Proceedings in MANUFACTURING SYSTEMS

Proceedings in Manufacturing Systems, Volume 17, Issue 1, 2022, 23-30 ISSN 2067-9238

PREDICTION OF SURFACE ROUGHNESS IN MILLING BASED ON ACOUSTIC SIGNALS USING DIFFERENT TYPES OF INTELLIGENT ALGORITHMS

Goran MUNĐAR[1,*], Uroš ŽUPERL[2]

[1)] MSc Eng., Researcher, Laboratory for Mechatronics, Faculty of Mechanical Engineering, University of Maribor, Maribor, Slovenia
[2)] Assoc. Prof., Researcher, Laboratory for Mechatronics, Faculty of Mechanical Engineering, University of Maribor, Maribor, Slovenia

Abstract: *Surface roughness is playing very important role in the performance of finished part. Roughness measurements are typically done off-line after the part has already been machined, but recently the focus has changed to online monitoring. Through advancements in the fields of computers and sensors, it is now possible to measure and control the machining processes. With the advancement of artificial intelligence and intelligent algorithms new system can be build, that can describe complex, non-linear, multi-variant machining processes. The main focus of this paper is to develop three different prediction models for predicting surface roughness of machined parts during milling. Prediction models will be able to predict surface roughness based on four inputs: feed rate, spindle speed, depth of cut and vibration. For the development of different intelligent prediction models MATLAB software tool will be used. Prediction models will be based on artificial neural networks (ANN) and adaptive neuro-fuzzy inference system (ANFIS). The results of developed prediction models will be compared to statistical regression analysis results and to experimental results.*

Key words: *Machining, Milling process, Surface roughness, Artificial neural networks, ANFIS, Online monitoring system.*

1. INTRODUCTION

In recent years, due to increasing international and domestic competition, most manufacturers are increasingly focusing on the implementation of automation and flexible production as a means to increase productivity and improve the quality of finished products. Numerically controlled (NC) and computer numerically controlled (CNC) machined machines have been increasingly implemented in recent years to achieve complete automation of machining. NC and CNC machine tools require less operator attention and input during execution improve productivity and increase the quality of the workpiece surface [1].

Among many NC or CNC industrial machining processes, milling is one of the basic machining operations. End milling is the most common metal removal operation. It is widely used in a variety of manufacturing industries, including the aerospace and automotive sectors, where quality is an important factor in the production of slots, pockets and precision molds. The quality of the treated surface plays a particularly important role in milling itself, as a quality treated surface significantly improves the properties of the material such as fatigue strength, corrosion resistance and creep [2]. Surface roughness also affects some other properties, such as the contact surface, which causes surface friction, wear, light reflection, heat transfer, and the ability to distribute and retain lubricant. Therefore, before manufacturing, the desired quality (roughness) of the final surface is usually determined and based on this, appropriate procedures are selected to achieve the required quality.

The machining processes of milling are basically complex, non-linear, multi-variant and often exposed to various unknown external disturbances. The machining process is usually performed by a qualified operator that uses intuition-based decision-making methods and rules derived from experience. This process is usually not precise enough and in many cases product defects occur. For this reason, and for the realization of highly productive and flexible machining, a reliable, automated machining system with intelligent functions is required, which can also be called intelligent machining and represents Industry 4.0 [1, 2]. For the manufacture of such systems, it is necessary to implement appropriate methods of measuring the surface of the workpiece or wear of the machining tool, which will be able to monitor the operation of the machining process and find defects that occur on the machine (such as worn or broken tools and inadequate surface quality) and try to improve the process without some additional input from the machine operator. Recently, intelligent algorithms such as neural networks, genetic algorithms, particle swarm theories, etc. have been increasingly used to make such systems, which can be used to determine the surface roughness with sufficient accuracy based on input data from the machining system. As part of this paper, we developed three different models using artificial neural networks

[*] Corresponding author: Goran Munđar, Smetanova ulica 17, 2000 Maribor, Slovenia,
Tel.: +38622207564,
E-mail addresses: *goran.mundjar@um.si* (G. Munđar), *uros.zuperl@um.si* (U. Župerl).

and ANFIS that will be able to accurately predict the roughness of machined surface during the processing in milling.

2. APPLICATION FIELD

2.1. Surface roughness definition

The final surface roughness obtained during machining can be considered as the sum of two independent influences. The first impact is the ideal or geometric finish, which is the result of the geometry and kinematic movements of the machining tool. The ideal finish can be calculated from the feed rate per tooth, the tool nose radius and the tool lead angle. Another impact is the natural finish resulting from tool wear, vibration, machine movement errors and the effects of the workpiece itself, such as inhomogeneity, edge formation and breakage at low cutting speeds [3].

For the purposes of quantitative comparison and analysis, it is useful to express the roughness of the treated surface with a single factor or index. Typically, a profile can be described using two sets of parameters: wavelength and surface height [4]. Wavelength parameters include wavelength and surface slope. The height of the surface can be described by the parameters *Ra*, *Rq*, *Rz*, etc. *Ra* represents the arithmetic mean roughness. *Rq* represents the mean square roughness. *Rz*, on the other hand, represents the highest height of roughness from the highest peak to the lowest valley. Since *Ra* and *Rq* are the two most frequently used parameters, we will use *Ra* in our paper to express the surface roughness of the treated surface.

