (Figures A.18–A.20).

Results obtained for relative errors are depicted in Figure 9. It is distinct that the initial region of the stress-strain data was problematic; analysis of results showed this region merely corresponds to the linear region before

the plateau or the initial phase of the plateau. Mechanically, the beginning of the plateau is related to the compressive strength  $\sigma_c$ . One might suspect that, in case of small targets appearing in the denominator (in the aforementioned region, they are at least one order smaller than that for all other data), relative errors could be a misleading measure. Moreover, large values of RE<sub>i</sub>s from the beginning interval of stresses could significantly

TABLE 4: Summary results of ANN computations using the 11-sample input in the testing stage; data for sample no. 8 were excluded from inputs.

Condition	Number of neurons in the hidden layer for which the condition is fulfilled	RMSE or RMSE <sub>[5]</sub> (MPa)	MARE or MARE <sub>[5]</sub> (%)	R <sup>2</sup> or R <sub>[5]</sub> <sup>2</sup>
MARE <sub>[5]</sub> ≤ 10%	≥6	0.22818	9.631	0.99421
MARE ≤ 10%	≥8	—	—	—
MARE ≤ 10% individually for the first time	4	0.35459	9.168	0.98622
MARE <sub>[5]</sub> ≤ 5%	≥9	0.18558	4.873	0.99616
MARE ≤ 5%	≥13	—	—	—
MARE ≤ 5% individually for the first time	8	0.20295	4.945	0.99549
Minimal MARE <sub>min</sub>	48	0.00380	0.208	1.00000

TABLE 5: Summary results of ANN computations using the 11-sample input in the testing stage; data for sample no. 12 were excluded from inputs.

Condition	Number of neurons in the hidden layer for which the condition is fulfilled	RMSE or RMSE <sub>[5]</sub> (MPa)	MARE or MARE <sub>[5]</sub> (%)	R <sup>2</sup> or R <sub>[5]</sub> <sup>2</sup>
MARE <sub>[5]</sub> ≤ 10%	≥6	0.24622	9.380	0.99301
MARE ≤ 10%	≥7	—	—	—
MARE ≤ 10% individually for the first time	5	0.24736	6.159	0.99318
MARE <sub>[5]</sub> ≤ 5%	≥10	0.04595	4.095	0.99975
MARE ≤ 5%	≥12	—	—	—
MARE ≤ 5% individually for the first time	7	0.21344	4.569	0.99492
Minimal MARE <sub>min</sub>	50	0.00323	0.209	1.00000

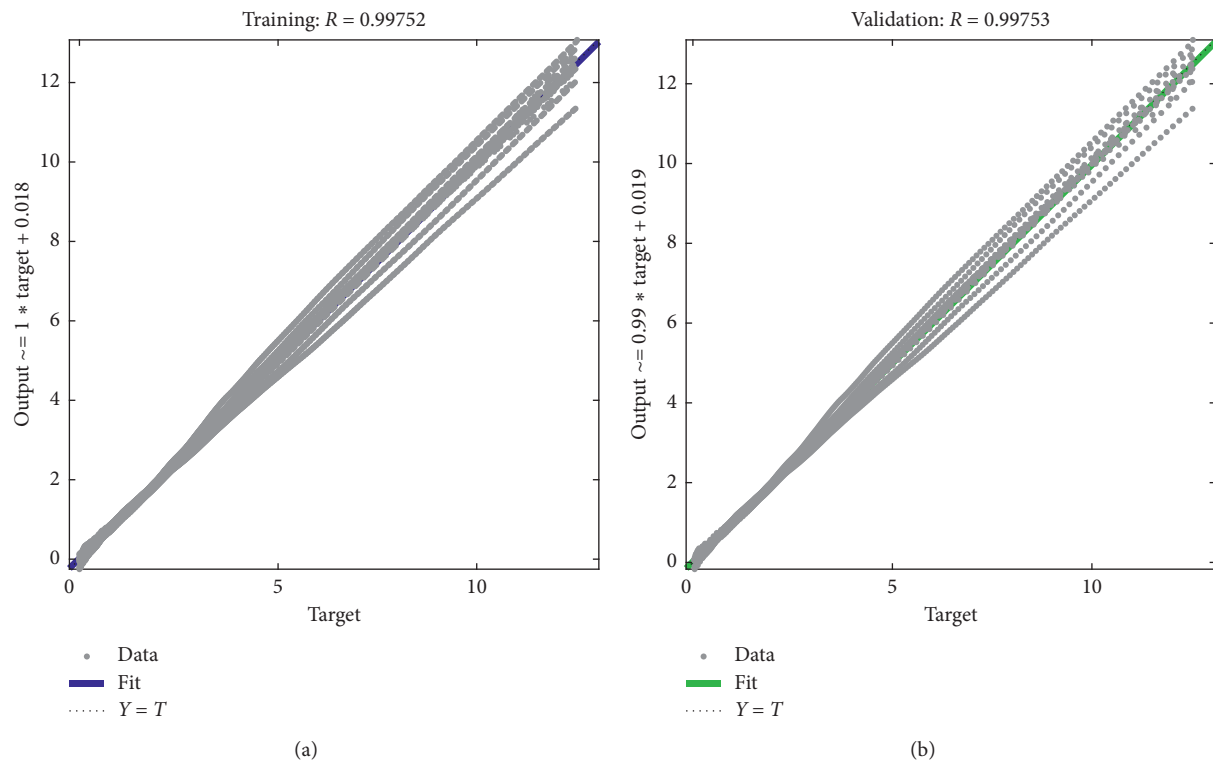


FIGURE 7: Continued.

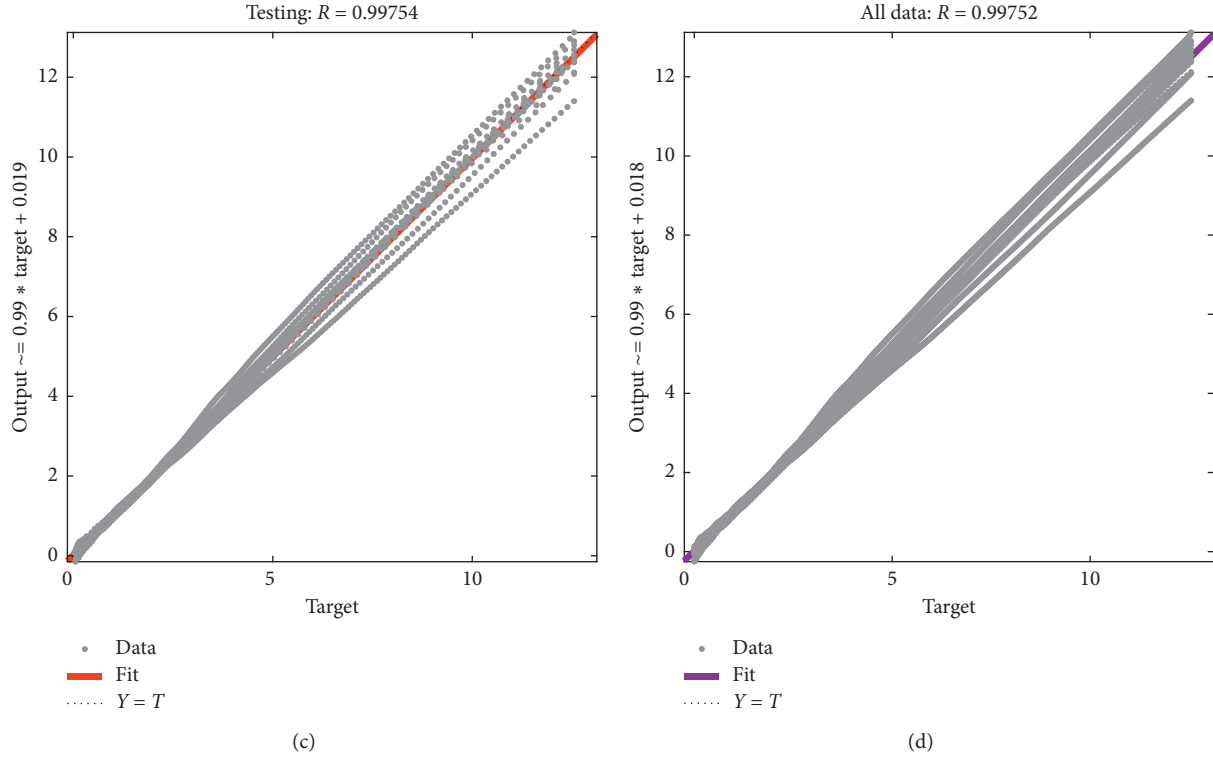


FIGURE 7: Correlation plot between the network's outputs and targets for all learning stages separately and combined for the network taught on the 12-sample set with 7 neurons in the hidden layer;  $\text{MARE} \leq 5\%$  individually for the first time.

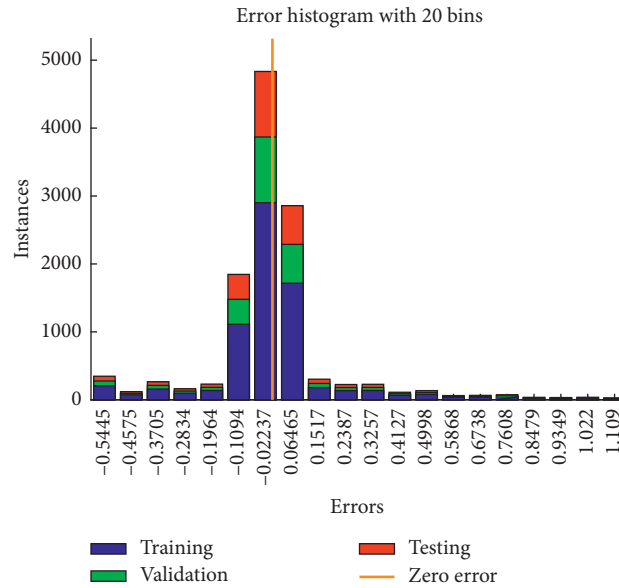


FIGURE 8: Error histograms for the network taught on the 12-specimen set with 7 neurons in the hidden layer;  $\text{MARE} \leq 5\%$  individually for the first time.

disturb MARE calculated for the whole network, implying the overrated number of required neurons in the hidden layer. This hypothesis seems reasonable, as the problem occurs even for the best networks (Figure 9(b)). Nevertheless, including the problematic results leaves potential

designers on the safe side. The solution to this issue was based on the approach proposed in [42] and will be described in Section 3.3. Analogous figures for remaining networks are in Supplementary Materials (Figures A.21–A.23).



