ISSN 1330-3651 (Print), ISSN 1848-6339 (Online)
https://doi.org/10.17559/TV-20250131002314
Original scientific paper

The Impact of Different Artificial Neural Network Structures on the Results of Preprocessing 3D Digitization Data

Slađan LOVRIĆ*, Alan TOPČIĆ, Edin CERJAKOVIĆ

Abstract: Three-dimensional digitization (3D) is one way of collecting data in the formation of 3D models, for the development of new products, the reconstruction of existing ones, or general engineering practice. In order for the 3D model obtained by applying 3D digitization to be usable, the "raw" cloud of data generated by the 3D digitization process must be pre-processed before forming the 3D model. There are severel ways to preprocess data obtained from the 3D digitization process, and one of them is the application of artificial neural networks (ANN) in the process of preprocessing of 3D digitization data. The problems that arise when applying artificial neural networks (ANN) in the preprocessing process of 3D digitization data are in the structure of the ANN, which directly affects the preprocessing results. The aim of this paper is to highlight the impact of the structure of artificial neural networks on the preprocessing of 3D digitization data.

Keywords: artificial neural network (ANN); preprocessing; raw point cloud; three-dimensional digitization (3D)

1 INTRODUCTION

From the aspect of product development, production and placing products on the market, the goal of small, medium and large companies is to meet the requirements of end users, that is, product buyers. In general, the market demands imposed on enterprises are becoming increasingly higher, so manufacturers, at certain times, are unable to meet the set requirements, which relate to the quality of the product itself. The above imposes on economic entities the need to their existing production programs. Defined technological processes are redesigned by introducing newer ones, more modern, cheaper and faster development ways, particularly in product design.

One of the ways to completely or partially solve this problem, satisfy demands of market, is to introduce reverse engineering systems into existing engineering practices of product development and design. Initially, reverse engineering was limited to the replication of existing products or models, [1] but today reverse engineering is based on 3D digitization of existing physical objects, which avoids conventional classical modeling and enables the generation of 3D models of very complex geometric forms [2-4], as shown in Fig. 1.

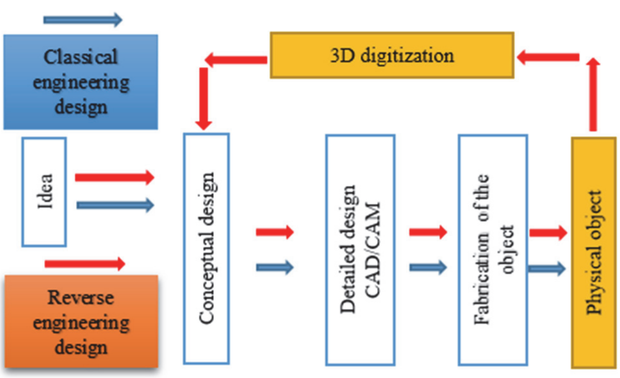

Figure 1 Schematic representation of classical and reverse engineering design

The application of reverse engineering in product development and design is related to the geometric accuracy of the digitized form - the directly collected points in 3D space, which in principle represent "raw" data that needs to be processed - pre-processed. Preprocessing of "raw" data of three-dimensional digitization is necessary in order to achieve the desired geometric accuracy of the designed product. According to some studies, 85% of the time in the reverse engineering process is spent on the phase of generating of surfaces model [5]. There are a number of methods which are applied during preprocessing generating of "raw" cloud of data obtained through the 3D digitization process, which belongs to one of the three preprocessing phases: impulse noise elimination, data smoothing, and reduction of the number of 3D digitized data, as shown in Fig. 2 [6].

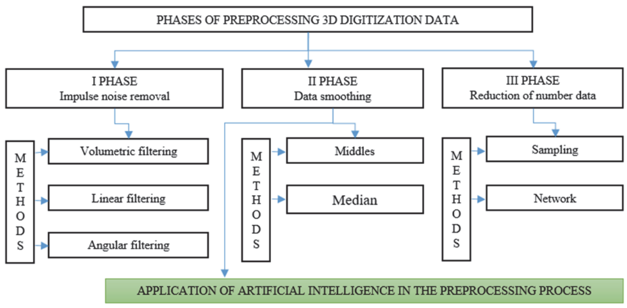

Figure 2 Presentation of methods and phases of preprocessing 3D digitization data

Taking into account that the application of the mentioned preprocessing methods for the 'raw' data cloud obtained from 3D digitization cannot always meet the established requirements regarding the accuracy of the generated data, there is a need to develop new ways and approaches for preprocessing data collected through the 3D digitization process. One possible approach is the development of a methodology based on the application of artificial intelligence tools in the preprocessing of the 'raw' data cloud obtained from 3D digitization. It is important to note that the field known today as artificial intelligence is the result of a combination of several distinct research areas: data processing, neurobiology, and physics, making it a typical example of an interdisciplinary field [7]. Although there are many other advanced non-algorithmic systems, such as genetic algorithms, adaptive memories, associative memories, fuzzy logic, etc., it has been shown that artificial neural networks are currently the most mature and widely applied approach for solving problems such as recognition, optimization, and control. They have also proven indispensable in situations where the ability to identify hidden connections and patterns is crucial for successful predictions [8]. The research which is conducted and presented in the paper relates to the possibility of applying artificial neural networks in the data preprocessing process of 3D digitization, and it has shown that the structure of the artificial neural network significantly affects the result of the mentioned process. It is important to note that as the object of 3D digitization, a spatial axis was taken, which was 3D digitized using a contact method, and the resulting 'raw' data cloud from this 3D digitization was analyzed in the experimental part of this paper, as shown in Fig. 3.