Ra can be calculated from the following equation:

$$R_a = \frac{1}{L}\int_0^L |y(x)|\,dx, \qquad (1)$$

where *Ra* is the arithmetic mean deviation from the mean line and *L* is the sampling length, *y* is the ordinate of the profile curve.

2.2. Methods for measuring surface roughness

Methods based on two different principles have been developed to control surface roughness: the contact principle and the non-contact principle. Contact-based instruments, commonly used to measure the roughness using so-called "styluses", are the most common in practice. Instruments operating in non-contact mode are based on the principles of optical interferometry, displacement, vibration, sound reflection or electron beams. It is very important that measuring instruments for measuring surface roughness are as accurate, reliable, inexpensive, fast and non-destructive as possible.

In most industrial environments, surface roughness inspection and assessment is performed on the basis of off-line methods using a touch probes or on the basis of on-line methods by operators.

Off-line measurement usually requires the removal of the workpiece from the machine, its cleaning and testing with an off-line instrument to measure surface roughness. The disadvantage of this process is that it is time consuming and uneconomical, as after the measurement, the machine and the workpiece need to be readjusted and prepared for further processing.

On-line measurements also require interruption of machining and cleaning of the workpiece before measurement. Even if portable instruments that do not require changing the machine settings have been used to measure the workpiece, the machine must still be stopped and the workpiece cleaned before the measurement itself. If the workpiece does not meet the specifications, the workpiece or the entire batch may be discarded or recovered. Due to the aforementioned disadvantages, the above methods are not suitable and flexible for real-time process control and full automation. Therefore, to solve these problems, so-called "in-process" methods are desirable, which are able to measure the roughness of the treated surface during the treatment itself.

In recent years, intelligent algorithms (such as neural networks, ANFIS, genetic algorithms, etc.) have been increasingly used to determine the roughness of the machined surface, which are able to determine the quality of the treated surface from signals obtained from sensors [5, 6].

The use of cutting parameters and other properties such as workpiece hardness, tool geometry, cutting time and acceleration as inputs to neural networks to predict surface roughness gave very accurate results [7, 8]. Benaros et al. [9] was able to predict the roughness of the machined surface during CNC milling using neural networks and the Taguchi method. The results of the experiment showed that the feed rate, the vertical component of the cutting force Fx, the depth of cut and the use of coolant are the parameters that most affect the quality of the surface roughness. Risbood et al. [10] was able to determine the roughness of the machined surface using neural networks by measuring cutting force and vibration.

They proved that the use of neural networks could model surface roughness and vibration in machining processes. Huang et al. [11] presented in his work the use of neural networks in combination with the Poka-yoke method for predicting surface roughness in end-milling. In their study, cutting parameters and cutting speed were used as input values.

The use of intelligent algorithms in optimizing cutting conditions has high accuracy. Predicting surface roughness in milling using genetic programming has shown that roughness is particularly sensitive to feed rate. Brezocnik et al. [12] in his work demonstrated the use of genetic programming to construct a model that included three cutting parameters, in addition to which vibrations provided high accuracy for predicting surface roughness. Oktem et al. [13] demonstrated that the link between genetic programming and response surface methodology has improved cutting conditions such as feed rate, cutting speed and axial depth of cut. They managed to optimize the quality of the treated surface area by up to 10%.

Soft logic has also established itself as an accurate method for predicting surface roughness in machining processes. The use of an adaptive neural network-based fuzzy inference system (ANFIS) to predict surface roughness in milling yielded very small errors between predicted and measured values (approximately 4%) [14, 15].

3. METHODS USED

3.1. Adaptive neuro-fuzzy inference system

Adaptive neuro-fuzzy inference system (ANFIS) is an integration system that uses neural networks to optimize a soft decision-making system. ANFIS consists of a series of soft "if-then" rules with corresponding membership functions to generate specific input-output pairs. The initial soft rules and membership functions are first determined using human expertise on the results to be modeled. ANFIS can then change these soft "if-then" membership rules and functions to reduce the output error rate or explain the I/O relationship of a complex system [16]. Two soft if-then rules below describe the ANFIS architecture:

- If (x is A_1) and (y is B_1) then ($z_1 = p_1x + q_1y + r_1$),
- If (x is A_2) and (y is B_2) then ($z_2 = p_2x + q_2y + r_2$),

where x and y are inputs, A_i and B_i are soft sets, z_i (i = 1, 2) are outputs within the soft range defined by soft rules, and p_i, q_i, and r_i are parameters defined during the training process. To implement these two rules, the ANFIS architecture has five layers as shown on Fig. 1.