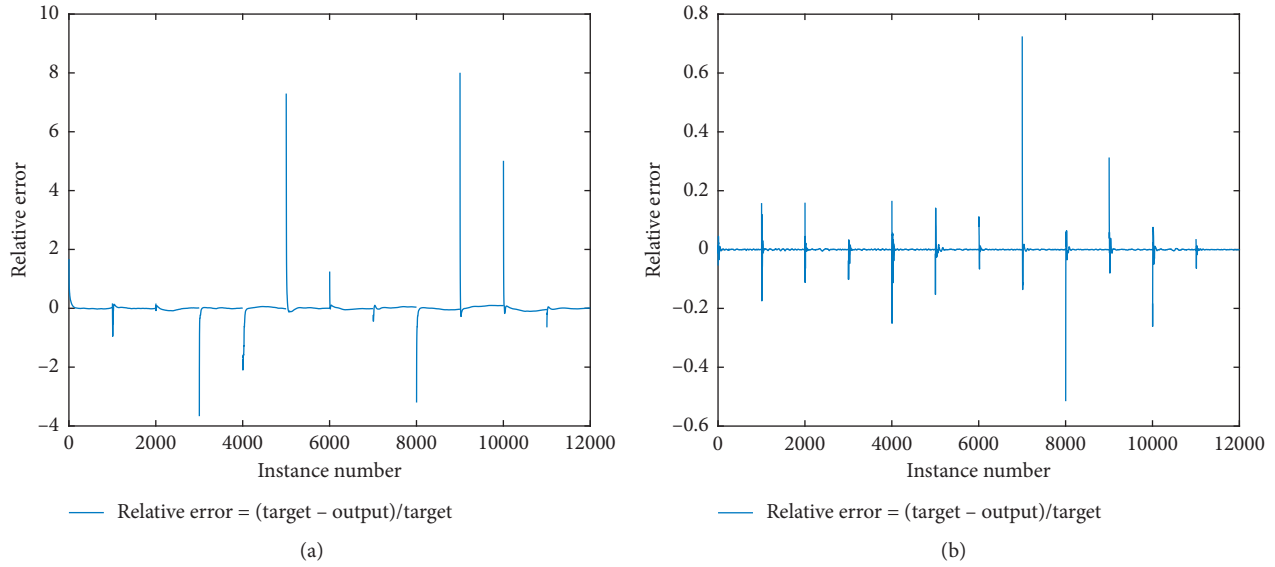


FIGURE 9: Relative errors for the network taught on the 12-specimen set. (a) With 7 neurons in the hidden layer and  $\text{MARE} \leq 5\%$  individually for the first time. (b) With 47 neurons in the hidden layer and best MARE.

TABLE 6: Summary results of testing ANNs on external data.

Tested sample	Number of neurons in the hidden layer	RMSE (MPa)	MARE (%)	$R^2$	MARE <sub>part</sub> (%)
1	5	5.21296	371.46	0.62446	—
	8	0.55000	30.672	0.99875	—
	8'	0.47269	10.871	0.99912	—
	47	6.02577	727.19	0.32892	—
8	4	0.74588	15.516	0.99860	14.634
	8	0.42953	9.9869	0.99953	7.241
	48	0.23199	10.100	0.99957	4.358
12	5	0.41103	12.252	0.99775	—
	7	0.26181	9.0404	0.99924	—
	7'	1.42536	57.417	0.99855	—
	50	28.2092	847.69	0.08041	—

3.2. *Approximation within and Extrapolation outside the Density Interval: Results of Testing the Taught Networks with Previously Excluded Aluminum Samples.* Characteristic networks, taught before on the 11-sample data sets, were appointed for testing on previously excluded data:

- (i) ANNs for which  $\text{MARE} \leq 10\%$  was reached for the first time
- (ii) ANNs for which  $\text{MARE} \leq 5\%$  was reached for the first time
- (iii) ANNs for which the minimum  $\text{MARE}_{\min}$  was reached

In order to include a remark on sensitivity of the networks to random weight assumption and to extrapolation, two of the tests were performed twice. Data for sample 1 were also used in another taught network with 8 neurons—the one for which  $\text{MARE} \leq 5\%$  was individually achieved for the second time in the algorithm. Similar calculations were performed for sample 12. Results for all above-mentioned networks are summarised in Table 6. Similarly as in Section 3.1, detailed description will be

provided here for the networks for which  $\text{MARE} \leq 5\%$ . Figures for the remaining networks are in Supplementary Materials. Figure 10 presents linear fits between networks' outputs and targets, corresponding to Figures A.24–A.26 in Supplementary Materials. Figure 11 gives error histograms, corresponding to Figures A.27–A.29 in Supplementary Materials. Figure 12 depicts  $\text{RE}_i$ ; further analogous figures are transferred to Supplementary Materials as Figures A.30–A.32. At the end of this section, graphs showing extrapolation results are depicted in Figures 13 and 14.

Generally, it can be stated that at least 8 neurons in the hidden layer are required in order to achieve results with the engineering accuracy of  $\text{MARE} \leq 10\%$ . One can also observe that linear fits show an overall trend of underestimation of stresses in case of the considered specimen, and this is more visible in the postplateau region.

Judging only from the results in the MARE column in Table 6, one could be misled that approximation of the networks is unsatisfactory; yet, if Figures 12 and A.30–A.32 are also examined, it can be seen that it is the initial region of

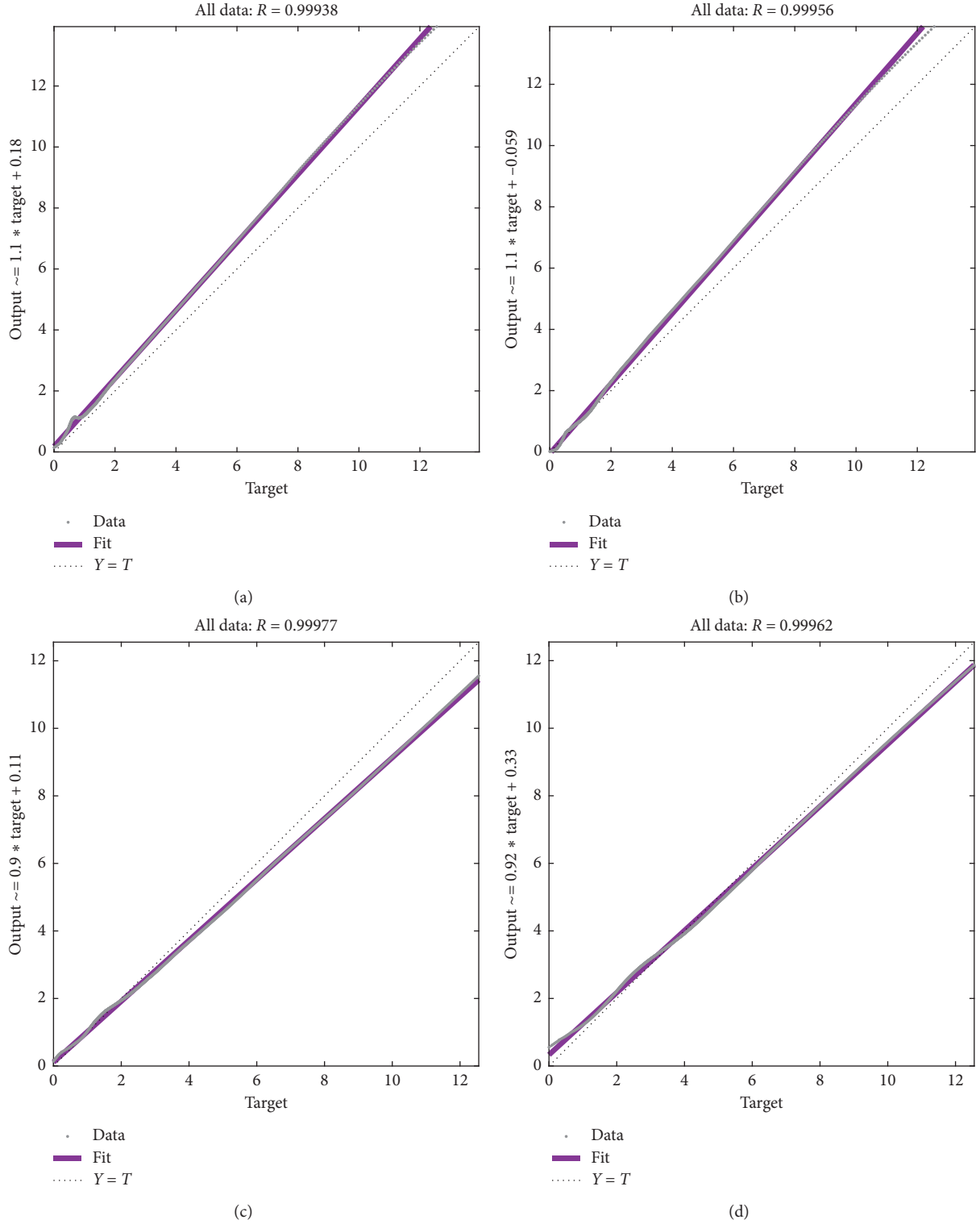


FIGURE 10: Correlation plots between networks' outputs and targets: (a) sample 1, 8 neurons; (b) sample 1, 8 neurons (alternative network); (c) sample 8, 8 neurons; (d) sample 12, 7 neurons.

the stress-strain relation that constituted a challenge for the ANNs. After initial  $\sim 100$   $\text{RE}_i$ s, they are uniformly distributed around one clear mean. As an example, partial means of absolute values ( $\text{MARE}_{\text{part}}$ ), calculated

disregarding the first 100 results for sample 8, are set in Table 6 in the last column; however, their purpose is only illustrative. Again, considerable initial  $\text{RE}_i$ s might be attributed to the fact that, in case of the beginning of the data

