



Prediction of surface roughness in duplex stainless steel face milling using artificial neural network

Guilherme Augusto Vilas Boas Vasconcelos¹ · Matheus Brendon Francisco² · Lucas Ribeiro Alves da Costa² · Ronny Francis Ribeiro Junior¹ · Mirian de Lourdes Noronha Motta Melo¹

Received: 26 February 2024 / Accepted: 29 May 2024 / Published online: 8 June 2024
© The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2024

Abstract

This study proposes the inclusion of noise variables in an experimental design to develop a predictive model of surface roughness in the face milling process of duplex stainless steel. A central composite design arrangement was conducted, incorporating controlled variables (cutting speed, feed rate, milling width, and depth of cut) and noise variables (tool flank wear, fluid flow, and protrusion length). Each experimental configuration was employed in duplex stainless steel milling, with the collection of roughness data under each condition. The collected data were used to train eight configurations of artificial neural networks, which were then applied to predict roughness. The results indicate that the 7-20-14-1 network configuration exhibited the lowest root mean square error, which is a measure of the difference between predicted and observed values of (0.063), followed by 7-64-32-1 (0.064) and 7-14-12-1 (0.068), respectively. Additionally, these configurations also demonstrated the lowest mean absolute error values, which calculate the average of the absolute differences between predicted and observed values of (0.046, 0.053, and 0.055, respectively), and the coefficient of determination, which is a statistical measure indicating the proportion of data variability explained by the statistical model of (0.914, 0.908, and 0.901, respectively). Therefore, the inclusion of noise variables alongside controllable process factors resulted in a more accurate and robust predictive model of surface roughness for duplex stainless steel face milling.

Keywords Machining · Artificial intelligence · Mean square error · Surface quality · Robust parameter design

1 Introduction

Duplex stainless steels (DSS) are applied in varied industrial sectors, due to their exceptional properties, which combine remarkable mechanical characteristics, such as high strength and hardness, with an effective resistance to corrosion. These properties give DSS a wide range of possible uses, especially in the petroleum, chemical, and energy industry segments.

Stainless steels often exhibit a distinct machining behavior compared to other types of steel. They are mainly notable for their high hardening rates, which results in mechanical modifications and heterogeneous behavior on the machined surfaces, leading to the formation of unstable chips and vibrations. They

have low thermal conductivity, resulting in higher temperatures at the interfaces between the tool and the chip, as well as between the tool and the workpiece [1]. These characteristics contribute to more pronounced wear of the cutting tools and impair the quality of the machined surface finish [2].

In this context, the quality of the machined surface plays a critical role in evaluating the standard of excellence of the products. The measurement of surface roughness (R_a) is often employed as an essential metric to measure surface condition in machining operations [3]. Modeling techniques for R_a prediction can be categorized into three groups: experimental models, analytical models, and artificial intelligence (AI)-based models [4]. In recent times, models driven by artificial intelligence have emerged as the primary option and are widely adopted by researchers in the design of predictions related to machining processes. Several authors have employed artificial neural networks (ANNs) to predict surface roughness in machining operations [5].

The study by Thangarasu et al. [6] developed an artificial neural network model to predict surface roughness in the

✉ Guilherme Augusto Vilas Boas Vasconcelos
guilhermevasconcelos01@yahoo.com.br

¹ Mechanical Engineering Institute, Federal University of Itajubá–UNIFEI, Itajubá, Brazil

² Production and Management Engineering Institute, Federal University of Itajubá–UNIFEI, Itajubá, Brazil

machining process of EN8 steel. They trained a propagation neural network feedback with various algorithm approaches and evaluated performance based on the mean squared error and calculation time. The BFGS quasi-Newton backpropagation algorithm demonstrated the lowest mean squared error and minimum calculation time.

Yeganefar et al. [7] addressed the prediction and optimization of surface roughness and cutting forces during grooving in aluminum alloy 7075-T6. The authors employed regression analyses, support vector regression (SVR), artificial neural networks (ANN), and a multi-objective genetic algorithm. The performance of the regression, SVR, and RNA models was compared in relation to each response of the machining process.

Huang et al. [8] studied the prediction of tool wear based on controllable process parameters by deep convolutional neural network in milling operations. The experimental results show that the prediction accuracy of the proposed method is significantly higher than other advanced methods. The performance of the proposed tool wear prediction method is experimentally validated using three sets of run-to-tool failure data, measured from a high-speed CNC machine three-flute ball-tip tungsten carbide cutter under dry milling operations.

Wang et al. [9] predicted the cutting force in milling using a transfer net. This approach combines simulation data with transfer learning theory. Compared to the traditional neural network based on experimental samples, the transfer network has clear advantages. It reduces prediction error by using the same samples and requires fewer samples in total to achieve the same level of accuracy. Chen et al. [10] applied an artificial neural network to predict surface roughness in the CNC milling process. The experimental results show that the root mean square error (RMSE) obtained using the backpropagation neural network is 0.008.

Xie et al. [11] presented a multi-objective optimization of three-axis rough milling feed based on artificial neural network. Rodrigues et al. [12] presented a proposal for the application of artificial neural networks in the estimation of machining times for standard injection mold parts. Sharma et al. [13] applied an artificial neural network model to predict circularity errors in the milling process of stainless steel DNS2205. Sivarajan et al. [14] performed the prediction of surface roughness in EN31 hard machining steel with TiAlN-coated cutting tool using fuzzy logic. Arunadevi et al. [15] carried out the application of artificial neural networks to improve the performance of the CNC milling process, among several others studies.

Outemsaa et al. [16] presented an artificial model called BBNN to estimate the roughness of a machined surface. This model was adjusted based on four cutting parameters that are important for roughness: cutting speed, feed rate, depth of cut, and tool tip radius. The model is an ideal

BPNN (backward propagation neural network) whose hyperparameters are tuned, including learning algorithm, activation function, number of hidden layers, and number of neurons. The roughness function of the neural network has been optimized through a genetic algorithm to find the best cutting parameters. Several tests are performed to compare the accuracy of the optimized BBNN artificial model with other previous work. The results indicated that the developed model has a good precision in relation to the estimation of surface roughness.

Several interesting studies were published in current year. Deshpande et al. studied a predictive model for predicting the shear force using neural networks in duplex stainless steel machining. Knap et al. [17] applied long-term and short-term memory (LSTM) networks for tool wear detection in milling processes. The input parameters of the network were the controllable variables of the process. Ponnusamy and Tamilperuvalathan [18] evaluated the performance of a deep recurrent neural network for prediction with the aim of improving the machinability of SS304 with an optimal minimum amount of lubrication (MQL). Kumar et al. [19] performed the application of an artificial neural network in the end milling process to predict the material removal rate (MRR) values. Cheng et al. [20] proposed the prediction of tool wear in the milling process based on the BP neural network optimized by the firefly algorithm through the signal-to-noise ratio. Bai et al. [21] investigated the feasibility of combining milling stability analysis and a backpropagation neural network (BP) model to predict the surface roughness of aerospace aluminum alloy 7075Al in high-speed precision milling.

As can be seen, there is a lot of work being done in this area, showing great interest from the community on the subject. The application of artificial neural networks has been successful in predicting responses in machining processes, including surface roughness. However, few studies aim to incorporate uncontrollable variables, known as “noise,” into the controllable process variables to create more robust artificial neural network models. The inclusion of noise variables allows for a more comprehensive analysis of the relationship between input variables and quality characteristics, recognizing the impacts of external factors on outcome variability and identifying hidden patterns. It is crucial to consider noise to get realistic and resilient predictions.

The present study aims to close a gap in the literature and demonstrate that the use of noise variables in the deep learning model training process is extremely important. The results obtained in this work are very consistent with experimental results, demonstrating that the training of the model was done adequately.