- Layer 1: Input membership function

The first layer is used to convert the input numeric values to soft values, and all nodes of the first layer are flexible. The outputs from the first layer are membership estimates of input values, which can be represented by the equations 2 and 3:

$$o_i^1 = u_{A_i}(x),\ i = 1, 2, \tag{2}$$

$$o_i^1 = u_{B_{i-2}}(y),\ i = 3, 4, \tag{3}$$

where $u_{A_i}(x)$ and $u_{B_{i-2}}(y)$ represent soft membership functions.

- Layer 2: Rules

The nodes of the second layer are fixed nodes. They are denoted by M and they act as multipliers. The outputs of the second layer represent the soft intensities (ω_i) of each rule and can be expressed by the equation (4):

$$o_i^2 = \omega_i = u_{A_i}(x)u_{B_i}(y),\ i = 1, 2. \tag{4}$$

- Layer 3: Normalization

In the third layer, the nodes are also fixed. They are denoted by N and they play the role of normalizing the soft intensities from the previous layer. The normalization factor is calculated as the sum of the weighting functions. The outputs of this layer are called normalized soft intensities and are calculated by the Eq. (5).

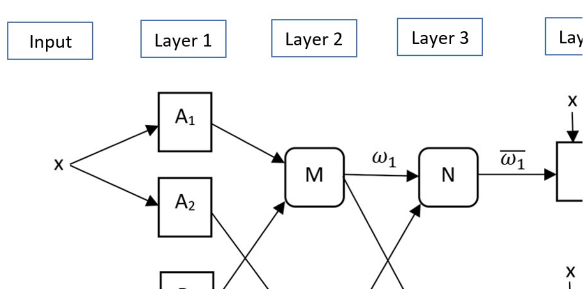

Fig. 1. ANFIS architecture.

$$o_i^3 = \overline{\omega_i} = \frac{\omega_i}{\sum_{i=1}^{2}\omega_i},\ i = 1, 2. \tag{5}$$

- Layer 4: Output membership function

The nodes in the fourth layer are flexible. The outputs of the fourth layer can be calculated using the Eq. (6):

$$o_i^4 = \overline{\omega}_i z_i = \overline{\omega_i}(p_i x + q_i y + r_i),\ i = 1, 2. \tag{6}$$

- Layer 5: Output

The fifth layer consists of only one fixed node, denoted by S. This node calculates the sum of the input signals. The total output of the fifth layer can be expressed by the following equation:

$$o_i^5 = z = \sum_{i=1}^{2}\overline{\omega}_i z_i = \frac{\sum_{i=1}^{2}\omega_i z_i}{\sum_{i-1}^{2}\omega_i},\ i = 1, 2. \tag{7}$$

After making the ANFIS model, his training follows. In the ANFIS training process, a hybrid learning algorithm is used to adjust the parameters of membership functions, which works on the principle of gradient descent and least squares estimation. The outputs of the nodes move forward until the layer of the output membership function and consequently also the parameters are identified by the least squares estimate.

3.2. Artificial neural network

Neural networks are made up of simple elements whose action is inspired by biological nervous systems. The operation of the system of these elements largely depends on the connections between the elements. Neural networks can be trained to perform certain functions by adjusting the values of connections (weights) between elements.

Artificial neural networks (NM) are made up of artificial neurons. Artificial neurons try to mimic the functioning of natural neurons found in the nervous systems of animals and are the basic cell of the nervous system. The artificial neuron works by multiplying the inputs with the appropriate weights and then summing the obtained products and comparing them with the threshold over the threshold function.

Artificial neural networks consist of several layers or. levels of neurons. Neurons can be located in one or more hidden layers and one output layer. Such neural networks can approximate any nonlinear function, and only when initial weights are chosen to prevent trapping in local minima. Multilayer NMs are composed in such a way that the output signals of one layer enter as input signals into the next layer of neurons. An example of a two-layer NM structure is shown in Fig. 2.

The output of the NM can be calculated with Eqs. (8) and (9).

$$o_{Jj} = g\left(\sum_{i=1}^{n} w_{Jij}\cdot o_{Ii}\right),\ j = 1 \cdots m, o_{Ii} = i_i, \tag{8}$$

$$o_{Lr} = g\left(\sum_{j=1}^{m} w_{Ljr}\cdot o_{Jj}\right),\ r = 1 \cdots l. \tag{9}$$

The process of learning multilayer NM is done using the back-propagation rule (BPG).

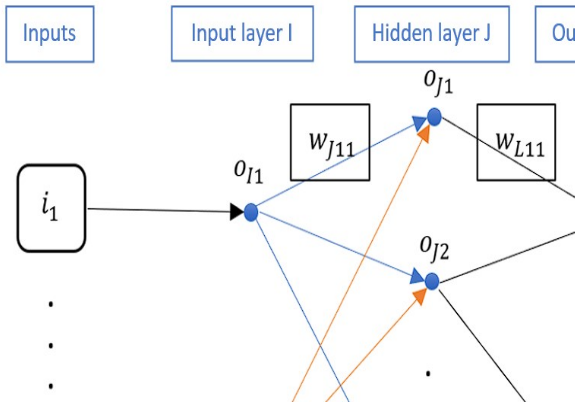

Fig. 2. Two-layer ANN structure.