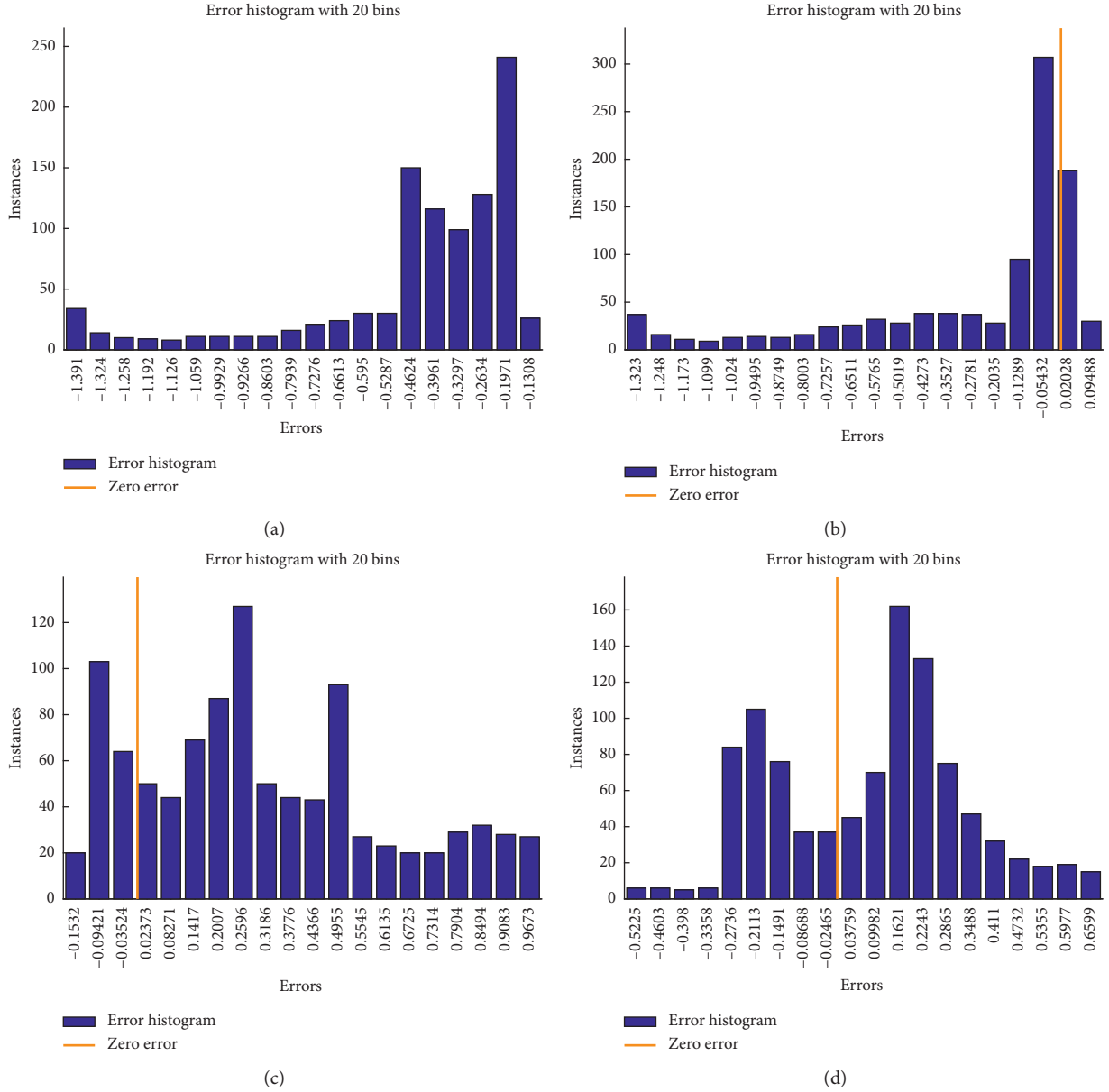


FIGURE 11: Error histograms: (a) sample 1, 8 neurons; (b) sample 1, 8 neurons (alternative network); (c) sample 8, 8 neurons; (d) sample 12, 7 neurons.

scope, denominators in relative errors are very small and that prototype samples might have behaved irregularly due to structural imperfections and this region is mechanically more sensitive to such an impact. As has been already mentioned, the solution for the described issue of the initial region was inspired by the study [42] and separate networks were trained for the initial stress-strain curve region. Again, this was performed only for sample 8, as an example. It is described in Section 3.3.

With regard to investigation of networks' extrapolation capabilities, previously taught networks were tested on data external for them. Sample 1 had apparent density below the taught density interval and was regular; on the contrary, sample 12 had apparent density above the upper-domain

boundary and was prototype. Figures 13 and 14 present stresses with respect to strains for corresponding samples which were outside the density interval used in teaching of networks. Additionally, these networks were tested twice (first two networks of a given hidden layer size for which  $\text{MARE} \leq 5\%$ ) in order to give insight into sensitivity to random weight assumption and influence of specimens' structure quality (regular vs prototype).

Regular samples turned less problematic for the networks to approximate, yet still the difference in the main criterial measure is visible:  $\text{MARE}_8 = 30.672\%$  and  $\text{MARE}_{8'} = 10.871\%$  (Figures 13(a) and 13(b), respectively). This confirms a well-known fact that internal network parameters (weights) may influence the approximation quality to a significant extent. This

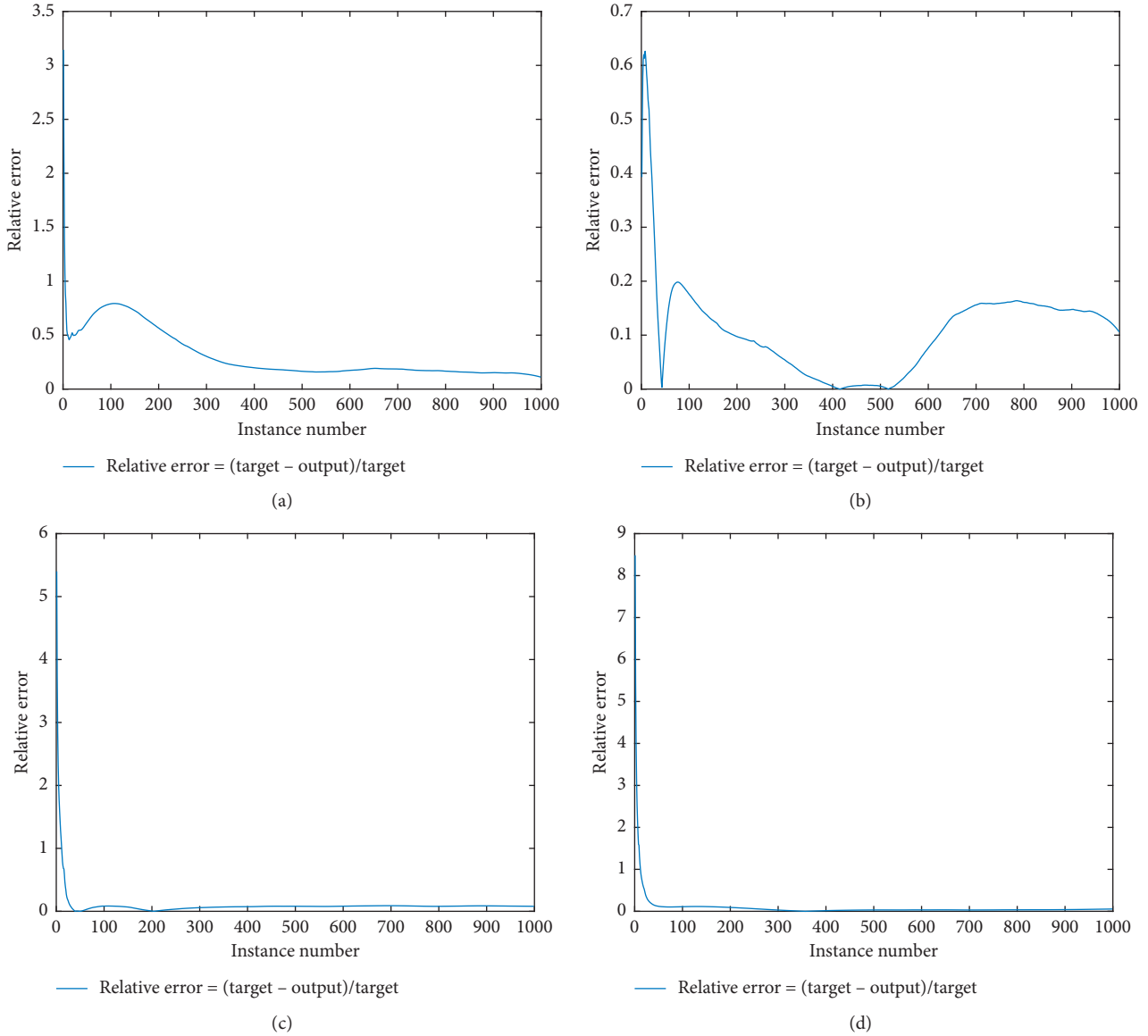


FIGURE 12: Relative errors: (a) sample 1, 8 neurons; (b) sample 1, 8 neurons (alternative network); (c) sample 8, 8 neurons; (d) sample 12, 7 neurons.

can be interpreted as both adverse and favourable effects. The negative aspect is the considerable sensitivity to finding different local minima in the error hypersurface and lack of the absolute minimum finding algorithm, but the positive is that, first, this can be to some extent countermanded and that, second, weights do not refer to physical material properties (compare [43]), in the mechanical sense, but are constants in mathematical expressions enabling the network to find local minima. The first step to at least partially diminish the disadvantageous randomness influence is to understand the phenomenon and input data possibly well (we are referring here to the general approximation theorem [44]). One of the very purposes of the study presented in this article was to understand better to what relation type the input data comply optimally, that is, what should the structure of the ANN with given learning parameters be. Additionally, in order to gain some control over random

processes, one could repeat teaching networks with similar structures, as was performed in our study, and examine obtained approximations, as is being done just now. It is worth mentioning that parameter sensitivity is an affliction present not only in neural networks but also in other attempts on building mechanical models, e.g., [45].

Comparison of plots in Figure 14 leads to a conclusion that, in case of extrapolation over the upper rim of the density interval and for an irregular sample, the two tested networks showed very different results:  $\text{MARE}_7 = 9.0404\%$  and  $\text{MARE}_{7_i} = 57.417\%$  (Figures 14(a) and 14(b), respectively). This fact can be explained by a few hypotheses. The first interpretation is that we see here nothing more but again just the influence of random weight assumption. However, it might also be that the upper density limit is problematic for networks since it is



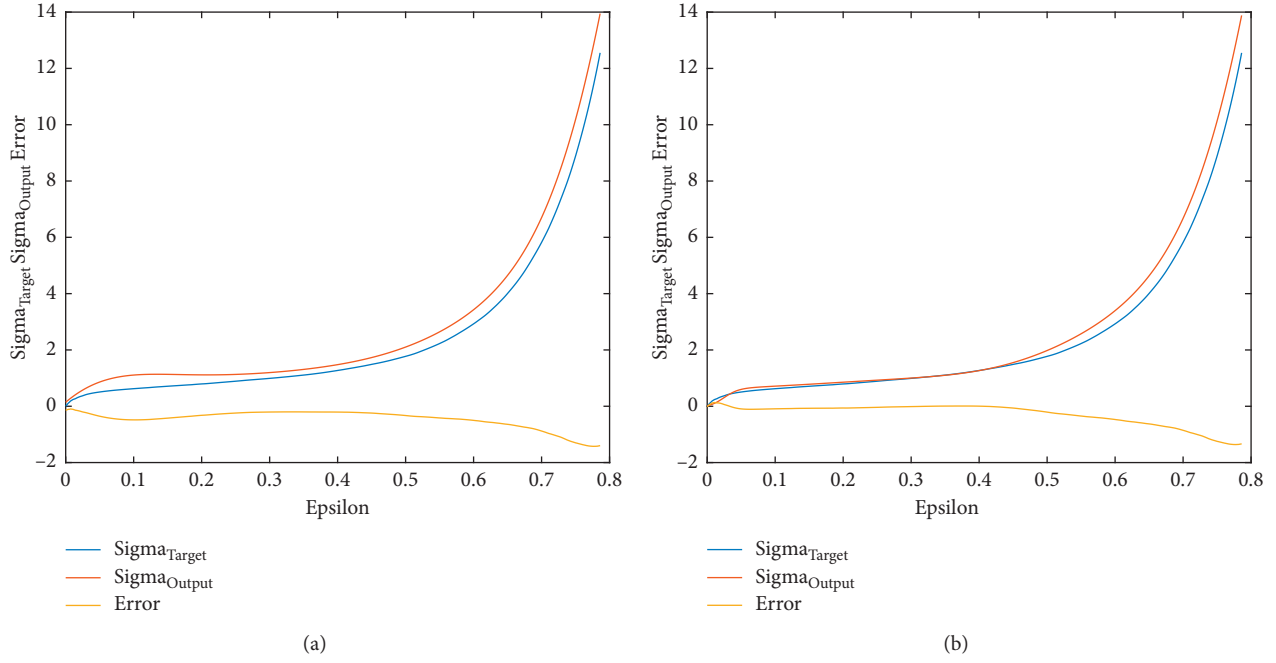


FIGURE 13: Extrapolated (output) and experimental (target) stresses with respect to strains for sample 1. (a, b) Results from two different networks with 8 neurons, previously taught, with the accuracy of  $\text{MARE} \leq 5\%$ .