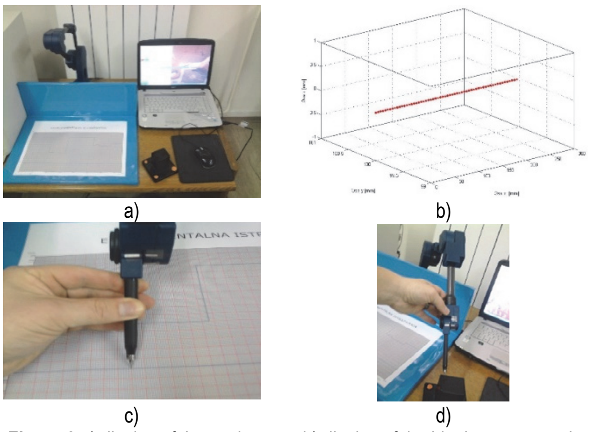

Figure 3 a) display of the workspace, b) display of the ideal programmed line", c) process of direct 3D digitization, d) display of the origin of the coordinate system of the 3D digitization device

In order to carry out a comparison of the 3D digitized preprocessed data cloud (line), it is necessary to define an 'ideal' data cloud of the same object, i.e., spatial axis. The ideal spatial axis, identical to the line being digitized, was programmed using the Matlab software package (Fig. 3b). All data obtained from the 3D digitization, describing the real spatial axis (Fig. 3c), were compared with the ideal programmed spatial axis, enabling the determination of deviations between the real 3D digitized and the ideal set of analyzed spatial axis data.

In line with the previously presented information, the following section of the paper showcases the impact of various structures of applied artificial neural networks on the preprocessing of the "raw" data cloud obtained from the 3D digitization of a spatial axis. It also indicates the deviations between the ideal programmed spatial axis and the real 3D digitized spatial axis, identifying the neural network structure that provides the best results - minimal deviations between the analyzed objects.

2 PREPROCESSING OF 3D DIGITIZATION DATA USING ARTIFICIAL NEURAL NETWORKS

After the process of 3D digitization of the real line in space - collecting the "raw" data cloud - has been completed, the preprocessing of the data is carried out using an artificial neural network with the aim of obtaining a usable CAD model and the final product. It is important to emphasize that in recent years, there has been a significant increase in the use of neural networks for solving complex, unstructured data [9]. The characteristic of good generalization in artificial neural networks is crucial, as it allows for accurate predictions on data sets that were not included in the training set after the network's training process on the appropriate input data set is completed.During the process of training the artificial neural network, in this particular case, an early stopping technique was employed, which required proper data preparation. This essentially involved splitting the initial data set into three subsets.

The first subset represents the training dataset, used to calculate gradients and update the weight coefficients and activation thresholds of artificial neural networks. The second subset is the validation dataset, which is used during training to validate the data. During neural network training, the mean squared error of the validation set is monitored. It typically decreases during the initial phase of training, just like the error of the training set. In the case of data overfitting, the validation error usually starts to increase, and when it reaches a certain threshold of increase over several iterations, the training process of the neural network is stopped. At this point, the weight coefficients and activation thresholds retain the values they had at the minimum validation error. The third subset is the test dataset, which is not used during the training process of the artificial neural network and primarily serves to verify the design of the artificial neural network.

The dataset was randomly divided from the initial dataset obtained through 3D digitization of a line, such that 60% of the data is used for training, 20% for validation, and 20% for testing [10]. Considering that the input dataset is of significant dimensions with a certain degree of correlation between components, the preprocessing included reducing the size of the input vector. The reduction process was performed using principal component analysis, which resulted in the normalization of the input set components, a reduction in the degree of correlation among components, and the arrangement of orthogonal components so that those with the highest degree of variation are prioritized. This process eliminates components that contribute less than a predefined proportion to the dataset's variation. After completing the preprocessing operations on the dataset subsets, the formed datasets were presented to an artificial neural network with a variable structure (different numbers of neurons per layer and varying numbers of layers). Various activation transfer functions were applied per layer (Purelin, Tansig, Logsig), as well as different methods for updating weight coefficients and activation thresholds of the considered Feed Forward Back Propagation neural networks. Additionally, various training functions, i.e., learning methods for the neural network, were tested, including the resilient backpropagation algorithm (TrainRP), Levenberg-Marquardt algorithm (TrainLM), Polak-Ribiere conjugate gradient algorithm (TrainCGP), Powell-Beale conjugate gradient algorithm (TrainCGB), quasi-Newton algorithm (TrainBFG), gradient descent algorithm (TrainGD), gradient descent with momentum (TrainGDM), one-step secant algorithm (TrainOSS), Bayesian regularization algorithm (TrainBR), and gradient descent adaptive learning algorithm (TrainGDA). The aim of this approach was to select an appropriate architecture for the artificial neural network and subsequently analyze, by varying its structure, the impact of structural changes on the deviations between the 3D digitization data of the real and theoretical line. The generated input/output datasets were presented to the selected artificial neural network, and its training was performed. Among the analyzed neural networks, the architecture of the network chosen for further analysis is shown in Fig. 4. The specified artificial neural network has one hidden layer with a variable number (m) of neurons (m = 2, 5, 10, 15, and in some cases 17 and 19 neurons). It is important to note that the number of neurons in the output layer remains constant and equals one in all training cases. The activation transfer functions differ by layer, with the hidden layer using the tansig function and the output layer using purelin. The first step involved training the artificial neural network with the architecture shown in Fig. 4 using different training functions to determine whether various training methods for the neural network impact the error of the preprocessed 3D digitization data. In the initial phase, seven different training sessions were conducted, each employing a distinct learning method for the neural network. To evaluate the influence of the neural network's structure on the training outcomes and determine the structure with the least deviation for presenting the analyzed datasets, training was performed with varying structures, i.e., different numbers of neurons in the hidden layer and using different training algorithms [10].