This article is divided into 6 sections: Section 2 presents a review of the literature. Section 3 discusses the research

methodology adopted, offering valuable information about the process and procedures used. Section 4 presents the results and discussions, providing a critical and enlightening analysis of the data collected. Finally, Section 5 presents the conclusions of this work, offering an engaging synthesis of all the elements addressed, culminating in a satisfactory and significant result, and Section 6 presents the references that provided a scientific basis for the study.

2 Background

2.1 Design of experiments

The concept of design of experiments (DOE) is a statistical methodology employed to systematically and efficiently plan, execute, and analyze experiments. The primary objective of DOE is to obtain valuable and pertinent insights into how specific variables impact a process or system. By using this methodology, researchers can enhance processes, refine products, and pinpoint the critical factors that affect the experiment's outcomes [22].

Within the context of DOE, factorial design stands out as an invaluable strategy for exploring the influence of multiple factors on an experimental system. These factors often comprise independent variables, such as varying levels of speed, depth, and cutting feed, which can significantly influence the outcomes of machining processes. Factorial design entails testing all conceivable combinations of factor levels, facilitating the analysis of each factor's primary effects as well as their interplay. This approach proves especially useful when discerning which factors exert the greatest impact on the experiment's outcomes and how they may interact.

Fractional factorial design extends the concept of factorial design and comes into play when the number of possible combinations of factor levels becomes impractical to test comprehensively. In certain scenarios, assessing every conceivable combination can be costly, time-consuming, or unfeasible. Consequently, researchers opt for a fractional factorial design, which entails strategically selecting a subset of the total combinations for testing. This selection is meticulously made using a fractional plan derived from the complete design.

For instance, consider an experiment with three factors, each possessing two levels (high and low). In a full factorial design, eight combinations ($2^3 = 8$) would require testing, whereas a fractional plan might select only four combinations, thus saving time and resources. While the fractional factorial design can yield significant information regarding the primary effects of factors, it may not fully capture certain interactions due to not testing all conceivable combinations. The choice of a fractional plan depends on various factors, including the experiment's nature, the number of factors involved, and the research's objectives.

In summary, both full factorial design and fractional factorial design are DOE techniques enabling the study of how multiple factors affect a system. The former covers all potential combinations, while the latter strategically chooses a subset to conserve resources while still yielding pertinent information about factors and their interactions.

Taking it a step further, the combined design is employed to investigate how both controllable and noise factors impact the response or variable of interest in an experiment. The core concept is to create an experimental plan allowing for the control and measurement of both types of factors to assess their influence on the outcomes. For instance, in a manufacturing study aiming to optimize roughness in machining processes, controllable factors might include cutting speed, feed rate, and depth, while noise factors could encompass variations in raw material quality or tool wear.

The combined design permits the design of an experiment that manages the selected (controllable) factors while also capturing and considering random variations (noise). This approach offers a more comprehensive analysis of how factors affect product roughness and aids in identifying the optimal configuration for achieving smoother surfaces.

In summary, the combined design represents a potent approach to experimental design that encompasses both controllable and noise factors. This technique facilitates a deeper understanding of how these factors influence experiment outcomes and empowers researchers to make informed decisions for process or product optimization.

2.2 Artificial neural network

Generally speaking, a neural network is a system designed to emulate the process by which the brain performs a specific task. It is usually built using electronic components or is simulated by propagating on a digital computer. To achieve effective performance, neural networks make use of a vast network of simple computational processing units, known as "neurons" [9].

The origin of neural networks dates back to the creation of the mathematical model of the biological neuron, which was proposed by Warren McCulloch and Walter Pitts in 1943 [23]. This model, known as the MCP neuron (McCulloch-Pitts), is characterized by a set of n inputs that are multiplied by specific weights and then the results are summed and compared to a threshold [24].

In 1958, Frank Rosenblatt presented a network configuration known as a "perceptron," which consisted of MLP neurons (multilayer perceptrons) arranged in a two-layered network [25]. This approach fueled a wave of research related to neural networks until 1969. However, in the same year, the publication by Minsky and Papert [26] revealed deficiencies and limitations in the MLP model, resulting in a decrease in interest in the area of studies related to ANNs. It was not until

1982, with the publication of Hopfield's (1982) work, that there was a resurgence of interest in neural networks.

Neural networks are often employed to solve complex problems in which the behavior of variables is not completely known. One of its fundamental characteristics is the ability to learn from examples and apply this knowledge in a generalized way, resulting in the creation of nonlinear models. This capability makes its application in spatial analysis highly effective [10].

When it comes to configuration, the implementation of a neural network requires the definition of several important variables, including (a) the number of nodes in the input layer (this variable corresponds to the number of variables that will serve as input to the neural network, usually representing the variables most relevant to the problem under analysis), (b) the number of hidden layers and the number of neurons allocated in these layers, and (c) the number of neurons in the outflow layer [10].

2.3 Neural network architecture

Artificial neural networks (ANNs) are computational algorithms that are inspired by the structure of intelligent beings, allowing the simplified incorporation of the functioning of the human brain into computers. Just like the human brain, ANNs have the ability to learn and make decisions based on their own experience. In essence, an ANN is a processing system that can acquire knowledge through learning and make it available for application in specific contexts.

According to Haykin [27], the neural network shares two fundamental characteristics with the human brain: (a) the acquisition of knowledge occurs through the process of learning from the environment and (b) the strengths of the connections between neurons (synaptic weights) are used to store the acquired knowledge.

A specific set of inputs and processing units is interconnected through synaptic weights. The inputs are transmitted through the structure of the neural network, where they are modified by the synaptic weights and the activation function (AF) of the neurons, as described by Machado et al. [28]. When it receives inputs from n neurons (y_i), the k neuron calculates its output, as shown in Eq. 1:

$$y_k = \left(\sum_{i=1}^n (y_i w_{ki}) + b_k \right) \quad (1)$$

where y_i represents the output calculated by neuron i , w_{ki} denotes the synaptic weight between neuron i and neuron k , and it is the weight associated with a constant, non-zero value, known as the bias of neuron k y_i .

To use an artificial neural network (ANN), it is essential to calculate synaptic weights and biases. This process of determining these parameters is called training and occurs in an iterative way, where the initial parameters are employed until

the process reaches convergence. Regarding the j -interaction, the weight w_{ki} is applied according to Eq. 2.

$$w^{(j)}_{ki} = w^{(j-i)}_{ki} + \Delta w^{(j)}_i \quad (2)$$

with $\Delta w^{(j)}$ being the correction vector to the parameter w_{ki} in iteration j .

2.4 Activation function

The activation function describes how the internal input and the current activation state influence the determination of the unit's next activation state. Each unit in the network can incorporate a non-linearity in its output, which needs to be mitigated. According to Chen et al. [10], various activation functions are available, with the most popular being the following.

The piecewise linear function can be interpreted as an approximation of a nonlinear amplifier (as shown in Fig. 1a) and is represented in Eq. 3:

$$f(u) = u \quad (3)$$

In the piecewise linear function, the amplification factor is considered to be equal to one within the linear operating range. This function can be seen in two special ways: (a) when the linear region of operation does not go into saturation, it transforms into a linear combinator and (b) if the amplification factor in the linear region is infinitely large, the piecewise linear function becomes a threshold function.