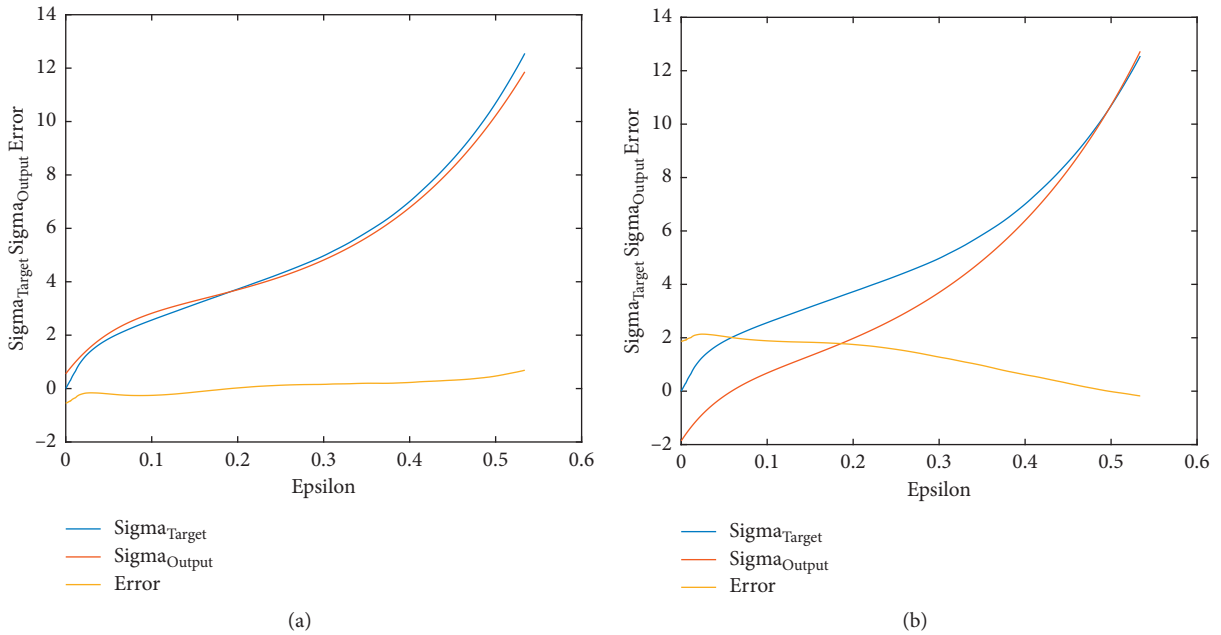


FIGURE 14: Extrapolated (output) and experimental (target) stresses with respect to strains for sample 12. (a, b) Results from two different networks with 7 neurons, previously taught, with the accuracy of  $\text{MARE} \leq 5\%$ .

the prototype samples which are heavier due to structural imperfections. In many cases, production of metal sponges is still at the noncommercial level and technology is being constantly improved [3, 10, 46]. At this point, a solution to this would be to supplement inputs for neural networks by more information about samples' structural quality [47].

**3.3. Additional Networks for the Initial Region of the Stress-Strain Curves.** Figure 15 shows MSE and MSE<sub>[5]</sub> for validation stages for all trained networks. It shows that 6 neurons are enough to obtain MSE for individual networks only slightly scattered.

Figure 16 depicts MARE, MARE<sub>[5]</sub>, RMSE, and RMSE<sub>[5]</sub> for all networks. Characteristic results are shown in Table 7.

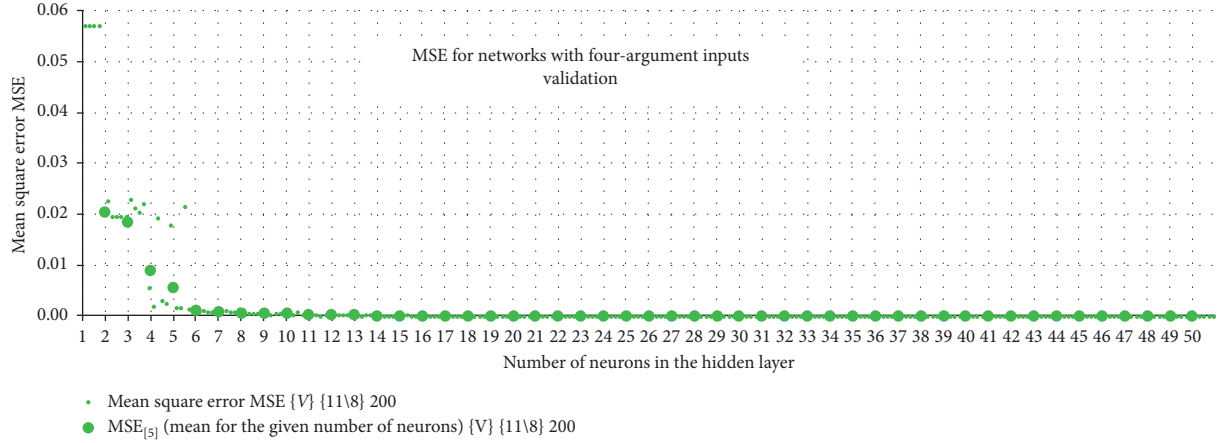


FIGURE 15: Values of MSE and  $MSE_{[5]}$  in function of the number of neurons in the hidden layer: case with 200 initial experimental data only.

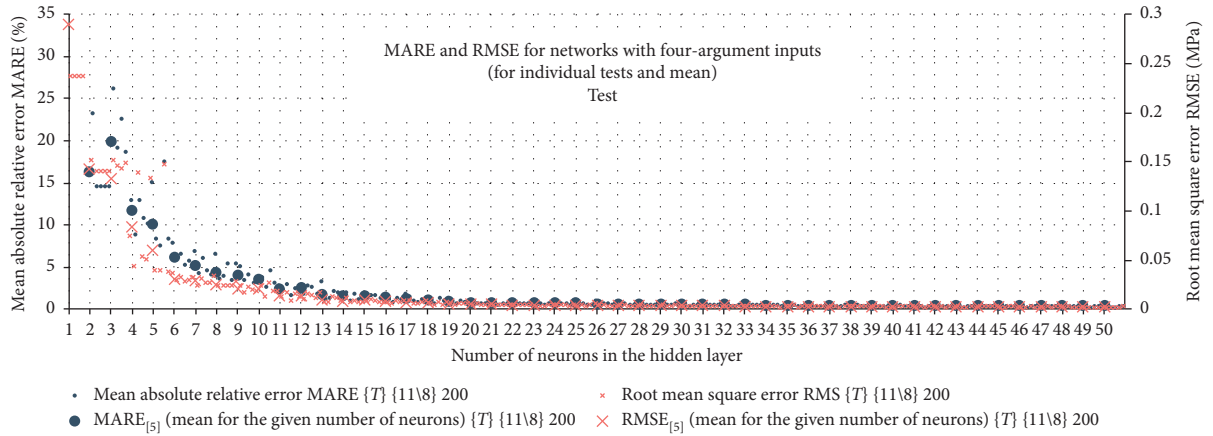


FIGURE 16: Values of MARE,  $MARE_{[5]}$ , RMSE, and  $RMSE_{[5]}$  in function of the number of neurons in the hidden layer: case with 200 initial experimental data only.

TABLE 7: Summary results of ANN computations in the testing stage: case with 200 initial experimental data only.

Condition	Number of neurons in the hidden layer for which the condition is fulfilled	RMSE or $RMSE_{[5]}$ (MPa)	MARE or $MARE_{[5]}$ (%)	$R^2$ or $R^2_{[5]}$
$MARE_{[5]} \leq 10\%$	$\geq 6$	0.03058	6.219	0.99728
$MARE \leq 10\%$	$\geq 6$	—	—	—
$MARE \leq 10\%$ individually for the first time	4	0.04450	8.970	0.99424
$MARE_{[5]} \leq 5\%$	$\geq 8$	0.02485	4.459	0.99820
$MARE \leq 5\%$	$\geq 10$	—	—	—
$MARE \leq 5\%$ individually for the first time	7	0.02474	4.343	0.99822
Minimal $MARE_{min}$	48	0.00204	0.336	0.99999

Threshold values of MARE: 10% and 5%, were first achieved for individual networks with 4 and 7 neurons, respectively.  $MARE_{[5]} \leq 10\%$  and  $MARE_{[5]} \leq 5\%$  were reached for 6 and 8 neurons, respectively. All values of MARE were smaller than the boundary values for 6 and 10 neurons, correspondingly. The minimum mean absolute relative error was  $MARE_{min} = 0.336\%$ , achieved in

the case with 48 neurons. Root mean square errors and coefficients of determination for all above instances are in Table 7.

In case of the coefficient of determination both for its mean and for consideration of individual instances, it turned out that 6 neurons guarantee the output-to-target relation at the assumed level of accuracy.