The deviation E_p is calculated, which is equal to the sum of the squares of the difference of the p-th desired response vector t_r^p on the p-th input sample and the output vector of the L-th level of the neural network o_{Lr}^p for $r = 1\ldots l$ on the same p-th input pattern.

The deviation can be calculated by the following equation:

$$E_p = \frac{1}{2}\sum_{r=1}^{l}\left(t_r^p - o_{Lr}^p\right)^2. \tag{10}$$

The NM weight adjustment process is carried out until the deviation E_p over all input samples p is minimized.

Calculating weight change during NM learning is calculated in two different ways for the output and hidden layer of neurons.

Change in the weights of the output layer of neurons is calculated using the Eq. (11).

$$\Delta w_{Ljr}^p = \varepsilon_L \cdot \delta_{Lr}^p \cdot o_{Jj}^p,\; j = 1 \cdots m, r = 1 \cdots l, \tag{11}$$

where δ_{Lr}^p can be calculated by Eq. (12).

$$\delta_{Lr}^p = \left(t_r^p - o_{Lr}^p\right) \cdot o_{Lr}^p \cdot \left(1 - o_{Lr}^p\right). \tag{12}$$

The weights of the output level of the neural network are then calculated using the Eq. (13).

$$w_{Ljr}^p = \Delta w_{Ljr}^p + w_{Ljr}^{p-1}. \tag{13}$$

The change in the weights of the hidden layer of neurons is calculated by Eq. (14).

$$\Delta w_{Jij}^p = \varepsilon_J \cdot \delta_{Ji}^p \cdot o_{Ii}^p;\; i = 1 \cdots n, j = 1 \cdots m, o_{Ii} = i_i, \tag{14}$$

where δ_{Jj}^p is calculated using the following equation:

$$\delta_{Ji}^p = o_{Ii}^p \cdot \left(1 - o_{Ii}^p\right) \cdot \sum_{r=1}^{l} \delta_{Lr}^p \cdot w_{Ljr}^p. \tag{15}$$

The weights of the hidden level of the neural network are calculated using the following Eq. (16).

$$w_{Jij}^p = \Delta w_{Jij}^p + w_{Jij}^{p-1}. \tag{16}$$

In the above equations, ε_L and ε_J are learning constants whose values are set according to a specific case.

3.3. Statistical regression model

Regression analysis is a statistical method that checks the influence of independent variables on a dependent variable. There are several types of regression analysis, the simplest and most commonly used is linear regression analysis.

Variables in regression analysis are named according to their role. In our work we will use one dependent variable and one or more independent variables. Dependent variables will be denoted by Y, and independent variables by X. Depending on the measurement properties of the dependent variable, we must select the appropriate type of regression analysis (linear, logistical, ordinal, …). The regression model is determined on the basis of the regression equation 17 [19]:

$$Y = b_0 + b_1 X_1 + \varepsilon, \tag{17}$$

where b is the regression coefficient and ε is the constant of the model error.

A statistical regression model for prediction can be constructed using four steps:

1. Determination of the regression model
2. Determination of R and $R2$
3. Determine if R is statistically significant
4. Determine the importance of variables for forecasting

To develop the model, we will use the general regression equation 18:

$$\bar{Y} = b_1 X_1 + b_2 X_2 + \cdots + b_k X_k + a, \tag{18}$$

where $\bar{Y}$ is the predicted value of the measurable variable, b_k the regression coefficients for the corresponding variables and a the regression constant. The regression equation is determined using the least squares criterion, which requires minimization of $\sum(Y - \bar{Y})^2$. The regression equation can also be written with the equation 19:

$$Z_{\bar{Y}} = \beta_1 Z_1 + \beta_2 Z_2 + \cdots + \beta_K Z_K, \tag{19}$$

where Z_K is the predictors and criteria transformed into z estimates and β are beta coefficients.

R can be expressed by the following equation 20:

$$R = \sqrt{\beta_1 r_{\gamma 1} + \beta_2 r_{\gamma 2} + \cdots \beta_K r_{\gamma K}}, \tag{20}$$

where R is the correlation coefficient and $r_{\gamma K}$ is the predictive variable.

4. DEVELOPMENT OF PREDICTION MODELS

Prediction models will predict the roughness of the machined surface based on the following input data:

- Machine spindle speed [rpm],
- Tool feed speed [mm / min],
- Cutting depth [mm],
- Vibration.

For development of intelligent algorithm models, we will use a training data set consisting of 400 data samples. The performance of the developed models will then be compared with a test data set consisting of 92 data samples and finally with a flexible test data consisting of 36 data samples. For the construction of prediction models, we used the experimental results obtained from [1].

Intelligent tool wear prediction algorithms will be developed in the Matlab software tool, and a statistical regression model will be developed in the Microsoft Excel software tool.

4.1. ANFIS

ANFIS is based on a combination of soft logic and neural networks. The model for predicting surface roughness will be developed in the Matlab software tool, using the extension "Fuzzy Logic Toolbox™" [20].