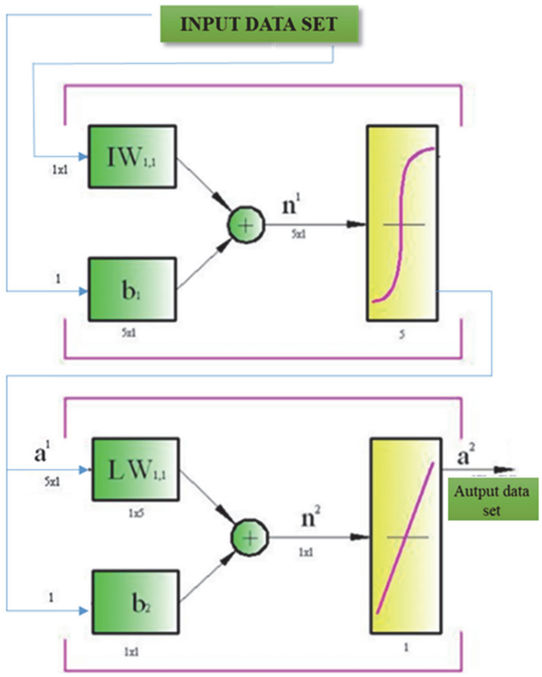

Figure 4 Architecture of a Feed-forward neural network with backpropagation, consisting of one hidden layer with 5 neurons and 1 neuron in the output layer [11]

3 RESULTS

The results of the artificial neural network training are shown in Fig. 5a, Fig. 5b, Fig. 5c, Fig. 5d, Fig. 5e, Fig. 5f, Fig. 5g.

From Fig. 5, it can be seen that the absolute error of the preprocessed line data in relation to the ideal line along the x, y, and z axes was taken as the accuracy parameter for the structure of the artificial neural network. The square in the figure represents the largest absolute error of the data-line obtained after training the neural network in relation to the target values along the z-axis. The circle in the figure represents the largest absolute error of the data obtained after training the neural network in relation to the target values along the y-axis, while the triangle in the figure represents the largest absolute error of the data obtained after training the neural network in relation to the target values along the x-axis.

Analysing Fig. 5, it can be seen that the artificial neural network structures shown in Fig. 5b (Train BR) and Fig. 5d (Train LM) provide the best results in terms of absolute error. Therefore, these two neural network structures will be further analyzed in the continuation of the work. Specifically, during the preprocessing of the data cloud describing the line, Feedforward Backprop neural networks with Levenberg-Marquardt and Bayesian Regularization training algorithms will be applied.

The results of the training of the selected artificial neural network structures have been organized into a series of cases, formed according to the training algorithm used, the number of neurons, and the activation transfer functions by layer. Each of the considered cases is represented by a series of diagrams and output data, shown in Fig. 6a, Fig. 6b, Fig. 6c, and Fig. 6d. The diagrams represent the dynamics of the conducted training, the comparison of the BPNN training results and target output values, the linear regression analysis of the BPNN training results and target output values, and the comparison of the results of the newly presented BPNN data after training and the target output values. According to the previously mentioned, the first case considered is the Feed-Forward Back Prop artificial neural network, training algorithm - Train LM (Levenberg Marquardt), neural network learning algorithm - Learngdm, number of layers in the neural network - two layers, one hidden layer with a variable number of neurons 10, 15, 17, and 19, output layer with one neuron, activation transfer functions by layers. Tansig and Purelin, number of iterations 3000 iterations. The results of this training are shown in Fig. 6a, Fig. 6b, Fig. 6c, Fig. 6d [10].

Fig. 6a, Fig. 6b, Fig. 6c, Fig. 6d show the variation/change in the squared error in relation to the epoch for training, validation, and testing. An epoch in this case represents the time period that passes for a maximum of 3000 training iterations of the specifically analyzed artificial neural network. It is important to emphasize that the training of the artificial neural network can be stopped before the "end" of the epoch, i.e., when the performance of the network on the validation set is at its best, meaning the training squared error is the smallest.

The second case that was analyzed in detail is the Feed-Forward Back Prop artificial neural network, training algorithm - TrainBR (Bayesian Regulation), neural network learning algorithm - Learngdm. The number of layers in the neural network consists of two layers: one hidden layer with a variable number of neurons (2, 5, 10, and 15) and an output layer with a single neuron. The activation transfer functions per layer are Tansig and Purelin. The number of iterations is 3000. The training results are shown in Fig. 7a, Fig. 7b, Fig. 7c, Fig. 7d.