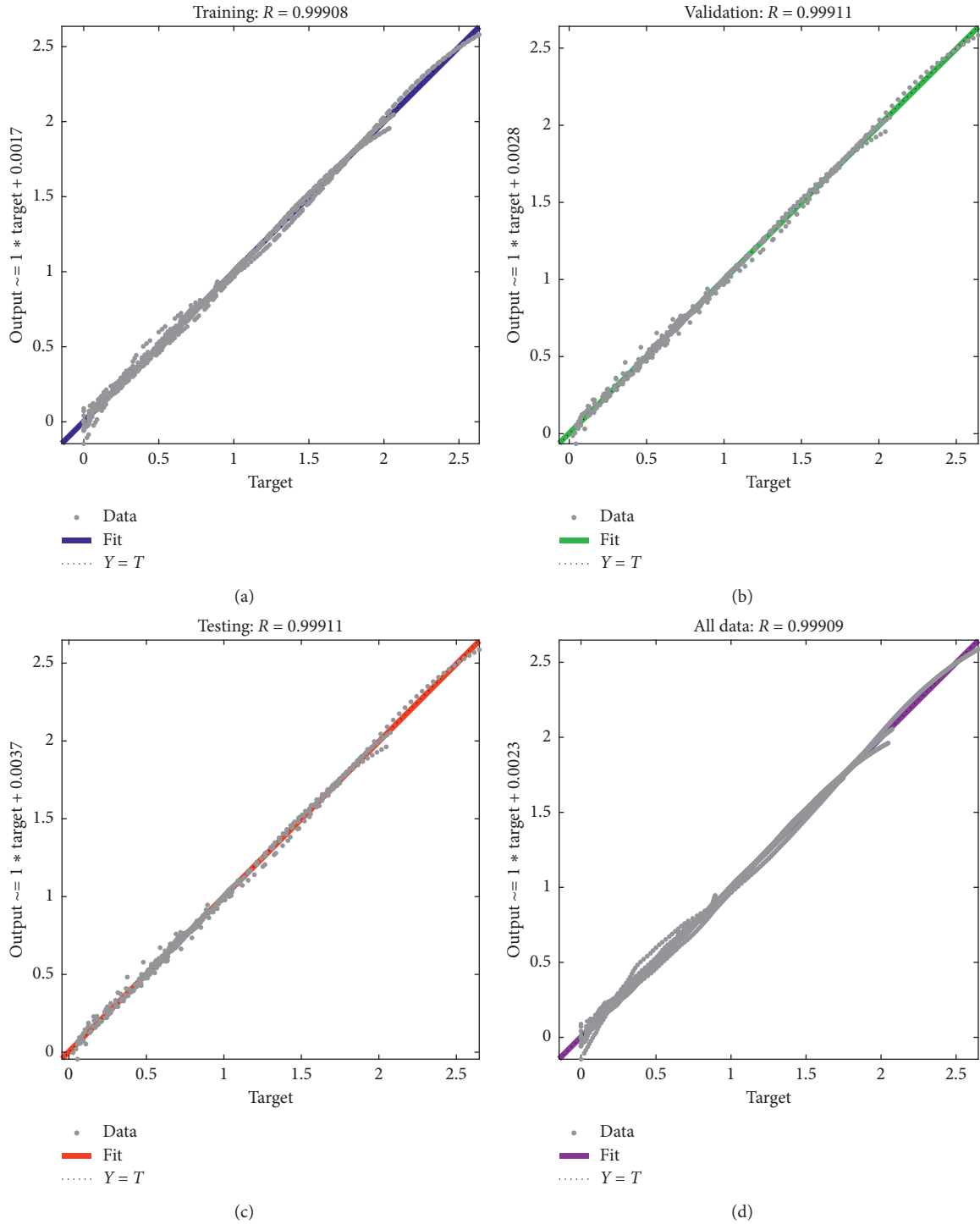


FIGURE 17: Correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set with 7 neurons in the hidden layer: case with 200 initial experimental data only.

Figures 17 and 18 present outputs from the networks plotted against targets for all learning stages separately (training, validation, and testing) as well as combined. The obtained equations are shown left to each of the graphs, and the Pearson coefficients are shown above them. Figure 17 is for the network with 7 neurons in the hidden layer, for which  $\text{MARE} \leq 5\%$ ; Figure 18 is for the network with 48 neurons.

Results for the network for which  $\text{MARE} \leq 10\%$  was reached for the first time are provided in Supplementary Materials (Figure A.33).

As for the errors, histograms are depicted in Figure 19. Results obtained for relative errors are presented in Figure 20. Analogously as before, figures in the main text give results for networks with 7 and 48 neurons. The

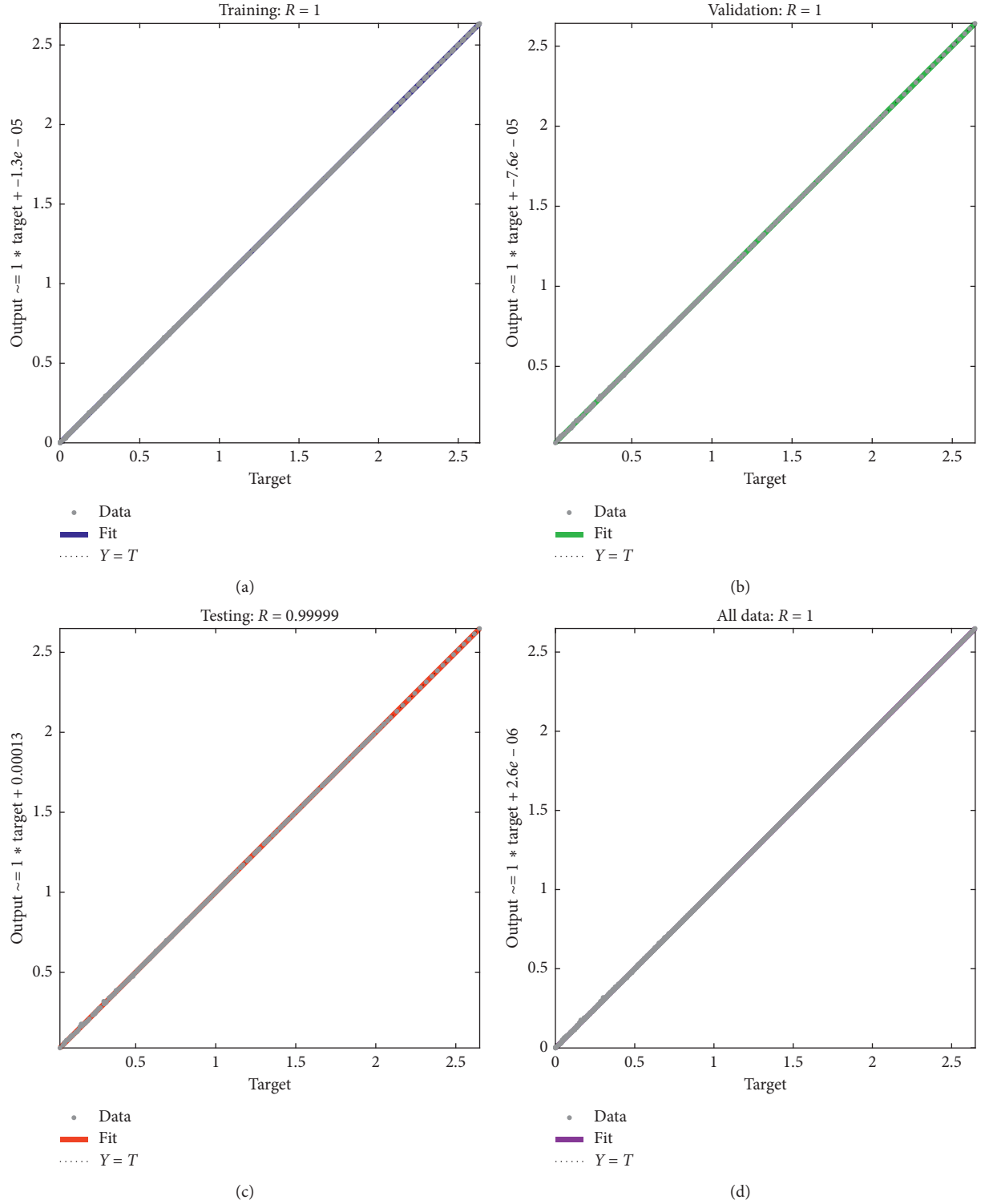


FIGURE 18: Correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set with 48 neurons in the hidden layer: case with 200 initial experimental data only.

histogram and the plot of  $RE_i$  for the 4-neuron ANN are in Supplementary Materials (Figures A.33 and A.34).

Three networks were chosen for testing on external data:

- (i) ANN with 4 neurons in the hidden layer, since it reached  $MARE \leq 10\%$  for the first time
- (ii) ANN with 7 neurons in the hidden layer, since it reached  $MARE \leq 5\%$  for the first time
- (iii) ANN with 48 neurons in the hidden layer, since it reached the minimum  $MARE_{\min}$



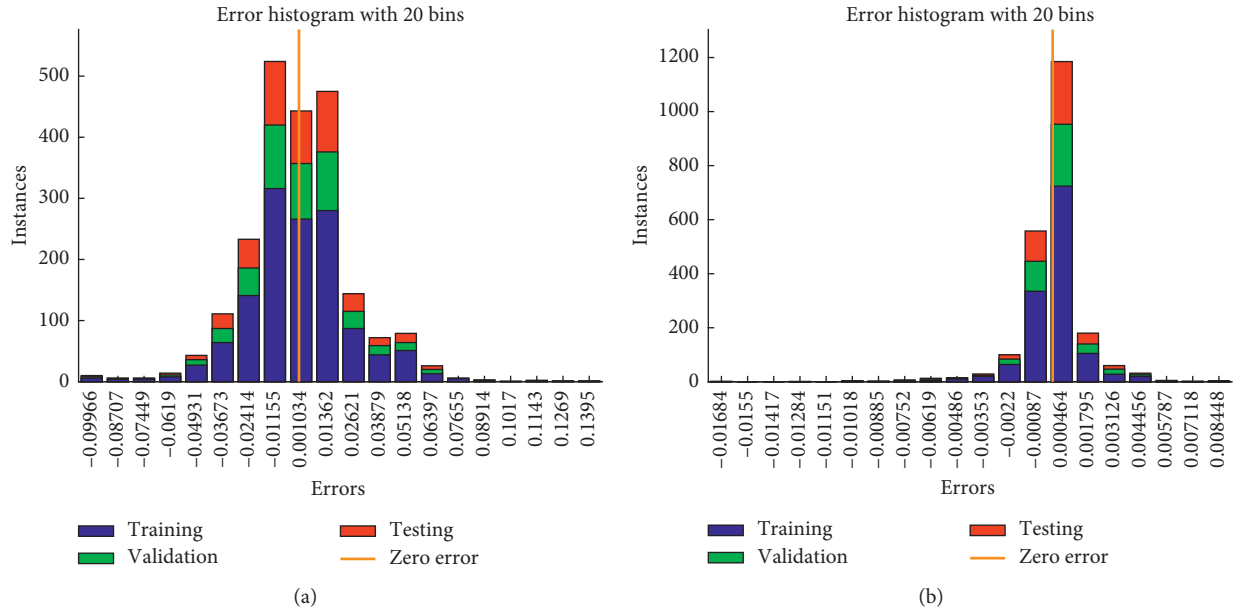


FIGURE 19: Error histograms for the case with 200 initial experimental data only: (a) network with 7 neurons; (b) network with 48 neurons.