The development of the ANFIS forecasting model consisted of the following steps:

1. Transfer of experimental data to Matlab,
2. Conversion of data into vector/matrix form,
3. Upload training and test data to Neuro-Fuzzy Designer,
4. Creating fuzzy inference system (FIS),
5. Fault tolerance selection and FIS training,
6. FIS testing.

The data was first saved in the form of a csv file in the Microsoft Excel software tool and named as follows: Train_data.csv, Test_data.csv and Flex_data.csv.

Data was transferred to the Matlab environment using the "readtable" command, which saves the data in the form of a table in Matlab.

The data stored in tabular form is not suitable for creating a model in the "Neuro-Fuzzy Designer" extension. Therefore, the data from the previous step had to be converted to vector / matrix format. The data were first divided into input data consisting of 4 inputs (Spindle Speed, Feed Rate, Depth of Cut, Vibration) and output data consisting of one output data (Ra). In Matlab, we used the "table2array" command to convert data from a table to a matrix / vector.

The ANFIS model will be created using the "Neuro-Fuzzy Designer" extension in the Matlab software tool. In order to successfully create a model, it is necessary to first load the relevant learning data. We did this using the "Load Data" command, which stored learning data with 900 data samples under "Training" and test data with 92 samples under "Checking". Data for flexible testing with 36 samples were not used in the development and learning of the ANFIS model.

FIS creation is possible using the "Generate FIS" function in the "Neuro-Fuzzy Designer" window. FIS can be created using two different options: "Grid partition" and "Sub. clustering". To make our model, the "Sub. clustering" option was used, for which the following parameters needed to be changed to create a model: Range of influence and Squash factor.

To create the model, ten different combinations of parameter values were used, which are shown in Table 1.

The developed FIS model needed to be properly trained for better performance in order to give better results when predicting. We did this in the "Train FIS" pane inside "Neuro-Fuzzy Designer". We can choose two different optimization methods for training: Hybrid and BPG. We chose the hybrid optimization method to train our model, as we tested both methods and found that using the hybrid method gives much better results.

Table 1

Selected values of parameters for ANFIS model

No.	Range of influence	Squash factor
1	0.5	1.25
2	0.3	1.25
3	0.1	1.25
4	0.8	1.25
5	0.5	1
6	0.5	0.5
7	0.5	1.5
8	0.1	1.8
9	0.1	2
10	0.2	2

Table 2

FIS training results

No.	Training RMSE	Checking RMSE
1	0.0017	0.0025
2	0.0008	0.0013
3	0.001	0.0002
4	0.097	0.097
5	0.0005	0.0004
6	0.0003	0.0004
7	0.0045	0.0097
8	0.00012	0.00017
9	0.00015	0.00017
10	0.0023	0.0028

Training was performed on all ten created models shown on Table 1. Table 2 shows the test results. The second line shows the relative error with respect to the training data (Training RMSE), and the relative error with respect to the test data (Checking RMSE). The last line shows the number of epochs we needed to learn the model to the optimal value.

From the results, we can see that we got the smallest error when using the model with sequence number 8, which had the smallest RMSE errors.

4.2. Artificial neural network (ANN)

Creating an ANN model for prediction consisted of the following steps:

1. Transfer of experimental data to Matlab.
2. Conversion of data into vector / matrix form.
3. Neural network initialization and configuration.
4. Neural network training.
5. Neural network testing.

The first and second steps have already been described in more detail in the development of the ANFIS model, so we will not describe them in detail here.

To create a neural network, we used the *feedforwardnet*(*X*) command, which creates a feedforward neural network with X number of neurons in the hidden layer. We developed four different neural networks, compared their performance and selected the neural network that best predicted the roughness of the treated surface. We created neural networks with 5, 10, 50 and 100 neurons in the hidden layer. After the successful construction of the neural network, it was

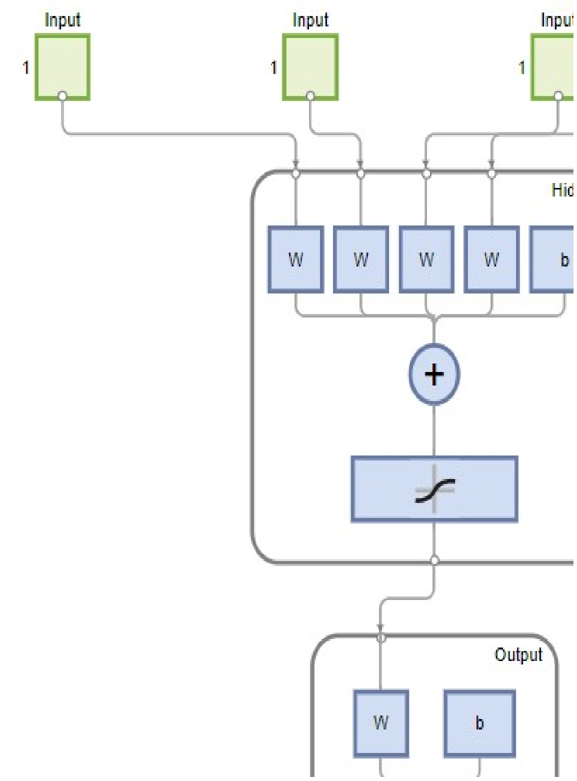

Fig. 3. Structure of developed ANN.

necessary to determine the number of parameters of the neural network so that it will work in accordance with our expectations. For our case number 4 was set as the number of inputs. Finally, the neural network needed to be configured based on the input experimental data, which we did using the "configure" command. Figure 3 shows the structure of the created neural networks.