Threshold functions are a subset of Boolean functions. In summary, a weight w_i is assigned to each x_i . The value of the function will be 1 if the weighted sum of the inputs is greater or equal to a value T . If the sum does not reach T , then the output of the function is 0 (as shown in Fig. 1b). Equation 4 represents the Threshold function:

$$f(u) = \begin{cases} 1, & \sum_{i=1}^n w_i x_i \geq T \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

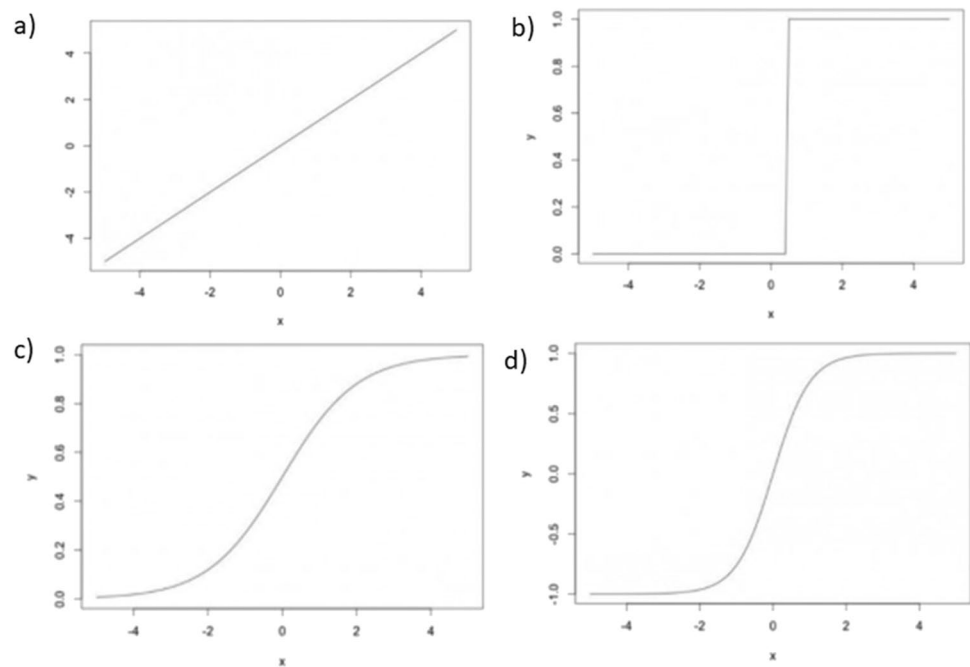
Sigmoidal function: this function is the most frequently used and is characterized by being an increasing function that appropriately balances linear and nonlinear behavior, maintaining its range of variation between 0 and 1 (as shown in Fig. 1c). An example of a sigmoidal function is the logistic function, the definition of which is represented in Eq. 5:

$$f(u) = \left(\frac{1}{1 + \exp(-a\mu)} \right) \quad (5)$$

where a is the slope parameter of the Sigmoid function (the higher the value of a , the steeper the curve becomes).

The hyperbolic tangent function is often preferred to the logistic function, since the latter only generates activation

Fig. 1 Activation function. **a** Piecewise linear function. **b** Threshold function. **c** Sigmoid function. **d** Hyperbolic tangent function



values in the interval $(0, 1)$. The hyperbolic tangent function, on the other hand, retains the Sigmoid shape of the logistic function but encompasses both positive and negative values. To obtain an equivalent sigmoid function, we can use the hyperbolic tangent function, which is defined according to Eq. 6 and shown in Fig. 1d:

$$f(u) = \tanh(\mu) \quad (6)$$

2.5 Multilayer perceptrons (MLP)

The perceptron, which was introduced by Rosenblatt in 1958, represents an elementary form of neural network, whose primary application lies in the area of pattern classification. The single-layer perceptron possesses the ability to only classify patterns that can be separated linearly. However, in practical situations, it is often not feasible to achieve a perfect linear separation. This requires the use of a multi-layered neural network [29].

Structures known as MLPs, or multilayer perceptrons, are widely recognized as the most common models of artificial neural networks. An MLP is made up of several layers, including the input layer, one or more intermediate layers (also known as hidden layers), and the output layer, as per [29].

Following this same line of reasoning, Akinwekomi et al. (2021) emphasize that a multilayered neural network is usually composed of organized layers of neurons. The ingress layer forwards the ingress information to the hidden layer(s) of the network. At the output layer, the solution to

the problem is obtained. Hidden layers play an intermediate role, whose primary function is to separate the information of the input layer from the output layer. It is important to note that the neurons of one layer are connected only to the neurons of the immediately subsequent layer, and there is no feedback or connections between neurons within the same layer. In addition, it is a typical feature that all layers are fully connected.

Looking at Fig. 2, we can see an example of an RNA network structure, which consists of three layers: the input layer, the hidden layer, and the output layer. In this structure, the input layer has seven nodes, the hidden layer has 8 nodes, and the output layer has a single node. The seven nodes in the input layer represent the seven decision values of the case study: cutting speed (v_c), feed rate (F), depth of cut (ap), milled width (ae), fluid flow (Q), cantilevered length (lt_0), and tool wear (vb). The node in the output layer represents the predicted value of surface roughness. The presented network has all the connections, which means that a neuron in any layer of the network is connected to all the other neurons in the previous layer. The flow of signals through the network is done positively, from left to right, layer by layer. When we consider applying a multi-layer feeder network in the m th hidden layer, with j , k , and l nodes in each hidden layer, the example structure shown in Fig. 2 can be described as a $7-j-k-l-1$ configuration. In general terms, the operation of this type of network can be described in terms of two main phases: the advance phase and the backpropagation phase [4].

The process of training MLP networks (multilayer perceptrons) using the backpropagation algorithm (BP) can be divided

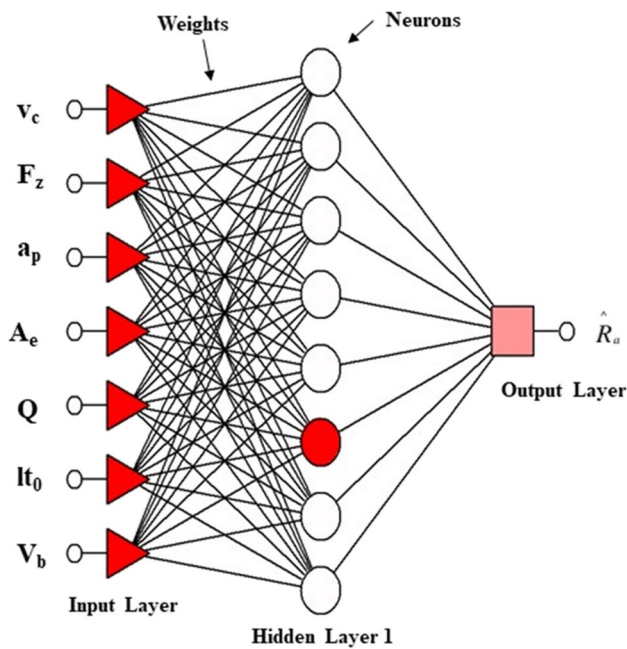


Fig. 2 Example illustration of an ANN network structure with layers and node

into two distinct phases: propagation and backpropagation. In the propagation phase, an activation pattern is applied to neurons in the input layer of the network, and its effects propagate through the network, layer by layer. Upon reaching the last layer, a set of outputs is generated, representing the actual response of the network. In the backpropagation phase, all synaptic weights are adjusted according to an error correction rule. The error signal is propagated back through the network, against the direction of the synaptic connections, and the synaptic weights are adapted to make the actual response of the network approximate the desired response, in statistical terms [10].

An essential feature of MLP networks is the non-linearity of neuron outputs. This nonlinearity is achieved through the use of an activation function, usually of the Sigmoid type, commonly known as the logistic function, as presented in Eq. 5.

To successfully create an artificial neural network (ANN) model, it is essential to go through a process of experimentation and adjustment, considering several elements. Many researchers use ANNs for modeling in various areas, such as machining, but there are still no definitive guidelines for creating the ideal model. Because of this uncertainty, this research explores the elements that may affect the efficacy of the RNA model, based on the features of the TensorFlow library, in order to develop the desired RNA model.

2.6 Performance indicators

To accurately assess our predictive models' accuracy in estimating surface roughness values, we have selected four

distinct performance indicators. These metrics include the coefficient of determination (R^2), the mean absolute error (MAE), the mean squared error (MSE), the square root of the mean squared error (RMSE), and the mean absolute percentage error (MAPE), expressed as a percentage of the actual value, as detailed in Eqs. 7 to 11, respectively.