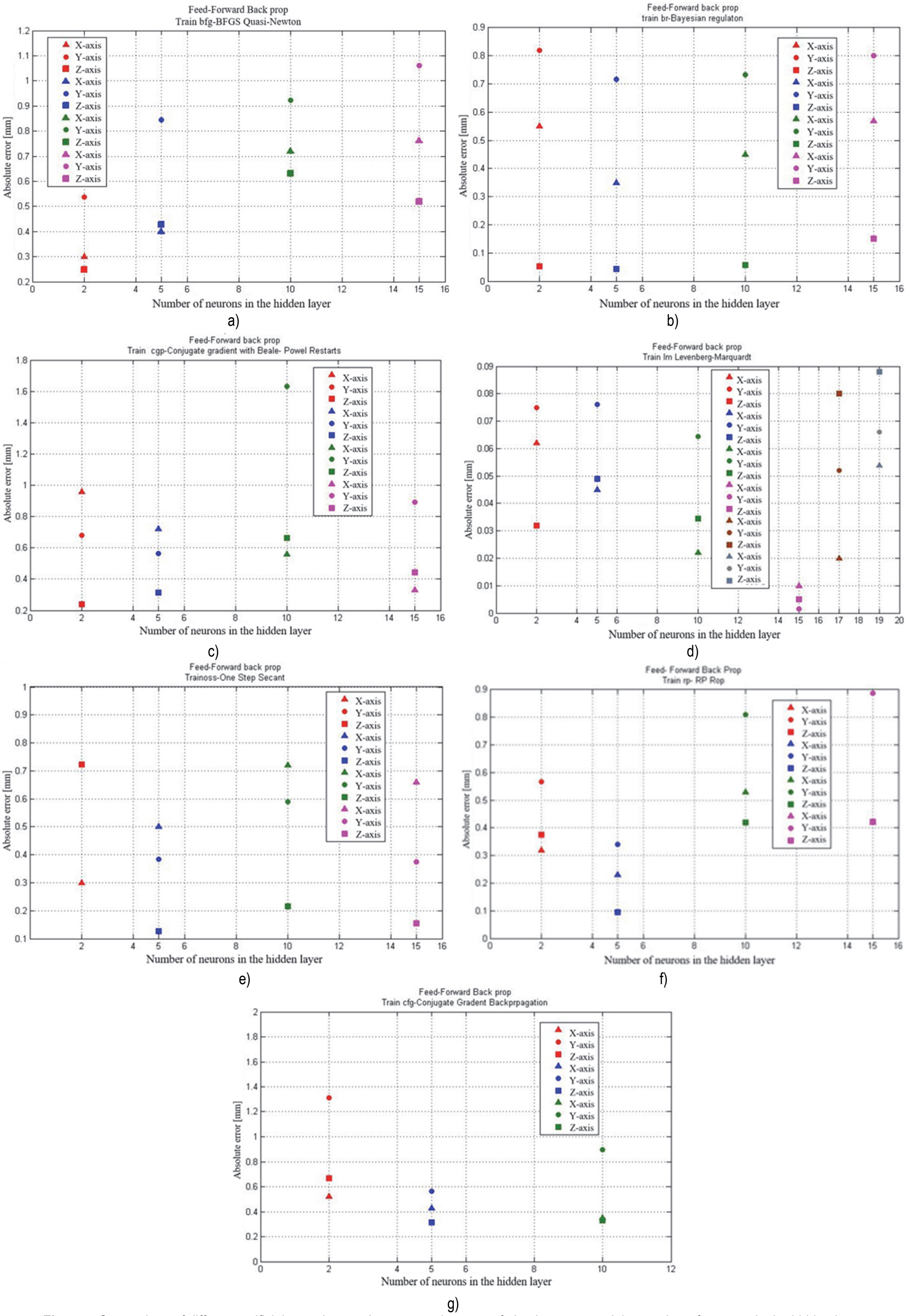

Figure 5 Comparison of different artificial neural network structures in terms of absolute error and the number of neurons in the hidden layer