TABLE 8: Results of testing ANNs on external data: case with 200 initial experimental data only.

Number of neurons in the hidden layer	RMSE (MPa)	MARE (%)	$R^2$
4	0.09174	20.310	0.99542
7	0.07430	7.193	0.98932
48	0.32127	21.112	0.81200

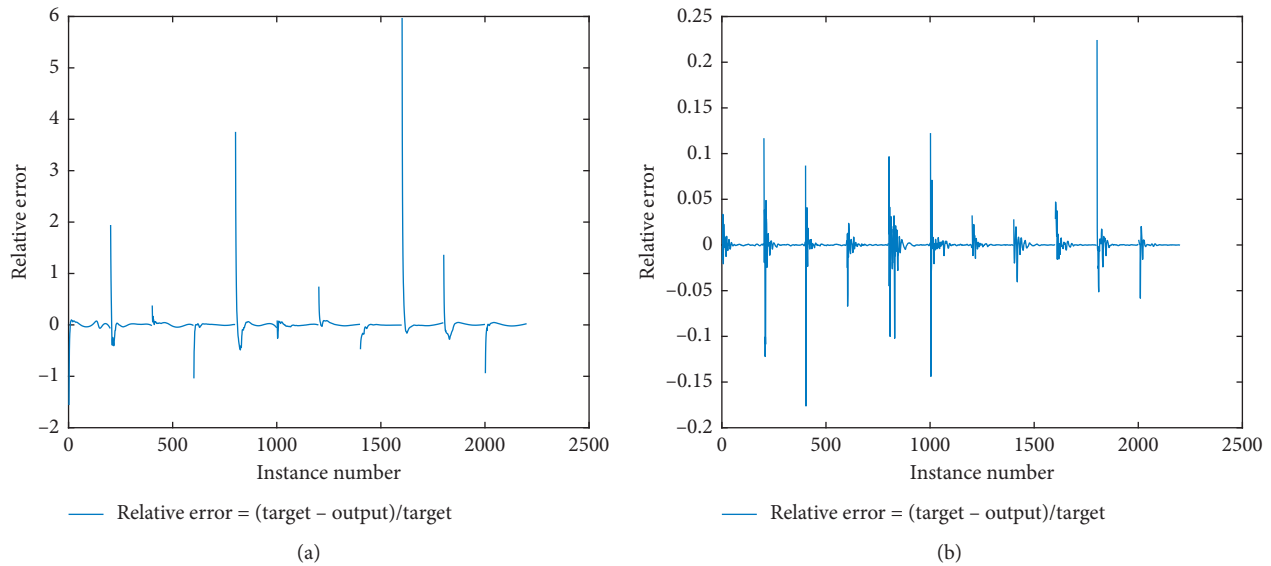


FIGURE 20: Relative errors for the case with 200 initial experimental data only: (a) network with 7 neurons; (b) network with 48 neurons.

Results are shown in Table 8. Figure 21 presents regression for NNs with 7 and 48 neurons. Figure 22 gives error histograms, and Figure 23 depicts  $ARE_i$  for the same networks. Respective graphs for the network with 4

neurons are moved to Supplementary Materials (Figures A.36–A.38).

It can be observed that 7 neurons in the hidden layer are required in order to achieve results with the engineering

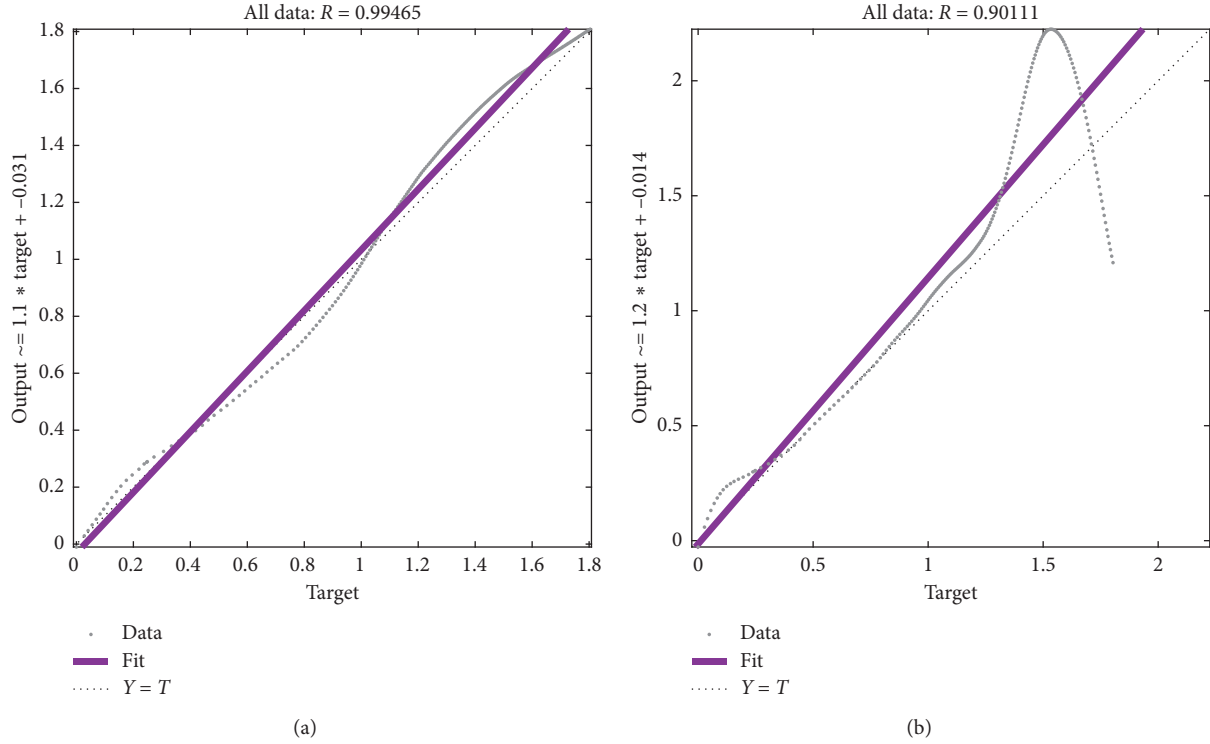


FIGURE 21: Correlation plots between the network's outputs and targets for the case with 200 initial experimental data only: (a) testing of the network taught on the 11-specimen set with 7 neurons; (b) testing of the network taught on the 11-specimen set with 48 neurons.

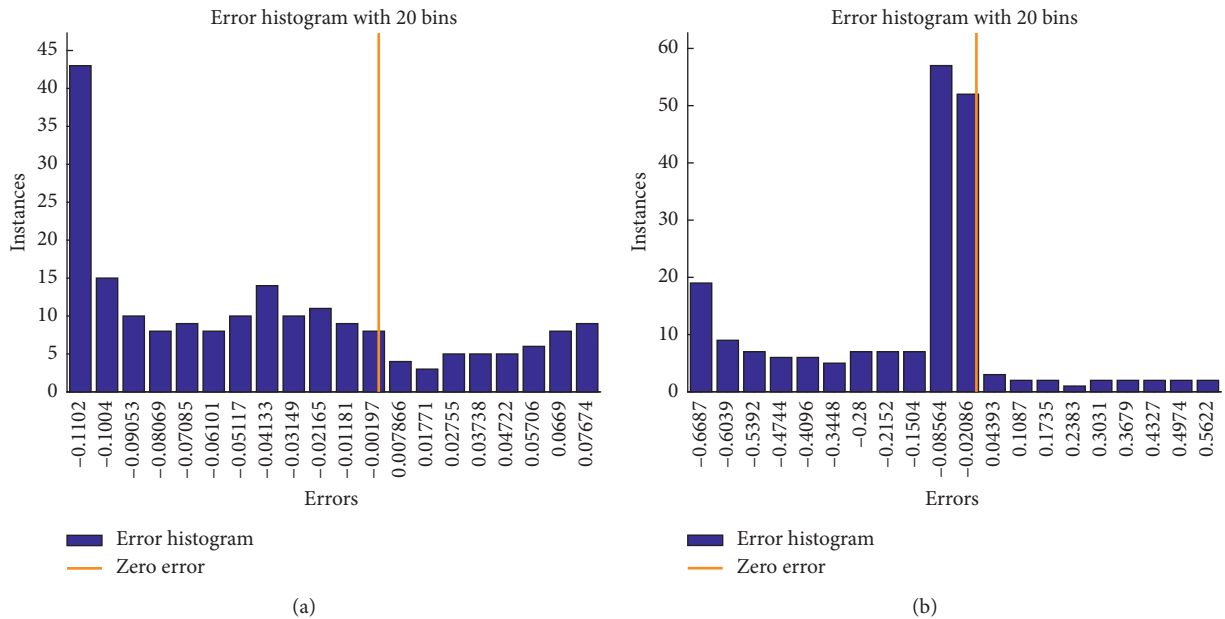


FIGURE 22: Error histograms for the case with 200 initial experimental data only: (a) testing of the network taught on the 11-specimen set with 7 neurons; (b) testing of the network taught on the 11-specimen set with 48 neurons.

accuracy of  $\text{MARE} = 7.193\% \leq 10\%$ . This result is a considerable improvement when compared to the value for the whole length of stress-strain curves ( $\text{MARE} = 9.989\%$ ). On

the contrary, 4 neurons are not enough ( $\text{MARE} > 20\%$ ). Also, 48 neurons give  $\text{MARE} > 21\%$ , which proves that the network used its potential for learning the particular data

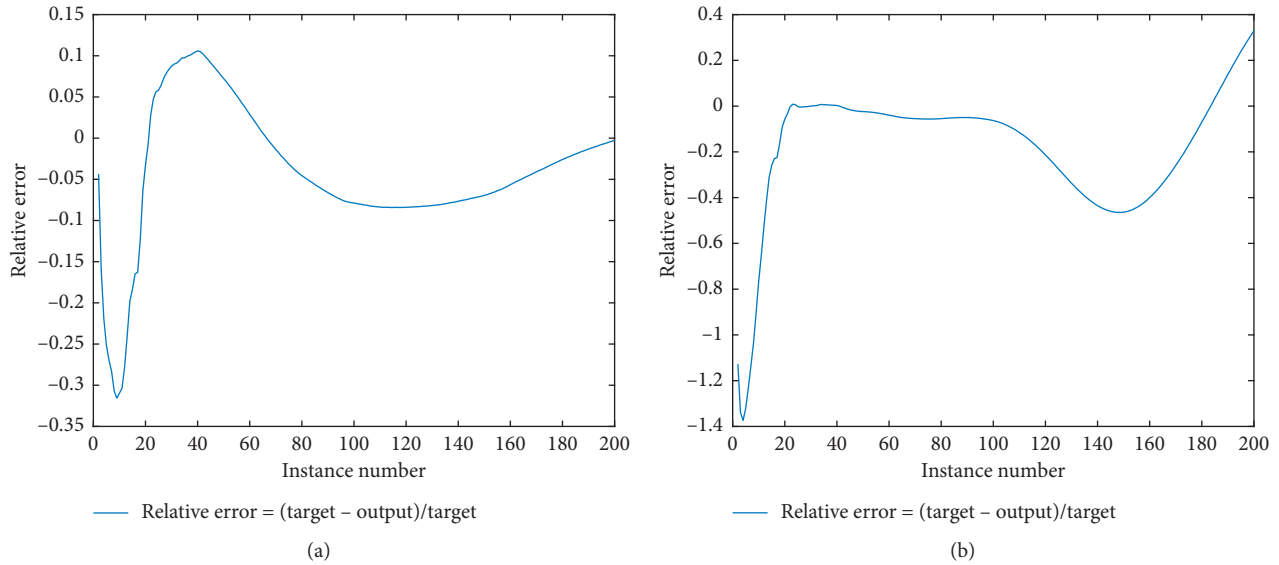


FIGURE 23: Relative errors for the case with 200 initial experimental data only: (a) testing of the network taught on the 11-specimen set with 7 neurons; (b) testing of the network taught on the 11-specimen set with 48 neurons.

instead for searching for the rule reflecting the stress-strain relation.