R^2 represents the proportion of the variance in Y that is predictable from the independent variable X ; a value closer to 1 indicates a greater ability of the model to explain and predict the observed values of Y . The MAE is a measure of absolute error ($|y-\hat{y}|$) that takes into account the total number of observations/predictions and is therefore expressed in the same units of measurement. The MSE and RMSE are characterized by the mean squared error and its square root, respectively. If the MSE is expressed in a unit that is hard to interpret, the square root calculated in the RMSE makes it expressed in the same unit of measurement as the observations, which facilitates its interpretation. MAPE represents the average of the absolute percentage errors, making it easier to compare between predictive models with different variables of interest.

Analyzing these metrics is crucial for a comprehensive evaluation of our models' predictive performance. By understanding the significance of these metrics, we can objectively assess the precision and effectiveness of our predictions, ensuring that our models are reliable tools for guiding decision-making processes and future strategies.

$$R^2 = \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^n (Y_i - \bar{Y})^2} \quad (7)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=0}^{n-1} |Y_i - \hat{Y}_i| \quad (8)$$

$$\text{MSE} = \frac{1}{n} \sum_{i=0}^{n-1} |Y_i - \hat{Y}_i|^2 \quad (9)$$

$$\text{RMSE} = \frac{1}{n} \sum_{i=1}^{n-1} (Y_i - \hat{Y}_i)^2 \quad (10)$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n-1} \frac{|Y_i - \hat{Y}_i|}{Y_i} \times 100 \quad (11)$$

Here, ' n ' represents the number of data points, ' Y_i ' denotes observed values, ' \hat{Y} ' represents predicted values, and ' \bar{Y} ' signifies the mean value of ' Y '.

When comparing these metrics, particular emphasis will be placed on RMSE as the preferred evaluation criterion. This

preference arises because RMSE is a more suitable method when model errors follow a normal distribution, as opposed to MAE. Furthermore, RMSE offers an advantage over MAE by avoiding the use of absolute values, which may not be desirable in many mathematical calculations [30]. Consequently, when evaluating the accuracy of various regression models, RMSE is a more appropriate choice due to its ease of calculation and differentiability. Additionally, a higher R^2 value is considered favorable.

It is worth noting that prior to employing machine learning models, a preliminary examination of the data will be conducted. An essential aspect of this examination is identifying and addressing outliers, which can significantly impact the accuracy of machine learning models. Outliers can distort results and undermine the model's ability to effectively generalize patterns within the data. The presence of outliers can also violate statistical assumptions, potentially compromising the validity of analyses and resulting interpretations [31].

Certain algorithms are sensitive to outliers, implying that their performance can be severely affected by the presence of such data points. Outliers may emerge due to measurement errors or data corruption. Therefore, the detection and correction of outliers are imperative to ensure data quality and integrity for model training. Consequently, conducting an outlier analysis on the data before applying machine learning algorithms is fundamental for obtaining more precise, robust, and dependable models, while also upholding the validity of statistical analyses and data quality.

In the realm of model performance assessment, overfitting can occur when a model excessively tailors itself to the training data, even capturing noise and outliers present within it. This results in a model that struggles to generalize effectively to new data. By addressing outliers, it is possible to mitigate the risk of overfitting and enhance the model's capacity to make accurate predictions on unseen data [32].

Lastly, optimizing machine learning models is a primary challenge in achieving effective machine learning solutions. Hyperparameter optimization aims to identify the optimal values for model parameters, ultimately yielding the best performance as measured by the validation set, within a given machine learning algorithm. These hyperparameters control the learning process and have a significant impact on predictive performance. Proper selection of hyperparameters can also help mitigate overfitting and underfitting issues, thereby enhancing prediction accuracy [33]. In this study, a comprehensive analysis of various hyperparameters was conducted using a GridSearch library, and the most suitable ones were selected for implementation.

3 Methodology

The top milling operation was performed in a ROMI D600 machining center, as shown in Fig. 3, with a power of 15 kW and a maximum rotation of 10,000 rpm. The part

to be machined is duplex stainless steel, which has low machinability due to its low thermal conductivity. The chemical structure of duplex stainless steel UNS S32205 is mentioned in Table 1. The insert used in the cutting operation was the CoroMill R390-11T308M-MM 2030, made of carbide and with double layer of titanium nitride (TiN) and aluminum titanium nitride (TiAlN), coated by the process of physical vapor deposition (PVD), fixed in the CoroMill® R390-025A25-11 M support, with a diameter of 25 mm, position angle $\chi r = 90^\circ$, cylindrical rod, with 3 inserts and mechanical fixation by tweezers. Both the inserts and the tool holder were provided by Sandvik Coromant.

The research will gather data using a statistical method called the design of experiments, specifically employing a CCD arrangement. This design includes both controllable and uncontrollable variables. The controllable factors are cutting speed, tooth advance, cutting depth, and cutting width, as detailed in Table 2. In contrast, uncontrollable parameters such as the cantilevered tool length, cutting fluid flow rate, and flank wear are outlined in Table 3. The response parameter of interest is surface roughness, which

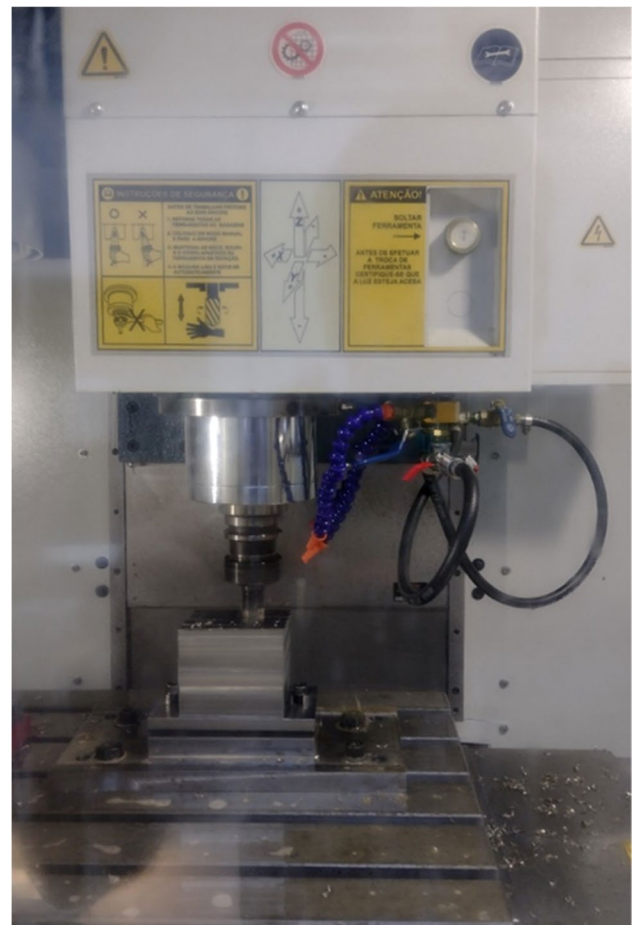


Fig. 3 ROMI® D 600 machining center

Table 1 Chemical composition (% by weight) of duplex stainless steel UNS S32205

C	Mn	P	S	N	Al	Cr	Mo	Ni	Cu	W	Co
0.47	1.22	0.019	0.01	0.19	0.008	22.24	3.14	5.62	0.19	0.02	0.05

Table 2 Control variables and their respective levels

Levels for controllable variables					
Coded levels	- 2.83	- 1.00	0.00	+1.00	+2.83
Cutting speed (m/min)	32.57	60	75.00	90.00	117.43
Tooth feed (mm/tooth)	0.04	0.10	0.13	0.16	0.21
Work penetration (mm)	12.26	15.00	16.50	18.00	20.74
Cutting depth (mm)	0.43	0.80	1.00	1.20	1.57

Table 3 Levels defined for noise variables

Noise variables	Process levels		
	- 1	0	+1
Flank wear (mm)	0	0.15	0.30
Fluid flow (l/min)	0	0.20	20
Cantilevered length (mm)	30	40	50

was measured using a portable Mitutoyo Surftest 201 roughness meter, calibrated before data collection. To minimize potential errors stemming from unmeasured or unknown variables, the experiments were conducted randomly.