The response of the constructed neural networks was tested using experimental training data used to teach the neural network.

From the test results, we found that the best results were obtained using a neural network with 50 neurons in the hidden layer, which had an average relative error of 0.0021 after 593 learning epochs.

4.3. Statistical regression model

A statistical regression model for prediction was constructed using the following equation 21:

$$Y_i = a_i + \beta_1 X_{1i} + \beta_2 X_{2i} + \beta_3 X_{3i} + \beta_4 X_{4i}, \quad (21)$$

where Y_i is the average surface roughness value, X_{1i} is the spindle speed, X_{2i} is the tool feed rate, X_{3i} is the cutting depth and X_{4i} is the absolute average of the vibrations per revolution.

For the construction of a statistical regression model, we used the experimental results obtained from [1]. The experiment included 4 different machine spindle speeds (750, 1000, 1250 and 1500 rpm), four different feed rates (152, 305, 457 and 610 mm / min) and 3 different cutting depths 0.25, 0.76, 1.27 mm). he statistical regression model was created in Microsoft Excel using the Data Analysis function, which enables the calculation of beta coefficients from which the equation of the regression model for prediction can be derived.

Table 3

Regression statistics

Multiple R	0.918337298
R Square	0.843343393
Adjusted R Square	0.841756997
Standard Error	0.36113793
Observations	400

Statistical analysis was performed on 400 samples of test data. Regression statistics are shown in Table 3.

The equation 22 represents the developed regression model.

$$Y_i = 3.1922 - 0.0013 \cdot X_{1i} + 0.0055 \cdot X_{2i} - 0.0009 \cdot X_{3i} - 0.0008 \cdot X_{4i} \quad (22)$$

5. RESULTS

This chapter will present the responses and effectiveness of the developed models for predicting the roughness of machined surface, which were developed in the previous chapter.

To determine the prediction efficiency of the regression model, we used the calculation of the relative error using equation 23:

$$\phi_i = \frac{|R_{ai} - \widehat{R_{ai}}|}{R_{ai}} \cdot 100\%. \quad (23)$$

where ϕ_i is the relative error of each data sample, R_{ai} is the actual measured surface roughness value and $\widehat{R_{ai}}$ is the predicted surface roughness value obtained from the prediction model. The efficiency of the whole data sample will be calculated by the equation 24:

$$\bar{\phi} = \frac{\sum_{i=1}^{m} \phi_i}{m}. \quad (24)$$

where $\bar{\phi}$ is the relative error of the whole data sample and m is the data sample size.

5.1. Results of the ANFIS model

Figure 4 shows the relative testing error [%] of the ANFIS model. The graph above shows the relative error of each of 400 training data samples. The center graph shows the relative error of each of 92 testing data samples. The graph below shows the relative error of each of 36 data samples for flexible testing data.

The average relative error when using training data was 0.0029 %, which means that the accuracy of the ANFIS model was 99.9971%. The average relative error when using testing data was 0.0039 %, which means that the accuracy of the ANFIS model was 99.9961%. The average relative error when using flexible testing data was 7.2277%, which means that the accuracy of the ANFIS model was 92.7723%.

5.2. Results of the ANN model

Figure 5 shows the relative testing error of the ANN model with 50 neurons in hidden layer. Results will be represented in three graphs.

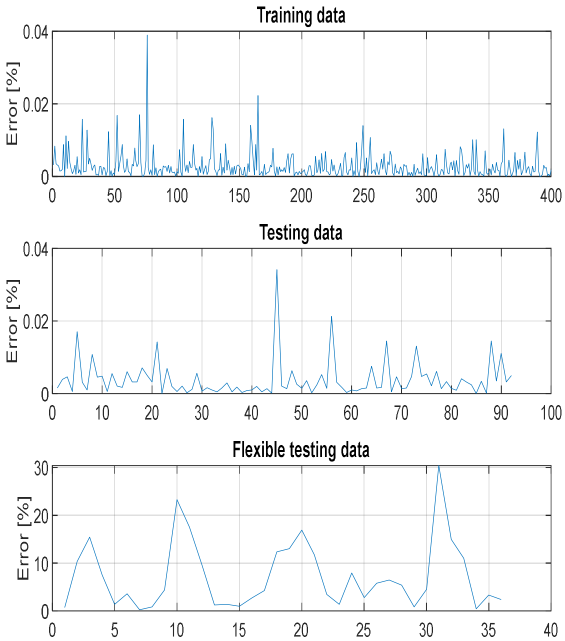

Fig. 4. Results of the ANFIS model.