#### 4. Conclusions

The ANN algorithm described in the present article allows for building, learning, and comparing results of approximation of experimental stress-strain characteristics of open-cell aluminum. Choosing four criterial measures: mean absolute relative error (MARE), coefficient of determination between outputs and targets ( $R^2$ ), root mean square error (RMSE), and mean square error (MSE), and setting their limit values, one can obtain details for the neural network architecture which computes results with desired accuracy. Such structural specifications can be used in building networks for providing designers with models of strength material characteristics of open-cell aluminum with regard to its apparent density. Also, a trained NN could serve as a surrogate model or in metamodeling.

The performed analysis of relative errors led to the observation that the presented approach provides satisfactory engineering approximation (MARE < 10%) of the compressive behaviour. Moreover, the results can be improved with the same network architecture, yet modelling an additional ANN for the initial stress-strain region (MARE  $\approx$  7%). Detailed specifications are presented at the end of this section.

An additional conclusion comprises the comparison between the usage of all samples and exclusion of the testing sample from the input: with 12 specimens, and 1000 experimental data sets each, leaving one sample out does not affect calculations adversely. Such conduct secures an opportunity to test the approximation quality.

Removal of one sample from the input data set was also used for extrapolation quality assessment in reaching outside of the apparent density domain. It was proved that weight

parameters are sensitive to the local minima finding intrinsic to artificial intelligence, but repetitions of tries can limit adverse effects. Nevertheless, it should be noted that not only the local minima issue but also belonging of a specimen to a given lot was reflected in the results shown in this study—regular sample approximation did not show as much sensitivity as the prototype one.

As for the further research, authors would like to develop the present ANN study in order to elaborate a new proposition of the mechanical model of compressive behaviour of aluminum sponges in relation to the material's apparent density [13]. Moreover, to make the computations more detailed, it might be beneficial to include also the influence of other material's parameters as inputs, for example, the cell dimensions, such as strut thickness or length or specimen quality [47]. It would also be interesting to investigate efficiency of other NN algorithms, e.g., SVM and ELM.

Now the specifications are summarised and given in a conscience list as follows:

- (i) The general structure of networks was two-layer feedforward networks consisting of one hidden layer with the tansig activation function and one output layer with the linear activation function.
- (ii) The teaching algorithm was the Levenberg–Marquardt algorithm with MSE as a performance function. Other learning parameters are given in Table 1.
- (iii) Data were divided in proportions of (60 + 20 + 20)% between training, validation, and testing. Processing with the Matlab function `mapminmax` was used.
- (iv) The approximated relation was given by equation (2):  $\sigma = f(\varepsilon, \rho)$ . For better results, augmentation of inputs by the parameter describing the sample group was done.

- (v) In case of the assumed criterion  $MARE \approx 10\%$ , its fulfillment was possible with 7 and 8 neurons in the hidden layer.
- (vi) An increase of accuracy assessed with MARE to the level of about 7% was achieved if the same network was used in the plateau and postplateau regions of the stress-strain relation, while the preplateau region was modelled with a separate 7-neuron network.

## Data Availability

The stress-strain experimental data and apparent density data for the tested samples used to support the findings of this study have not been made available.

## Disclosure

The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## Conflicts of Interest

The authors declare no conflicts of interest.

## Acknowledgments

The authors would like to express their gratitude to the following collaborators who helped in production and testing of the material: B. Lipowska, K. Wańczyk, A. Kwiecień, and B. Zajac. The authors would also like to indicate with gratitude that precursors for manufacturing of aluminum samples were gifted by Recticel Flexible Foams (Belgium). The funding for the presented research and publication was from the Cracow University of Technology (Kraków, Poland). The production of the aluminum sponge and experimental tests were partially supported by AGH University (Kraków, Poland; grant no. 15.11.130.488).

## Supplementary Materials

Supplementary materials for the present article give results for modelling with the use of the network with 4 neurons, both for the case when 12 samples were used as the data source and for the instance with 11 samples constituting the input material. The results have only auxiliary character for the whole research, so they were moved to this section; yet they might be of interest to some readers. The included figures are as follows: Figure A.1: values of MSE and  $MSE_{[5]}$  in function of the number of neurons in the hidden layer for networks for the 11-sample input, with sample no. 1 excluded. Figure A.2: values of MSE and  $MSE_{[5]}$  in function of the number of neurons in the hidden layer for networks for the 11-sample input, with sample no. 8 excluded. Figure A.3: values of MSE and  $MSE_{[5]}$  in function of the number of neurons in the hidden layer for networks for the 11-sample input, with sample no. 12 excluded. Figure A.4: values of MARE,  $MSE_{[5]}$ , RMSE, and  $RMSE_{[5]}$  for networks with the 11-sample input in function of the number of neurons in the

hidden layer, with data for sample no. 1 excluded from inputs. Figure A.5: values of MARE,  $MSE_{[5]}$ , RMSE, and  $RMSE_{[5]}$  for networks with the 11-sample input in function of the number of neurons in the hidden layer, with data for sample no. 8 excluded from inputs. Figure A.6: values of MARE,  $MSE_{[5]}$ , RMSE, and  $RMSE_{[5]}$  for networks with the 11-sample input in function of the number of neurons in the hidden layer, with data for sample no. 12 excluded from inputs. Figure A.7: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 12-specimen set with 4 neurons in the hidden layer;  $MARE \leq 10\%$  individually for the first time. Figure A.8: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 12-specimen set with 47 neurons in the hidden layer and with best MARE. Figure A.9: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set (sample no. 1. excluded) with 5 neurons in the hidden layer;  $MARE \leq 10\%$  individually for the first time. Figure A.10: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set (sample no. 1. excluded) with 8 neurons in the hidden layer;  $MARE \leq 5\%$  individually for the first time. Figure A.11: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set (sample no. 1. excluded) with 47 neurons in the hidden layer and with best MARE. Figure A.12: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set (sample no. 8. excluded) with 4 neurons in the hidden layer;  $MARE \leq 10\%$  individually for the first time. Figure A.13: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set (sample no. 8. excluded) with 8 neurons in the hidden layer;  $MARE \leq 5\%$  individually for the first time. Figure A.14: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set (sample no. 8. excluded) with 48 neurons in the hidden layer and with best MARE. Figure A.15: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set (sample no. 12. excluded) with 5 neurons in the hidden layer;  $MARE \leq 10\%$  individually for the first time. Figure A.16: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set (sample no. 12. excluded) with 7 neurons in the hidden layer;  $MARE \leq 5\%$  individually for the first time. Figure A.17: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set (sample no. 12. excluded) with 50 neurons in the hidden layer and with best MARE. Figure A.18: error histograms for networks for which  $MARE \leq 10\%$  individually for the first time: (a) 12-

sample input, 4 neurons; (b) 11-sample input (no. 1 excl.), 5 neurons; (c) 11-sample input (no. 8 excl.), 4 neurons; (d) 11-sample input (no. 12 excl.), 5 neurons. Figure A.19: error histograms for networks for which  $MARE \leq 5\%$  individually for the first time: (a) 11-sample input (no. 1 excl.), 8 neurons; (b) 11-sample input (no. 8 excl.), 8 neurons; (c) 11-sample input (no. 12 excl.), 7 neurons. Figure A.20: error histograms for networks for the best MARE: (a) 12-sample input, 47 neurons; (b) 11-sample input (no. 1 excl.), 47 neurons; (c) 11-sample input (no. 8 excl.), 48 neurons; (d) 11-sample input (no. 12 excl.), 50 neurons. Figure A.21: relative errors for networks for which  $MARE \leq 10\%$  individually for the first time: (a) 12-sample input, 4 neurons; (b) 11-sample input (no. 1 excl.), 5 neurons; (c) 11-sample input (no. 8 excl.), 4 neurons; (d) 11-sample input (no. 12 excl.), 5 neurons. Figure A.22: relative errors for networks for which  $MARE \leq 5\%$  individually for the first time: (a) 11-sample input (no. 1 excl.), 8 neurons; (b) 11-sample input (no. 8 excl.), 8 neurons; (c) 11-sample input (no. 12 excl.), 7 neurons. Figure A.23: relative errors for networks for the best MARE: (a) 11-sample input (no. 1 excl.), 47 neurons; (b) 11-sample input (no. 8 excl.), 48 neurons; (c) 11-sample input (no. 12 excl.), 50 neurons. Figure A.24: regression line from testing of the network taught on the 11-specimen set (sample 1 excl.): (a) structure with 5 neurons; (b) structure with 47 neurons. Figure A.25: regression line from testing of the network taught on the 11-specimen set (sample 8 excl.): (a) structure with 4 neurons; (b) structure with 48 neurons. Figure A.26: regression line from testing of the network taught on the 11-specimen set (sample 12 excl.): (a) structure with 5 neurons; (b) structure with 50 neurons. Figure A.27: error histogram from testing of the network taught on the 11-specimen set (sample 1 excl.): (a) structure with 5 neurons; (b) structure with 47 neurons. Figure A.28: error histogram from testing of the network taught on the 11-specimen set (sample 8 excl.): (a) structure with 4 neurons; (b) structure with 48 neurons. Figure A.29: error histogram from testing of the network taught on the 11-specimen set (sample 12 excl.): (a) structure with 5 neurons; (b) structure with 50 neurons. Figure A.30: relative errors from testing of the network taught on the 11-specimen set (sample 1 excl.): (a) structure with 5 neurons; (b) structure with 47 neurons. Figure A.31: relative errors from testing of the network taught on the 11-specimen set (sample 8 excl.): (a) structure with 4 neurons; (b) structure with 48 neurons. Figure A.32: relative errors from testing of the network taught on the 11-specimen set (sample 12 excl.): (a) structure with 5 neurons; (b) structure with 50 neurons. Figure A.33: correlation plots between the network's outputs and targets for all learning stages separately and combined for the network taught on the 11-specimen set (no. 8 excl.), with 4 neurons in the hidden layer: case with 200 initial experimental data only. Figure A.34: error histogram for the network taught on the 11-specimen set (no. 8 excl.), with 4 neurons in the hidden layer: case with 200 initial experimental data only. Figure A.35: relative errors for the network taught on the 11-specimen set (no. 8 excl.), with 4 neurons in the hidden layer: case with 200 initial experimental data only. Figure A.36: regression line from testing of the network taught on the 11-

specimen set with 4 neurons: case with 200 initial experimental data only, sample 8. Figure A.37: error histogram from testing of the network taught on the 11-specimen set with 4 neurons: case with 200 initial experimental data only, sample 8. Figure A.38: relative errors from testing of the network taught on the 11-specimen set with 4 neurons: case with 200 initial experimental data only, sample 8. (*Supplementary Materials*)