Fig. 4 Overhang length l_{t0} : **a** items used, **b** bench to open and close clamp

(a)



(b)

To control the overhang length (l_{t0}) during the experimental tests, a set of clamping devices was used, as shown in Fig. 4. The value of l_{t0} was verified using a Digimess® analog caliper with a resolution of 0.05 mm.

Regarding the amount of fluid (Q), two regulating valves (1 and 2) were used to control the flow during the face milling of duplex stainless steel UNS S32205. To ensure minimal flow in the machine tool, a small opening was made in valve 1, and the flow rate was measured using a graduated beaker. For maximum flow, both valves were fully opened. In the case of “dry” machining, the valves were closed to prevent the fluid from being directed to the cutting area. The valves used to control the fluid quantity in the process can be observed as shown in Fig. 5.

During the execution of the experiments, the measurements of tool flank wear (v_b) were obtained using the image analyzer (Global Image Analyzer), the Global Lab 97 Image software, and the stereoscopic microscope model SZ 61 (with 45 times magnification), as shown in Fig. 6.

Surface roughness measurements were obtained using a calibrated Mitutoyo Surftest 201 portable roughness tester before the start of measurements, as shown in Fig. 7. The cutoff parameter was set to 0.8 mm for all measurements, as for this sampling length, roughness values of R_a are expected to vary between 0.1

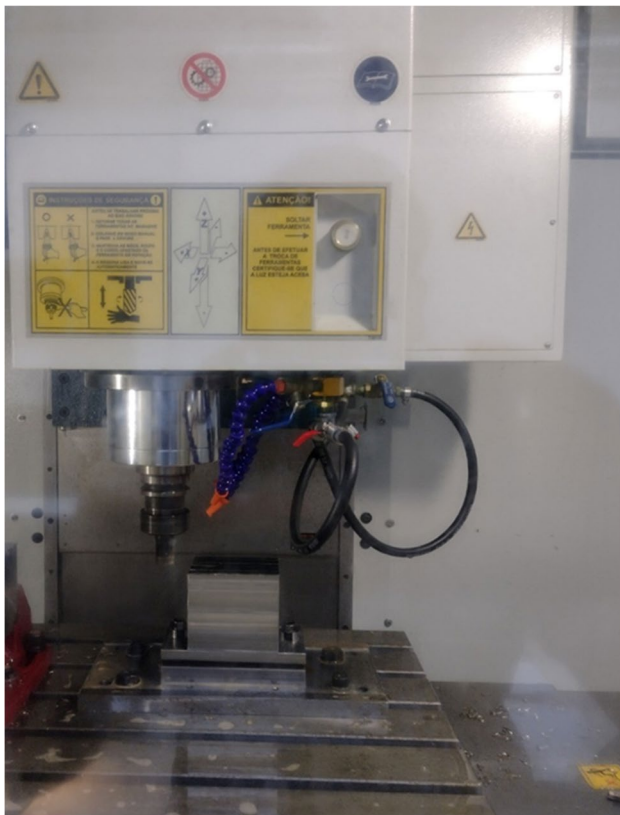


Fig. 5 Fluid quantity control

and 2 micrometer-meter. The measurements were taken perpendicular to the machining groove. Measurements were made at the beginning, in the middle, and at the end. Table 4 displays the experimental matrix used for collecting surface roughness data. The axial points of the noise were excluded from this matrix, as machining them is physically impossible.

After conducting the experiments, we proceed to the stage of building the artificial neural network model. The experimental

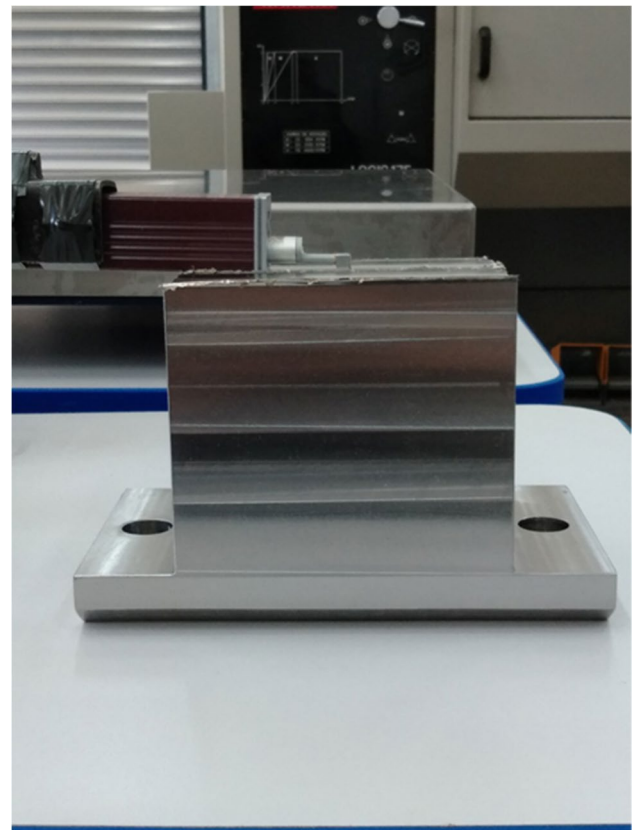
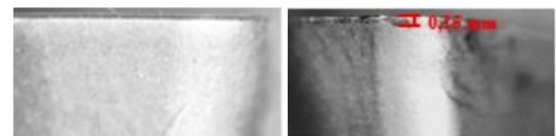


Fig. 7 Roughness measurement

data were divided into training and test sets, representing 70% and 30% of the total number of experiments performed, which corresponds to 50 training attempts and 22 test attempts. All models were constructed using the Python language and the TensorFlow library. The training and test datasets underwent a normalization process, adjusting the values to ensure a consistent scale and distribution for all variables. This normalization process was executed to set the mean of the values to 0 and