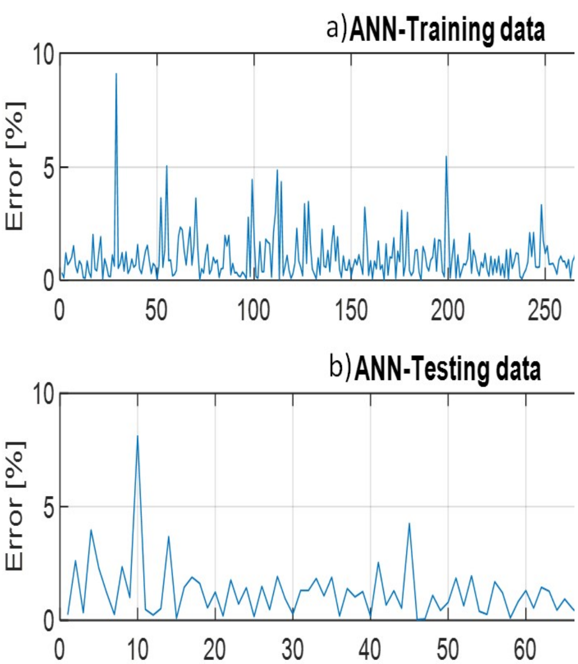

Fig. 5. Results of the ANN model.

The graph on Figure 5a shows the relative error of each training data sample. The graph on Figure 5b shows the relative error of each test data sample. The graph on Figure 5c shows the relative error of each data sample for flexible testing data.

The average relative error when using training data was 1.0105 %, which means that the accuracy of the ANN model was 98.9895 %. The average relative error when using testing data was 1.3435%, which means that the accuracy of the ANN model was 98.6565%. The average relative error when using flexible testing data was 10.9968 %, which means that the accuracy of the ANN model was 89.0032%.

5.3. Results of the statistical regression model

Figure 6 shows the relative testing error [%] of the regression model. The graph on Figure 6a shows the relative error of each training data sample. The graph on Figure 6b shows the relative error of each test data sample. The graph on Figure 6c the relative error of each data sample for flexible testing data.

The average relative error when using training data was 14.67%, which means that the accuracy of the regression model was 85.33%. The average relative error when using testing data was 12.54 %, which means that the accuracy of the regression model was 87.46%. The average relative error when using flexible testing data was 34.11%, which means that the accuracy of the regression model was 65.89%.

6. CONCLUSION

As part of the paper, various models for predicting the roughness of the machined surface during processing based on the following input data were developed:

1. Machine spindle speed [rpm],
2. Tool feed speed [mm / min],
3. Cutting depth [mm],
4. Vibration.

Three different prediction models were developed:

1. ANFIS model,
2. ANN model,
3. Statistical regression model.

Models were created based on 400 samples of experimental training data. The ANFIS model and the neural network model were developed in the Matlab software tool, and the statistical regression model was developed in the Microsoft Excel software tool.

The efficiency and accuracy of the models were first analyzed using training data, which contained 400 data samples, on the basis of which we developed these

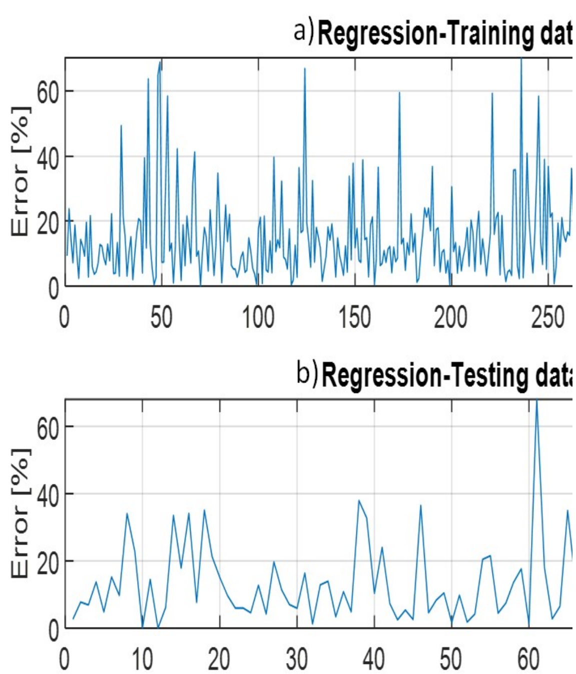

Fig. 6. Results of the regression model.

Table 5

Results of operation of prediction models

	Error [%]			Accuracy [%]		
	Train	Test	Flex	Train	Test	Flex
ANFIS	0.0029	0.0039	7.2277	99.9971	99.9961	92.7723
ANN	1.0105	1.3435	10.9968	98.9895	98.6565	89.0032
Reg.	14.67	12.54	34.11	85.33	87.46	65.89

models. The responses of the models were compared with the measured experimental values of the roughness of the machined surface. The operation of the developed models was then analyzed using testing data containing 92 data samples and finally using flexible testing data containing 36 data samples. Table 5 shows the results of the operation of prediction models.

When testing the performance of developed prediction models, we found that roughness is best predicted by the ANFIS model, which predicted with an accuracy of 99.9971% when using training data, with an accuracy of 99.9961% when using test data and with an accuracy of 92.7723% when using flexible testing data.