## References

- [1] J. Banhart and J. Baumeister, "Production methods for metallic foams," *MRS Proceedings*, vol. 521, pp. 121–132, 1998.
- [2] J. Sobczak, A. Wojciechowski, L. Boyko, L. Drenchev, P. Darlak, and P. Dudek, *Materiały Wysokoporowate*, Instytut Odlewnictwa, Kraków, Poland, 2005.
- [3] F. García-Moreno, "Commercial applications of metal foams: their properties and production," *Materials*, vol. 9, no. 2, pp. 85–111, 2016.
- [4] H. Degischer and B. Kriszt, *Handbook of Cellular Metals: Production, Processing, Applications*, Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 2002.
- [5] Duocel Copper Foam, 2018, <https://ergaerospace.com/materials/duocel-copper-foam>.
- [6] F. Simancik, J. Jerz, J. Kováčik, and P. Minár, "Aluminum foam—a new light-weight structural material," *Kovove Materialy*, vol. 35, pp. 265–277, 1997.
- [7] D. C. Dunand, "Processing of titanium foams," *Advanced Engineering Materials*, vol. 6, no. 6, pp. 369–376, 2004.
- [8] J. Banhart, "Gold and gold alloy foams," *Gold Bulletin*, vol. 41, no. 3, pp. 251–256, 2008.
- [9] S. R. Arwade, J. F. Hajjar, B. W. Schafer, M. Moradi, B. H. Smith, and S. Szyniszewski, "Steel foam material processing, properties, and potential structural applications," in *Proceedings of 2011 NSF Engineering Research and Innovation Conference*, Atlanta, Georgia, 2011.
- [10] A. M. Stręk, B. Lipowska, and K. Wańczyk, "Selected aspects of manufacturing of aluminum sponge," *Archives of Metallurgy and Materials*, vol. 3, pp. 1145–1150, 2019.
- [11] P. Kasza, B. Lipowska, R. B. Pęcherski, A. M. Stręk, and K. Wańczyk, "Compression of open-cell aluminum," *Engineering Transactions*, vol. 64, no. 4, pp. 629–634, 2016.
- [12] A. M. Stręk, "Assessment of strength and functional properties of cellular materials," Doctoral thesis, AGH University, Kraków, Poland, 2017.
- [13] A. M. Stręk and M. Dudzik, "An ANN model of stress-strain relationship for aluminium sponge in uniaxial compression," *Journal of Theoretical and Applied Mechanics*, 2020.
- [14] A. M. Stręk, M. Dudzik, A. Kwiecień, K. Wańczyk, and B. Lipowska, "Verification of application of ANN modelling for compressive behaviour of metal sponges," *Engineering Transactions*, vol. 67, pp. 271–288, 2019.
- [15] M. Jakubek, "Application of artificial neural networks in selected problems of experimental mechanics and structural engineering," Doctoral thesis, Cracow University of Technology, Kraków, Poland, 2007.
- [16] M. Kurzyński, *Metody Sztucznej Inteligencji dla Inżynierów* (English Title: *Methods of Artificial Intelligence for Engineers*), Państwowa Wyższa Szkoła Zawodowa im. Witelona, Legnica, Poland, 2008.
- [17] M. Lefik, *Zastosowanie Sztucznych Sieci Neuronowych W Mechanice I W Inżynierii* (English Title: *Application of*



- Artificial Neural Networks in Mechanics and Engineering*), Wydawnictwo Politechniki Łódzkiej, Łódź, Poland, 2005.
- [18] Z. Waszczyszyn and L. Ziemianski, "Neural networks in the identification analysis of structural mechanics problems," in *Parameter Identification of Materials and Structures*, Z. Mróz and G. E. Stavroulakis, Eds., pp. 265–340, Springer, New York, NY, USA, 2005.
  - [19] W. Sumelka and T. Łodygowski, "Reduction of the number of material parameters by ANN approximation," *Computational Mechanics*, vol. 52, no. 2, pp. 287–300, 2013.
  - [20] T. L. Paez, "Neural networks in mechanical system simulation, identification, and assessment," *Shock and Vibration*, vol. 1, no. 2, pp. 177–199, 1993.
  - [21] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: a review of two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 60, pp. 97–116, 2017.
  - [22] M. La Rocca and C. Perna, "Model selection for neural network models: a statistical perspective," in *Computational Network Theory: Theoretical Foundations and Applications*, M. Dehmer, F. Emmert-Streib, and S. Pickl, Eds., Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 1st edition, 2015.
  - [23] M. Bashiri and A. Farshbaf Geranmayeh, "Tuning the parameters of an artificial neural network using central composite design and genetic algorithm," *Scientia Iranica*, vol. 18, no. 6, pp. 1600–1608, 2011.
  - [24] U. Anders and O. Korn, "Model selection in neural networks," *Neural Networks*, vol. 12, no. 2, pp. 309–323, 1999.
  - [25] G. Notton, C. Voyant, A. Fouilloy, J. L. Duchaud, and M. L. Nivet, "Some applications of ANN to solar radiation estimation and forecasting for energy applications," *Applied Sciences*, vol. 9, no. 1, pp. 209–229, 2019.
  - [26] A. Oken, "An introduction to and applications of neural networks," 2019, <https://www.whitman.edu/Documents/Academics/Mathematics/2017/Oken.pdf>.
  - [27] T. Mareš, E. Janouchová, and A. Kučerová, "Artificial neural networks in the calibration of nonlinear mechanical models," *Advances in Engineering Software*, vol. 95, pp. 68–81, 2016.
  - [28] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. De Jesus, *Neural Network Design*, University of Colorado Bookstore, Boulder, CO, USA, 2nd edition, 2014.
  - [29] A. Famili, W. Shen, R. Weber, and E. Simoudis, "Data pre-processing and intelligent data analysis," *Intelligent Data Analysis*, vol. 1, no. 1–4, pp. 3–23, 1997.
  - [30] P. Koprinkova and M. Petrova, "Data-scaling problems in neural-network training," *Engineering Applications of Artificial Intelligence*, vol. 12, no. 3, pp. 281–296, 1999.
  - [31] M. Shanker, M. Y. Hu, and M. S. Hung, "Effect of data standardization on neural network training," *Omega*, vol. 24, no. 4, pp. 385–397, 1996.
  - [32] L. J. Gibson and M. F. Ashby, "The mechanics of three-dimensional cellular materials," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 382, no. 1782, pp. 43–59, 1982.
  - [33] J. Banhart and J. Baumeister, "Deformation characteristics of metal foams," *Journal of Materials Science*, vol. 33, no. 6, pp. 1431–1440, 1998.
  - [34] Doreswamy and C. M. Vastrad, "Performance analysis of neural network models for oxazolines and oxazoles derivatives descriptor dataset," *International Journal of Information Sciences and Techniques*, vol. 3, no. 6, pp. 1–15, 2013.
  - [35] S. Kanaun and O. Tkachenko, "Representative volume element and effective elastic properties of open cell foam materials with random microstructures," *Journal of Mechanics of Materials and Structures*, vol. 2, no. 8, pp. 1607–1628, 2007.
  - [36] W.-Y. Jang, A. M. Kraynik, and S. Kyriakides, "On the microstructure of open-cell foams and its effect on elastic properties," *International Journal of Solids and Structures*, vol. 45, no. 7–8, pp. 1845–1875, 2008.
  - [37] H. Demuth, M. Beale, and M. Hagan, *Neural Network Toolbox 6 User's Guide*, The MathWorks Inc., Natick, MA, USA, 2009.
  - [38] Mathworks Documentation: Mapminmax, 2019, <https://www.mathworks.com/help/deeplearning/ref/mapminmax.html>.
  - [39] Matlab and Automatic Target Normalization: Mapminmax. Don't Trust Your Matlab Framework! 2019, <https://neuralsniffer.wordpress.com/2010/10/17/matlab-and-automatic-target-normalization-mapminmax-dont-trust-your-matlab-framework/>.
  - [40] K. Madsen, H. B. Nielsen, and O. Tingleff, "Methods for non-linear least squares problems," Informatics and Mathematical Modelling, Technical University of Denmark, Kongens Lyngby, Denmark, 2nd edition, 2004.
  - [41] M. Dudzik, *Współczesne Metody Projektowania, Weryfikacji Poprawności I Modelowania Zjawisk Trakcji Elektrycznej (English Title: Modern Methods of Designing, Verification and Modelling of Phenomena Concerning Electric Traction)*, Wydawnictwo Politechniki Krakowskiej, Kraków, Poland, 2018.
  - [42] A. Dudzik, W. Snoch, P. Borowiecki et al., "Asymmetric reduction of ketones and  $\beta$ -keto esters by (S)-1-phenylethanol dehydrogenase from denitrifying bacterium *Aromatoleum aromaticum*," *Applied Microbiology and Biotechnology*, vol. 99, no. 12, pp. 5055–5069, 2015.
  - [43] S. Hartmann and R. R. Gilbert, "Identifiability of material parameters in solid mechanics," *Archive of Applied Mechanics*, vol. 88, no. 1–2, pp. 3–26, 2018.
  - [44] V. Zocca, G. Spacagna, D. Slater, and P. Roelants, *Deep Learning. Uczenie Głębokie Z Językiem Python. Sztuczna Inteligencja I Sieci Neuronowe (English Title: Deep Learning with Python. Artificial Intelligence and Neural Networks)*, Helion, Gliwice, Poland, 2018.
  - [45] S. Hartmann, "A remark on material parameter identification using finite elements based on constitutive models of evolutionary-type," *Computer Assisted Methods in Engineering and Science*, vol. 24, no. 2, pp. 113–126, 2017.
  - [46] Research-and-Development Meeting with Industrial and Research Partners from Poland and Germany, Centre for Technology Transfer at Cracow University of Technology, Krakow, Poland, 2018.
  - [47] A. M. Stręk and M. Dudzik, "Quality measures in modelling of metal sponges," 2020.