Fig. 6 Flank wear of cutting inserts



$v_b = 0 \text{ mm}$

$v_b = 0,15 \text{ mm}$



$v_b = 0,30 \text{ mm}$

Table 4 Experimental matrix

EXP	Controllable variables				Noise variables			Response
	vc (m/min)	Fz (mm/tooth)	ae (mm)	Ap (mm)	VB (mm)	Q (l/min)	lt0 (mm)	Ra (μ m)
1	60.00	0.10	15.0	0.80	0.00	0.00	50.0	0.520
2	90.00	0.10	15.0	0.80	0.00	0.00	30.0	0.347
3	60.00	0.16	15.0	0.80	0.00	0.00	30.0	0.630
4	90.00	0.16	15.0	0.80	0.00	0.00	50.0	0.717
5	60.00	0.10	18.0	0.80	0.00	0.00	30.0	0.503
6	90.00	0.10	18.0	0.80	0.00	0.00	50.0	0.360
7	60.00	0.16	18.0	0.80	0.00	0.00	50.0	0.657
8	90.00	0.16	18.0	0.80	0.00	0.00	30.0	0.747
9	60.00	0.10	15.0	1.20	0.00	0.00	30.0	0.503
10	90.00	0.10	15.0	1.20	0.00	0.00	50.0	0.303
11	60.00	0.16	15.0	1.20	0.00	0.00	50.0	0.653
12	90.00	0.16	15.0	1.20	0.00	0.00	30.0	0.700
13	60.00	0.10	18.0	1.20	0.00	0.00	50.0	0.487
14	90.00	0.10	18.0	1.20	0.00	0.00	30.0	0.317
15	60.00	0.16	18.0	1.20	0.00	0.00	30.0	0.653
16	90.00	0.16	18.0	1.20	0.00	0.00	50.0	0.737
17	60.00	0.10	15.0	0.80	0.30	0.00	30.0	0.837
18	90.00	0.10	15.0	0.80	0.30	0.00	50.0	0.867
19	60.00	0.16	15.0	0.80	0.30	0.00	50.0	0.637
20	90.00	0.16	15.0	0.80	0.30	0.00	30.0	0.960
21	60.00	0.10	18.0	0.80	0.30	0.00	50.0	0.930
22	90.00	0.10	18.0	0.80	0.30	0.00	30.0	0.813
23	60.00	0.16	18.0	0.80	0.30	0.00	30.0	0.667
24	90.00	0.16	18.0	0.80	0.30	0.00	50.0	1.003
25	60.00	0.10	15.0	1.20	0.30	0.00	50.0	0.958
26	90.00	0.10	15.0	1.20	0.30	0.00	30.0	0.858
27	60.00	0.16	15.0	1.20	0.30	0.00	30.0	0.683
28	90.00	0.16	15.0	1.20	0.30	0.00	50.0	1.033
29	60.00	0.10	18.0	1.20	0.30	0.00	30.0	0.971
30	90.00	0.10	18.0	1.20	0.30	0.00	50.0	0.719
31	60.00	0.16	18.0	1.20	0.30	0.00	50.0	0.670
32	90.00	0.16	18.0	1.20	0.30	0.00	30.0	1.097
33	60.00	0.10	15.0	0.80	0.00	20.0	30.0	0.453
34	90.00	0.10	15.0	0.80	0.00	20.0	50.0	0.277
35	60.00	0.16	15.0	0.80	0.00	20.0	50.0	0.570
36	90.00	0.16	15.0	0.80	0.00	20.0	30.0	0.630
37	60.00	0.10	18.0	0.80	0.00	20.0	50.0	0.403
38	90.00	0.10	18.0	0.80	0.00	20.0	30.0	0.270
39	60.00	0.16	18.0	0.80	0.00	20.0	30.0	0.487
40	90.00	0.16	18.0	0.80	0.00	20.0	50.0	0.657
41	60.00	0.10	15.0	1.20	0.00	20.0	50.0	0.440
42	90.00	0.10	15.0	1.20	0.00	20.0	30.0	0.287
43	60.00	0.16	15.0	1.20	0.00	20.0	30.0	0.523
44	90.00	0.16	15.0	1.20	0.00	20.0	50.0	0.693
45	60.00	0.10	18.0	1.20	0.00	20.0	30.0	0.430
46	90.00	0.10	18.0	1.20	0.00	20.0	50.0	0.243
47	60.00	0.16	18.0	1.20	0.00	20.0	50.0	0.540
48	90.00	0.16	18.0	1.20	0.00	20.0	30.0	0.667

Table 4 (continued)

EXP	Controllable variables				Noise variables			Response
	vc (m/min)	Fz (mm/tooth)	ae (mm)	Ap (mm)	VB (mm)	Q (l/min)	lt0 (mm)	Ra (μ m)
49	60.00	0.10	15.0	0.80	0.30	20.0	50.0	0.716
50	90.00	0.10	15.0	0.80	0.30	20.0	30.0	0.707
51	60.00	0.16	15.0	0.80	0.30	20.0	30.0	0.447
52	90.00	0.16	15.0	0.80	0.30	20.0	50.0	0.840
53	60.00	0.10	18.0	0.80	0.30	20.00	30.0	0.820
54	90.00	0.10	18.0	0.80	0.30	20.0	50.0	0.773
55	60.00	0.16	18.0	0.80	0.30	20.0	50.0	0.510
56	90.00	0.16	18.0	0.80	0.30	20.0	30.0	0.800
57	60.00	0.10	15.0	1.20	0.30	20.0	30.0	0.823
58	90.00	0.10	15.0	1.20	0.30	20.0	50.0	0.723
59	60.00	0.16	15.0	1.20	0.30	20.0	50.0	0.593
60	90.00	0.16	15.0	1.20	0.30	20.0	30.0	0.830
61	60.00	0.10	18.0	1.20	0.30	20.0	50.0	0.873
62	90.00	0.10	18.0	1.20	0.30	20.0	30.0	0.603
63	60.00	0.16	18.0	1.20	0.30	20.0	30.0	0.620
64	90.00	0.16	18.0	1.20	0.30	20.0	50.0	0.870
65	32.50	0.13	16.5	1.00	0.15	0.20	40.00	0.754
66	117.40	0.13	16.50	1.00	0.15	0.20	40.00	0.767
67	75.00	0.05	16.50	1.00	0.15	0.20	40.00	0.692
68	75.00	0.21	16.50	1.00	0.15	0.20	40.00	1.071
69	75.00	0.13	12.26	1.00	0.15	0.20	40.00	0.933
70	75.00	0.13	20.74	1.00	0.15	0.20	40.00	0.940
71	75.00	0.13	16.50	0.43	0.15	0.20	40.00	0.743
72	75	0.13	16.5	1.57	0.15	0.2	40	0.784

the standard deviation to 1, ensuring that the variables had a uniform and comparable pattern.

Subsequently, hyperparameter optimization was performed. The hyperparameters were optimized using the GridSearch library. There are various common strategies for optimizing hyperparameters, including manual tuning, grid search, random search, Bayesian optimization, gradient-based optimization, and evolutionary optimization [34]. In this study, we utilized grid search with the GridSearch CV method, a traditional technique for adjusting hyperparameters. This approach enables finding the best hyperparameters through a grid of combinations in each order [35]. Several hyperparameters were tested for the neural networks, and the best grid values found for the models are presented in Table 5. The complete methodology is illustrated in Fig. 8.

4 Results and discussion

4.1 Outlier precision in neural network

The examination of outliers in the controllable variables within this study is depicted in Fig. 9. Notably, no

Table 5 Hyperparameters for neural network models

Model	Hyperparameters	Results
Sequential	Activation hidden layers	ReLU
	Activation output	Linear
	Kernel_regularizer	L2
	Optmizer	Adam
	Learning_rate	0.002
	Loss	Mean_squared_error
	Metrics	Mean_absolute_error
	Epochs	2000

outliers are observed among the controllable variables. It is important to highlight that the analysis of outliers in noise variables is typically omitted. These variables are often perceived as stochastic and beyond control. Noise variables in a dataset contribute to unexplained variance, independent of the explanatory variables and the model itself. Handling outliers in noise variables differs from how outliers in variables of interest are treated. Typically, outliers in noise variables are not considered problematic or requiring correction or removal. They are viewed