All developed intelligent models (ANFIS and ANN) had much better accuracy than the statistical regression model.

As part of the paper, the rapid construction, training and analysis of the operation of various prediction models in the Matlab software tool was demonstrated. The developed ANFIS and NM models have a very high accuracy when using testing data compared to statistical regression model. The ANFIS model and the NM model also managed to predict the roughness of the treated surface very well on the flexible testing data, which did not match the training data the most.

REFERENCES

[1] S.J. Lou, *Development of four in-process surface recognition systems to predict surface roughness in end milling*, Iowa State University, 1997.

[2] C.H. Chen, S. Y. Jeng, C. J. Lin, *Prediction and analysis of the surface roughness in CNC end milling using neural networks,* Applied Sciences, vol. 12, no. 1, 393, 2021.

[3] M.K. Shilpa, V. Yendapalli, *Surface roughness estimation techniques for drilled Surfaces: A review*, Materials Today: Proceedings, 2021.

[4] B. Bhushan, *Surface roughness analysis and measurement techniques,* Modern tribology handbook, two volume set, pp. 79–150, CRC press, 2000.

[5] M. Altug, M. Erdem, C. Ozay, *Experimental investigation of kerf of Ti6Al4V exposed to different heat treatment processes in WEDM and optimization of parameters using genetic algorithm,* The International Journal of Advanced Manufacturing Technology, vol. 78. no. 9, 2015, pp. 1573–1583.

[6] S. Ghobadi, K. Noghondarian, R. Noorossana, S.M. Mirhosseini, *Developing a fuzzy multivariate CUSUM control chart to monitor multinomial linguistic quality characteristics,* The International Journal of Advanced Manufacturing Technology, vol. 79, no, 9, 2015, pp. 1893–1903,.

[7] T.I. Liu, B. Jolley, *Tool condition monitoring (TCM) using neural networks,* The International Journal of Advanced Manufacturing Technology, vol. 78, no. 9, 2015, pp. 1999–2007.

[8] M. Elgargni, A. Al-Habaibeh, A. Lotfi, *Cutting tool tracking and recognition based on infrared and visual imaging systems using principal component analysis (PCA) and discrete wavelet transform (DWT) combined with neural* networks, The International Journal of Advanced Manufacturing Technology, vol. 77, no. 9, 2015, pp. 1965–1978.

[9] P.G. Benardos, G. C. Vosniakos, *Prediction of surface roughness in CNC face milling using neural networks and Taguchi's design of experiments,* Robotics and Computer-Integrated Manufacturing, vol. 18, no. 5-6, 2002, pp. 343–354.

[10] K.A. Risbood, U. S. Dixit, A. D. Sahasrabudhe, *Prediction of surface roughness and dimensional deviation by measuring cutting forces and vibrations in turning process*, Journal of Materials Processing Technology, vol. 132. No. 1–3, 2003, pp. 203–214.

[11] B.P. Huang, J. C. Chen, Y. Li, *Artificial-neural-networks-based surface roughness Pokayoke system for end-milling operations.* Neurocomputing, vol. 71, no. 4-6, 2008, pp. 544–549.

[12] M. Brezocnik, M. Kovacic, M. Ficko, *Prediction of surface roughness with genetic programming*, Journal of materials processing technology, vol. 157, 2004, pp. 28–36.

[13] H. Öktem, T. Erzurumlu, H. Kurtaran, *Application of response surface methodology in the optimization of cutting conditions for surface roughness*, Journal of materials processing technology, vol. 170, no. 1–2, 2005, pp. 11–16.

[14] W.H. Ho, J. T. Tsai, B. T. Lin, J. H. Chou, *Adaptive network-based fuzzy inference system for prediction of surface roughness in end milling process using hybrid Taguchi-genetic learning algorithm*, Expert Systems with applications, vol. 36, no. 2, 2009, pp. 3216–3222.

[15] Ho, W. H., Tsai, J. T., Lin, B. T., & Chou, J. H. (2009). *Adaptive network-based fuzzy inference system for prediction of surface roughness in end milling process using hybrid Taguchi-genetic learning algorithm*, Expert Systems with applications, vol. 36. no. 2, pp. 3216–3222.

[16] J.S. Jang, *ANFIS: adaptive-network-based fuzzy inference* system, IEEE transactions on systems, man, and cybernetics, vol. 23. no. 3, 1993, pp. 665–685.

[17] B. Yegnanarayana, *Artificial neural networks*, PHI Learning Pvt. Ltd., 2009.

[18] A. Abraham, *Artificial neural networks*, Handbook of measuring system design, 2005.

[19] A.O. Sykes, *An Introduction to Regression Analysis*, Coase-Sandor Institute for Law & Economics Working Paper No. 20, 1993.

[20] M. Works, *Fuzzy Logic Toolbox User's Guide, for Use the* Matlab, The Math Works, Inc., Natick, MA, Available at: `https://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy_ug.pdf`, accessed: 2022-05-15.