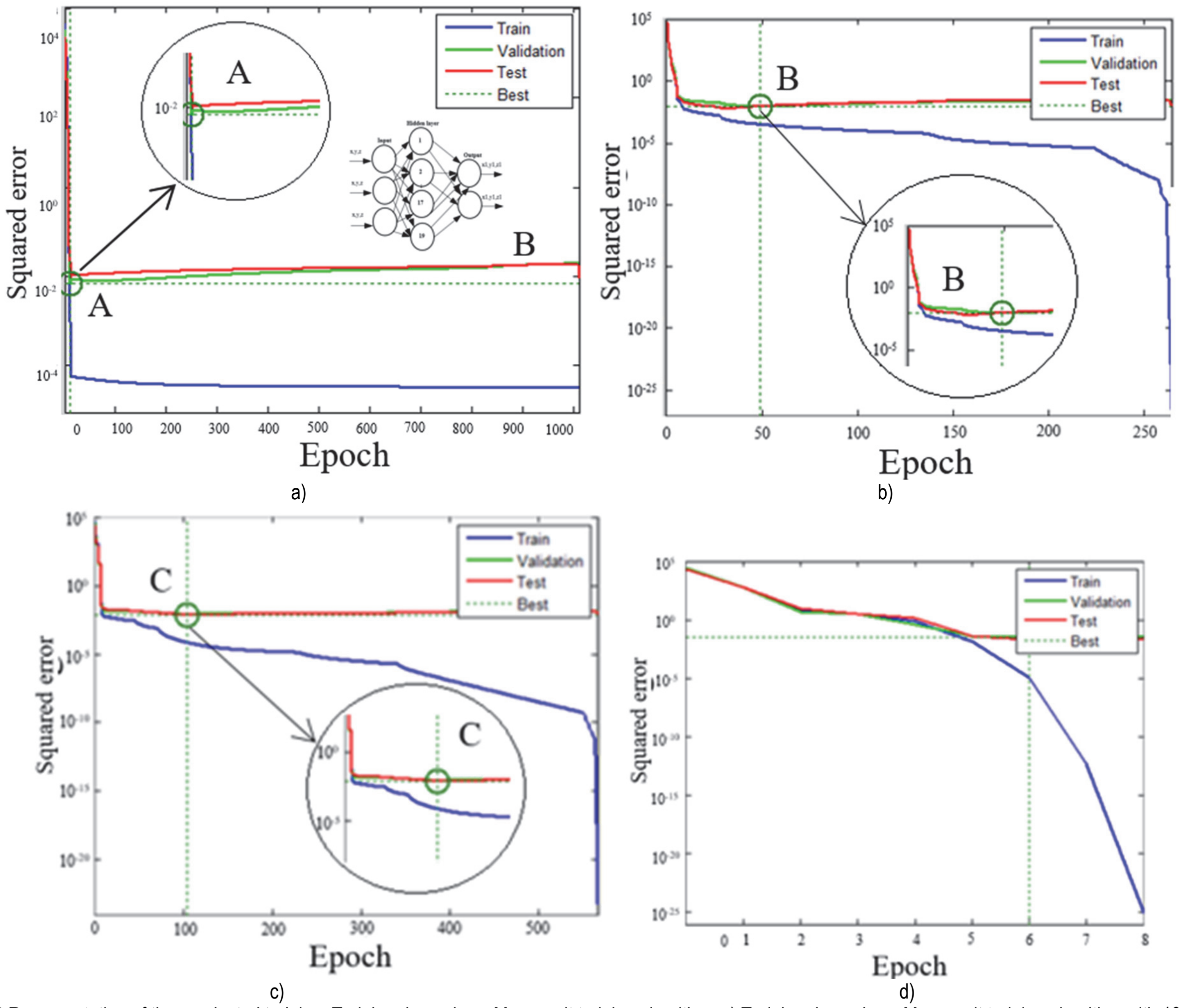

Figure 6 Representation of the conducted training, Trainlm - Levenberg-Marquardt training algorithm: a) Trainlm - Levenberg-Marquardt training algorithm with 10 neurons in the hidden layer, b) Trainlm - Levenberg-Marquardt training algorithm with 15 neurons in the hidden layer, c) Trainlm - Levenberg-Marquardt training algorithm with 17 neurons in the hidden layer, d) Trainlm - Levenberg-Marquardt training algorithm with 19 neurons in the hidden layer

In order to determine the impact of different structures and training methods of the Feed-Forward Back Propagation artificial neural network on the squared error between the output data obtained after training and the target values expressed through the corresponding x, y, z coordinates, the previously presented data in Fig. 6 and Fig. 7 were analyzed in detail, and the results are shown in Fig. 8a, Fig. 8b.

From Fig. 8a, it can be seen that in the first case, the trained artificial neural network with the Feed-Forward Back Propagation architecture and the Levenberg-Marquardt training algorithm had its structure changed by modifying the number of neurons in the hidden layer (thus, the artificial neural network was trained with 10, 15, 17, and 19 neurons in the hidden layer). The training of the mentioned artificial neural network in all four cases was stopped due to the occurrence of the "Over Learning" problem at the point when the squared error between the training, validation, and test data was minimal. After training, the obtained results are shown in Fig. 8a:

- Number of neurons in the hidden layer: 19, squared error 0.0223 (mm) at the 8th iteration out of a possible 3000 iterations in the set epoch;
- Number of neurons in the hidden layer: 10, squared error 0.0354 (mm) at the 20th iteration out of a possible 3000 iterations;
- Number of neurons in the hidden layer: 15, squared error 0.0100 (mm) at the 50th iteration out of a possible 3000 iterations;
- Number of neurons in the hidden layer: 17, squared error 0.0125 (mm) at the 100th iteration out of a possible 3000 iterations.

Fig. 8b shows an FFBP artificial neural network with the Bayesian Regulation training algorithm (the relationship between the number of iterations - squared error [mm] and the number of neurons in the hidden layer). The structure of the neural network, as in the previous case, was modified by changing the number of neurons in the hidden layer (2, 5, 10, and 15 neurons in the hidden layer). Unlike the previous case, in this case, due to the occurrence of the Over Learning problem, training was stopped for 2 and 5 neurons in the hidden layer (Fig. 7a and Fig. 7b), while for 10 and 15 neurons in the hidden layer, the network was trained for the entire defined epoch, i.e., 3000 iterations. After training, the results obtained are shown in Fig. 8b.

- Number of neurons in the hidden layer: 2, squared error 0.1000 [mm] in the 75th iteration out of a possible 3000 in the defined epoch;
- Number of neurons in the hidden layer: 5, squared error 0.1995 [mm] in the 2200th iteration out of a possible 3000;

- Number of neurons in the hidden layer: 10, squared error 0.2238 [mm] in the 3000th iteration out of a possible 3000;
- Number of neurons in the hidden layer: 15, squared error 0.2041 [mm] in the 3000th iteration out of a possible 3000.