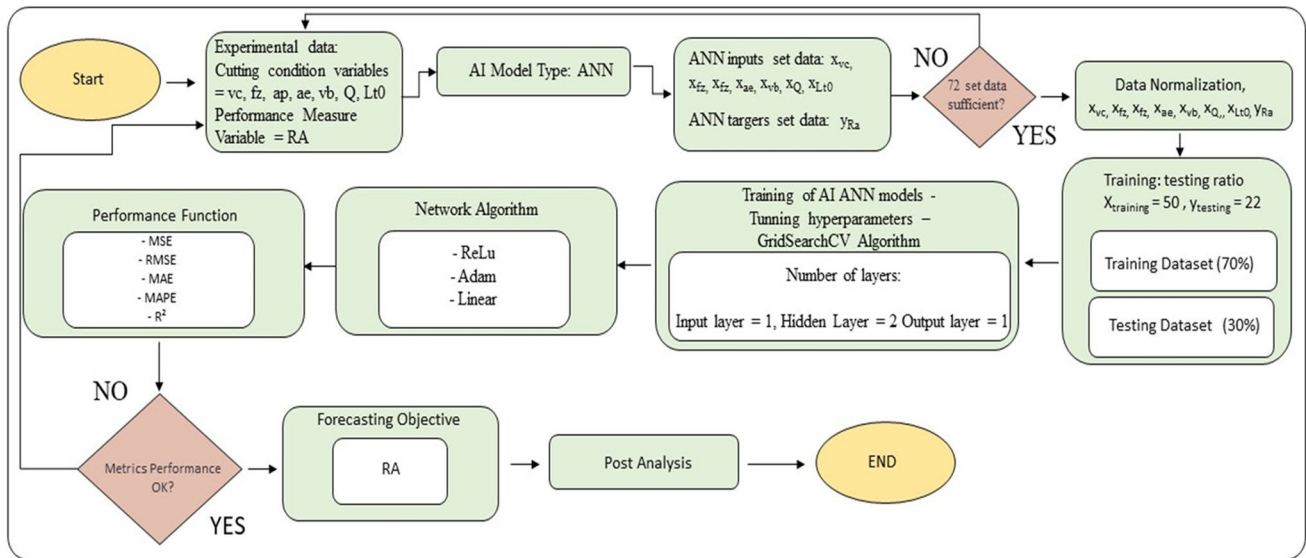
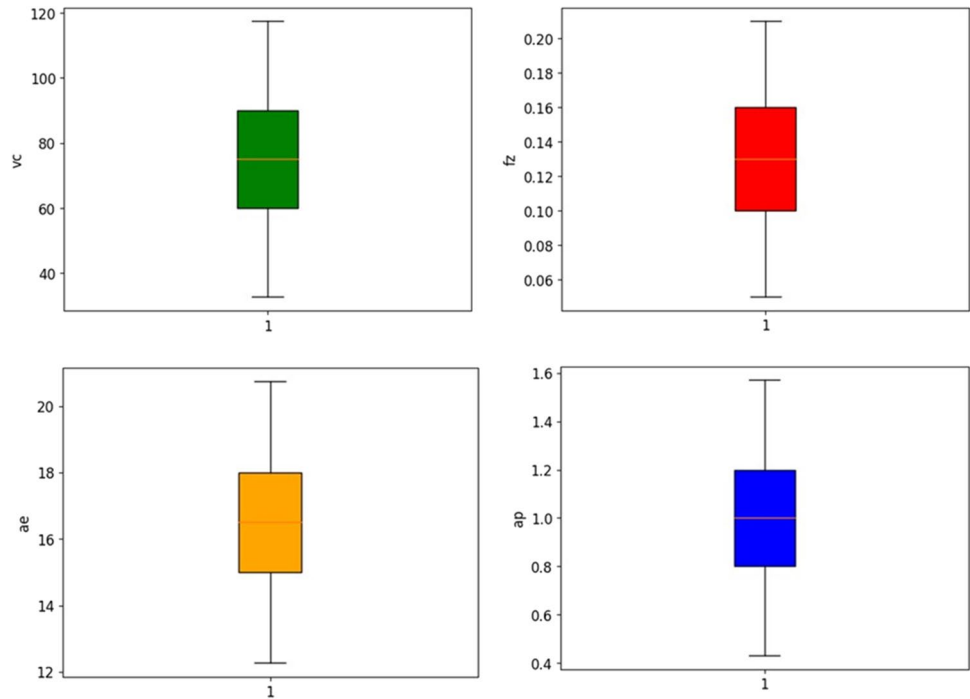


Fig. 8 General methodology used in this study

Fig. 9 Analysis of outliers



as an inherent component of random variability and do not exert a significant impact on model interpretation or performance.

4.2 Normality tests

One of the most important continuous distributions is the normal distribution. It describes the typical behavior of various phenomena and has great relevance in inferential statistics. This distribution directly affects the quality and reliability of the results in statistical analyses of scientific research

Table 6 Shapiro-Wilk test result

Model	Shapiro result	P-value
7-6-6-1	0.954	0.412
7-7-7-1	0.958	0.463
7-8-8-1	0.945	0.261
7-9-9-1	0.951	0.331
7-14-12-1	0.955	0.407
7-15-15-1	0.948	0.294
7-20-14-1	0.962	0.551
7-64-32-1	0.947	0.279