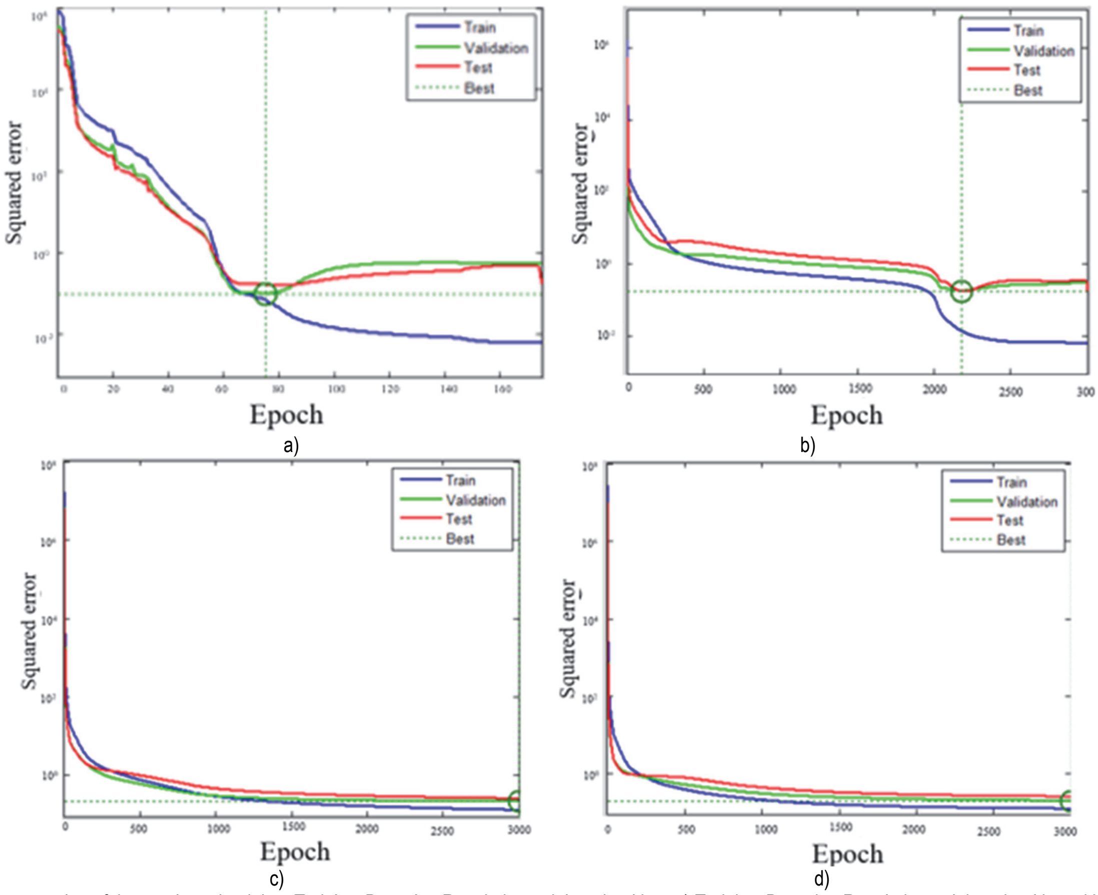

Figure 7 Representation of the conducted training, Trainbr - Bayesian Regulation training algorithm: a) Trainbr - Bayesian Regulation training algorithm with 2 neurons in the hidden layer, b) Trainbr - Bayesian Regulation training algorithm with 5 neurons in the hidden layer, c) Trainbr - Bayesian Regulation training algorithm with 10 neurons in the hidden layer, d) Trainbr - Bayesian Regulation training algorithm with 15 neurons in the hidden layer

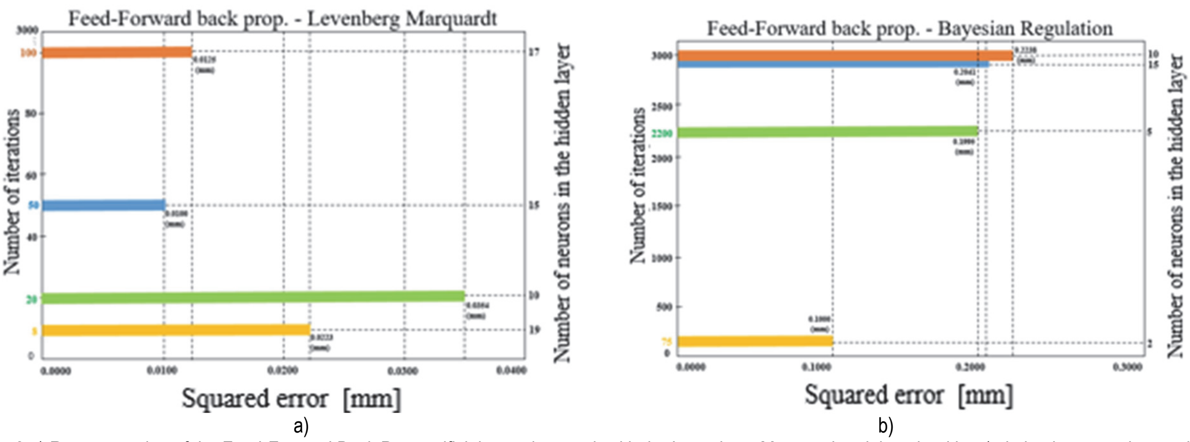

Figure 8 a) Representation of the Feed-Forward Back Prop artificial neural network with the Levenberg-Marquardt training algorithm (relation between the number of iterations - squared error [mm] and the number of neurons in the hidden layer), b) Representation of the Feed-Forward Back Prop artificial neural network with the Bayesian Regulation training algorithm (relation between the number of iterations - squared error [mm] and the number of neurons in the hidden layer)

According to the above, it can be seen that the different structure of the Feed Forward Back Prop. artificial neural network with the Levenberg Marquardt training algorithm and the Bayesian Regulation training algorithm, along with the varying number of neurons in the hidden layer, affects the squared error between the output data obtained after training and the target values of the data expressed through the *x*, *y*, *z* coordinates. A graphical representation of the change in squared error is shown in Fig. 9.

Fig. 9a shows a comparison of the target values of data describing the line and the data obtained from 3D digitization of the line, after being preprocessed using different structures of the artificial neural network. It is important to note that the applied artificial neural network architecture, as shown in Fig. 4, with the Levenberg Marquardt training algorithm, was used in each of the training variants, while the structure of the artificial neural network is modified by the number of neurons in the hidden layer. The artificial neural network was trained with 10, 15, 17, and 19 neurons in the hidden layer, and as shown in Fig. 9a, the structure of the artificial neural network with 15 neurons in the hidden layer gives the least deviation between the compared parameters.

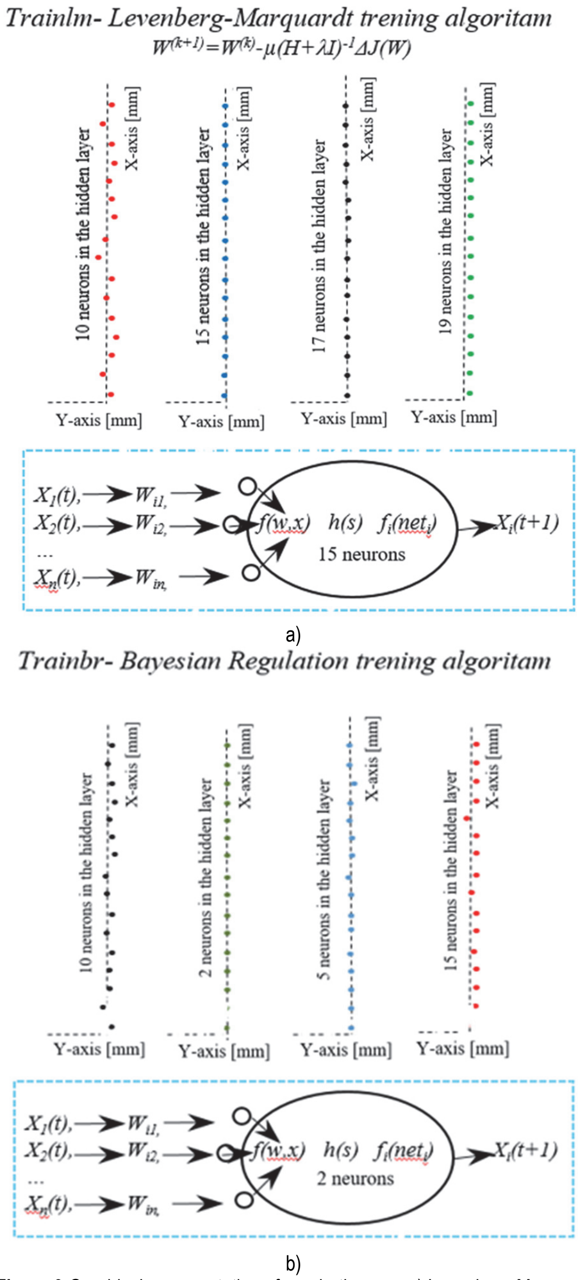

Figure 9 Graphical representation of quadratic error, a) Levenberg-Marquard training algorithm, b) Bayesian Regulation training algorithm

Fig. 9b shows a comparison of the data as in Fig. 9a, but with the artificial neural network architecture, as shown in Fig. 4, using the Bayesian Regulation training algorithm in each of the training variants, while the structure of the artificial neural network is modified by the number of neurons in the hidden layer. The artificial neural network was trained with 2, 5, 10, and 15 neurons in the hidden layer, and as shown in Fig. 9b, the structure of the artificial neural network with 2 neurons in the hidden layer gives the least deviation between the compared parameters. It is important to note that the Feed-forward neural network (Levenberg Marquardt and Bayesian Regulation training algorithms) is designed so that the functionality of the neurons in the hidden layer provides the network with the ability to represent knowledge based on three constituent elements: the input operator $f(W, X)$, which combines the input and weight relationships of the interconnections W and forms a unique value s (i.e., $s = f(W, X) = WT^*X$), which is then prepared for the transfer function; the transfer function $h(s)$, which processes the output from the neuron's input operator (performs integration), forming the required value for the activation function; the activation function φ(neti), which processes the output of the transfer function and controls the output value of the neuron in the output layer [10].

4 CONCLUSION

The process of three-dimensional (3D) digitization is based on both contact and non-contact methods. Errors arise in the 3D digitization process for various reasons, and it is necessary to preprocess the "raw" point cloud data to minimize errors and make the point cloud usable. One way in which the "raw" point cloud data obtained from the 3D digitization process can be preprocessed is by applying a Feed-forward back propagation artificial neural network in the preprocessing process. The error that occurs in the 3D digitization process between the real - "raw" (3D digitized) and ideal (programmed) point cloud data directly affects the generation of the CAD model and ultimately the creation of the prototype and the final product. Accordingly, this paper demonstrates how the different structure of the feed-forward back propagation artificial neural network applied to process the real - "raw" point cloud data obtained by contact-based 3D digitization affects the error between the real - "raw" (3D digitized) and ideal (programmed) point cloud data. The artificial neural network treated in this paper is Feed-forward back propagation. The architecture of the artificial neural network is: training algorithms - Trainlm - Levenberg Marquardt and Trainbr - Bayesian Regulation, the learning algorithm of the neural network - Learngdm, number of layers in the network - two layers, one hidden and one output layer with activation transfer functions per layer - Tansig and Purelin. The structure of the artificial neural network was changed by varying the training of the artificial neural network with respect to the training algorithm and the number of neurons in the hidden layer. Analyzing Fig. 8a and Fig. 8b, it can be concluded that the treated artificial neural network with a variable structure in terms of the training algorithm (Levenberg Marquardt and Bayesian Regulation) yields different results for the output data. It can be seen that the Levenberg Marquardt training algorithm gives better results when observing the squared error compared to the Bayesian Regulation training algorithm. Furthermore, it can be concluded that the number of neurons in the hidden layer also affects the output parameters, in this case, the squared error between the target (ideal) values describing the line and the data obtained from 3D digitization of the line, shown in Fig. 9a and Fig. 9b. In this paper, it has been shown that the structure of the treated Feed-forward back propagation artificial neural network affects the result of preprocessing 3D digitization data. It is important to emphasize that an adequate structure of the applied artificial neural network in the preprocessing of 3D digitization data will certainly lead to a reduction in the mean squared error between the real - 3D digitized and ideal (programmed) data. This research undoubtedly opens the way for new investigations related to other types and architectures of artificial neural networks, as well as to non-contact methods of 3D digitization.

5 LITERATURE

[1] Bernardini, F., Bajaj, C. L., Chen, J., & Schikore, D. R. (1997). *Automatic Reconstruction of 3DCAD Models from Digital Scans*. Department of Computer Sciences, Purdue University West Lafayette, USA.

[2] Balokhonov, R., Utyaganova, V., Gatiyatullina, D., Zemlianov, A., & Romanova, V. (2024). Interlayer effect on deformation and fracture of dendritic structure formed during wire-feed electron-beam additive manufacturing of Al-Si alloy. *Facta Universitatis, Series: Mechanical Engineering, 22*(3, SI), 515–527. https://doi.org/10.22190/FUME231225016M

[3] Sidhu, S. S. (2025). Development of scaffolds using 3D printing for tissue engineering. *Journal of Management and Engineering Sciences, 2*(1), 22–28. https://doi.org/10.61552/JMES.2025.01.002

[4] Shete, H., & Chavan, V. (2023). Development of fuel filter cap of Comet diesel engine using structural optimization technique and additive manufacturing. *Engineering Review*, 43(2), 48–56. https://doi.org/10.30765/er.2115

[5] Lee, K. H., Woo, H., & Suk, T. (2001). Point Data Reduction Using 3D Grids. *The International Journal of Advanced Manufacturing Technology*, 18(3), 201-210. https://doi.org/10.1007/s001700170075

[6] Lovrić, S., Topčić, A., & Cerjaković, E. (2024). Contribution of artificial neuron network in the development of reverse engineering systems. *Proceedings of Artificial Intelligence in Bosnia and Herzegovina / research, application and development perspectives*, 375-388.

[7] Lovrić, S., Topčić, A., & Cerjaković, E. (2022). Artificial intelligence as to perspectively tool for pre-processing of results 3D-digitization. *Proceedings of Artificial intelligence in Bosnia and Herzegovina / research, application and development perspectives*, 105-119.

[8] Topčić, A. (2006). *Optimization of the Operation of Transfer Segments in the Internal Transport System.* Doctoral Dissertation, University of Tuzla, Faculty of Mechanical Engineering in Tuzla, Bosnia and Herzegovina.

[9] Osmanović, A., Omerćehajić, T., Trakić, E., & Ćosić, S. (2022). VoiceBus - application of voice technology in passenger transport systems. *Artificial intelligence in Bosnia and Herzegovina / research, application and development perspectives/*, 92-104.

[10] Lovrić, S. (2015). *Development of methodology for the optimization of process parameters of 3D digitizing of complex geometric forms*. Doctoral dissertation, University "Džemal Bijedić" in Mostar, Faculty of Mechanical Engineering.

[11] Topčić, A., Cerjaković, E., & Herić, M. (2013). Simulation of Reloading Segments of Internal Transportation Systems by Artificial Neural Networks. *Journal of Production Engineering*, *16*, 51-54.

Contact information:

Slađan LOVRIĆ, PhD, Associate Professor
(Corresponding author)
Department of Production Engineering,
Faculty of Mechanical Engineering,
University of Tuzla,
75000, Tuzla, Bosnia and Herzegovina
E-mail: sladjan.lovric@untz.ba

Alan TOPČIĆ, PhD, Professor
Department of Production Engineering,
Faculty of Mechanical Engineering,
University of Tuzla,
75000, Tuzla, Bosnia and Herzegovina
E-mail: alan.topcic@untz.ba

Edin CERJAKOVIĆ, PhD, Associate Professor
Department of Production Engineering,
Faculty of Mechanical Engineering,
University of Tuzla,
75000, Tuzla, Bosnia and Herzegovina
E-mail: edin.cerjakovic@untz.ba