Fig. 10 Normality tests

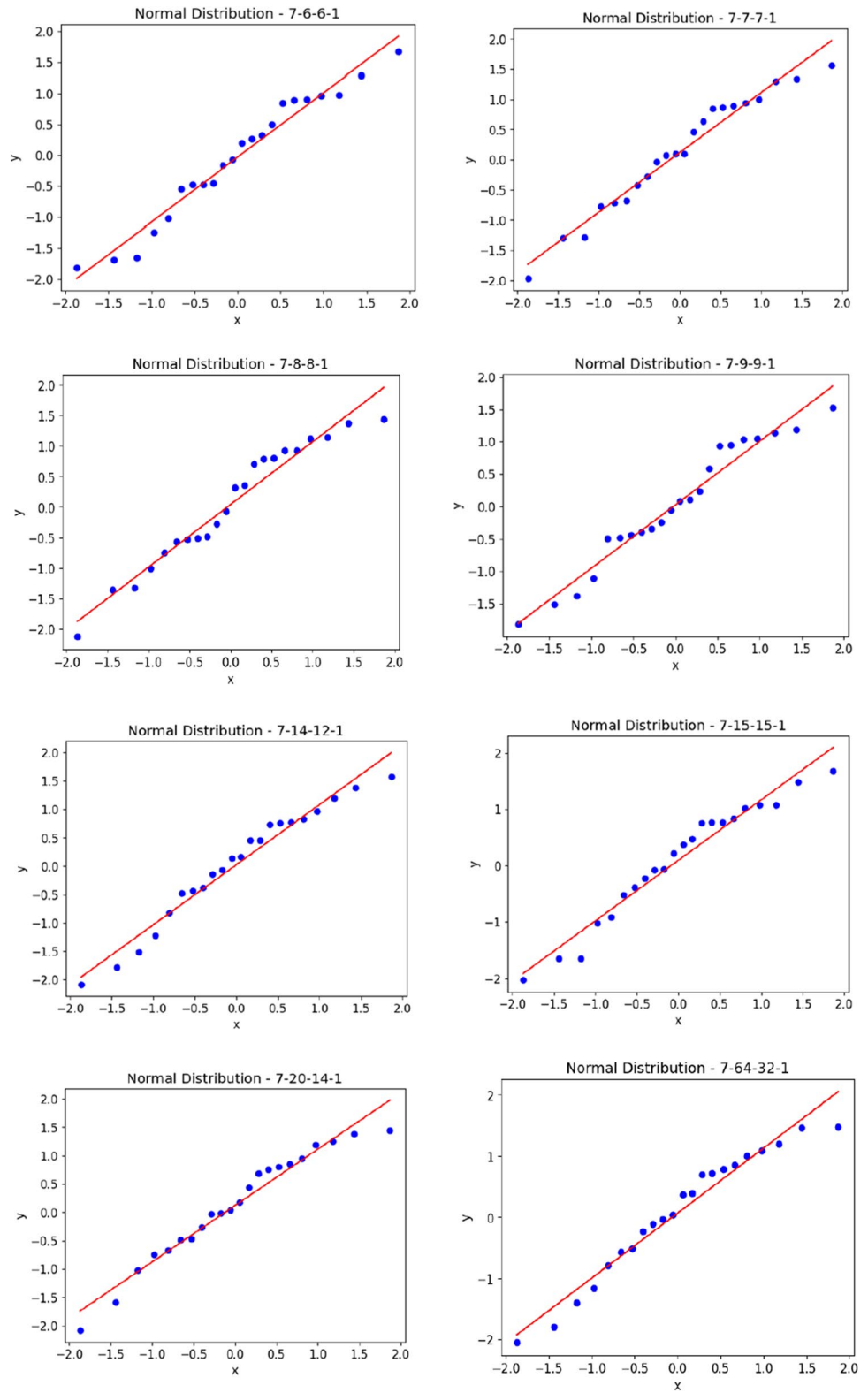


Fig. 11 Learning rate

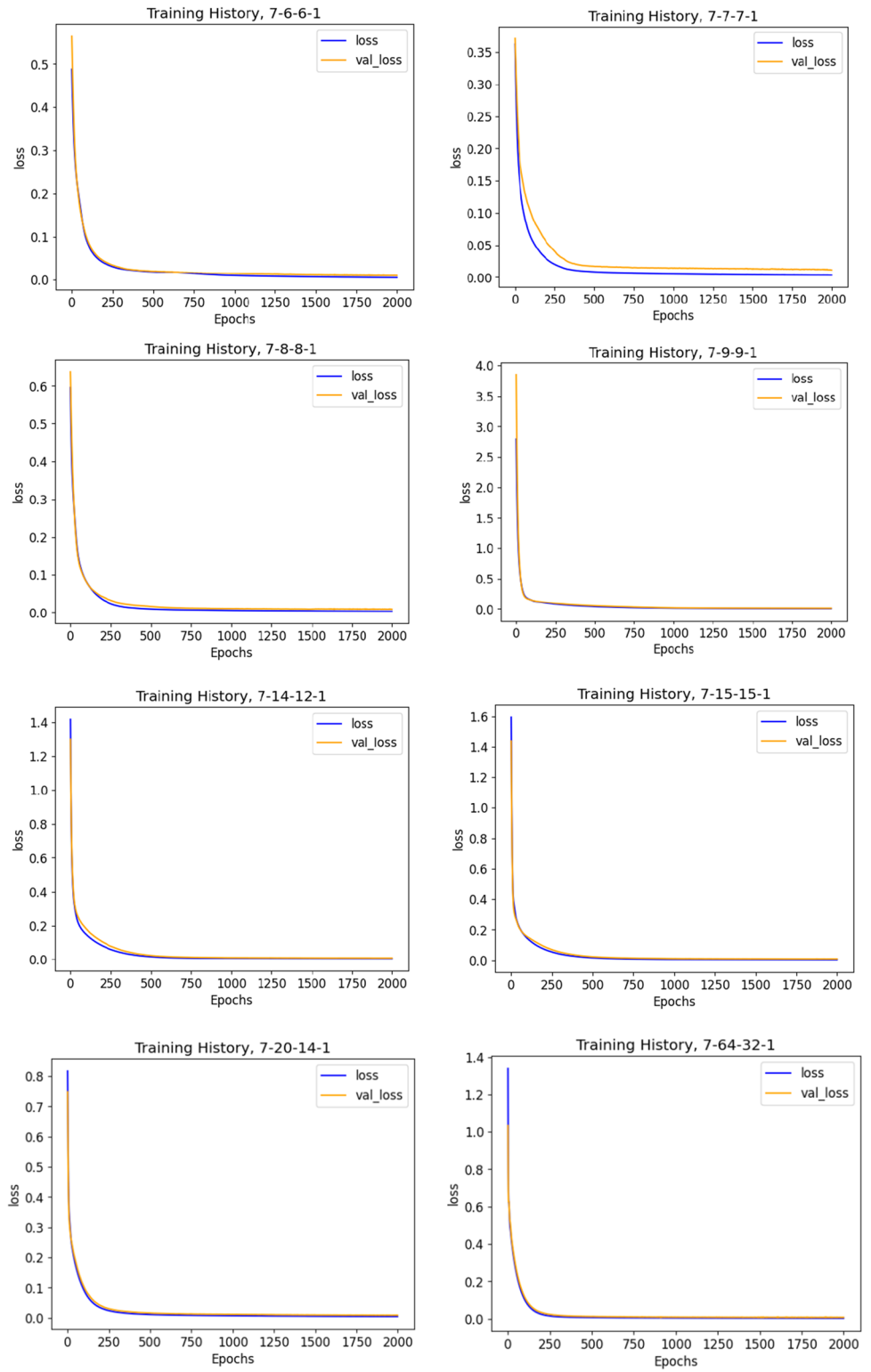


Table 7 Performance of neural network

Metrics	Configurations							
	7-6-6-1	7-7-7-1	7-8-8-1	7-9-9-1	7-14-12-1	7-15-15-1	7-20-14-1	7-64-32-1
MSE	0.012	0.006	0.006	0.006	0.004	0.005	0.003	0.004
RMSE	0.110	0.079	0.077	0.081	0.068	0.072	0.063	0.064
MAE	0.073	0.063	0.058	0.063	0.055	0.058	0.046	0.053
R^2	0.830	0.865	0.869	0.857	0.901	0.885	0.914	0.908
MAPE	0.11	0.10	0.092	0.095	0.081	0.086	0.076	0.078

that assume the normality of the data. Failure to confirm this assumption may result in inaccurate conclusions.

Therefore, the Shapiro-Wilk test was applied, which is one of the most recommended tests to test normality. It is a statistical tool used to verify the normality of the data. If the P -value is greater than or equal to 5%, your data can be considered approximately normal. However, if the P -value is less than 5%, you should consider that your data does not follow a normal distribution. This is important because many statistical methods, such as analysis of variance (ANOVA) and t -tests, assume data normality, and violation of this assumption can affect the interpretation of results [5]. Table 6; Fig. 10 show the results of the Shapiro-Wilk test for the eight network models. It is possible to observe that the sets of predictions generated by the networks tend to follow a normal distribution, since the means of the Shapiro-Wilk test and the P -value were between 0.945 and 0.962–0.261 and 0.551, respectively.

4.3 Learning rate

Critical indicators during the training and testing of a neural network employing the rectified linear unit (ReLU) activation function are the loss function (loss) and validation loss (val_loss). ReLU has been widely adopted in neural networks due to its non-saturation and non-linearity, providing significant advantages. Compared to activation functions that exhibit saturation, such as Sigmoid, ReLU is notably faster during training with gradient descent (Xu et al., 2020). Additionally, the simplicity in implementing the derivative of the ReLU neuron by applying a threshold to an activation matrix at zero stands out as an advantage over the sigmoid function.

The loss function reflects how effectively the model performs the desired task during training, while validation loss is associated with performance on a dataset not used during training. Evaluating the model's ability to generalize to unseen data is crucial. The ReLU activation function, with its non-linearities that accelerate training, requires careful monitoring of both loss and val_loss. The occurrence of low loss on the training set but high val_loss on the test

set suggests potential overfitting, indicating that the model is memorizing the training data instead of learning general patterns.

The training and test results, illustrated in Fig. 11, reveal the values of loss and val_loss for the created models: 0.0059/0.0108, 0.0039/0.0113, 0.0060/0.0115, 0.0031/0.0076, 0.004/0.0084, 0.0041/0.0096, 0.0043/0.0084, and 0.003/0.0076, respectively. These values indicate that the models are not prone to overfitting, providing good generalization to unseen data.

4.4 Predictive performance of neural network

Table 7 shows the performance of the eight neural network models in predicting Ra. Based on the results presented and considering the criterion for choosing the best network configuration based on the RMSE, we observed that the best network configuration is 7-20-14-1, followed by 7-64-32-1 and 7-14-12-1, since they obtained the lowest RMSE values, which were 0.063, 0.064, and 0.068, respectively. In addition, they also had the lowest MAE values, with results of 0.046, 0.053, and 0.055, respectively. The coefficient of determination R^2 was 0.914, 0.908, and 0.901, respectively. Figure 12 presents the graphs that show the relationship between the data predicted by the ANN (artificial neural network) and the output data experienced. This graph is generated using the predicted surface roughness values of the ANN structure in the test phase. When evaluating the graph, it can be summarized that the network structures showed a very similar line pattern between the ANN targets (Y_{Ra}) and the ANN outputs (Ra). Another relevant point is that even when the neural network makes a mistake in its prediction, the predicted value is remarkably similar. Table 8 represents the predicted values of the neural networks.

Based on the calculations of the mean absolute error (MAE) for the roughness analysis, it is evident that the value found, which is approximately 0.007875 microns, is quite small. This value can be considered practically negligible in the context of surface roughness, indicating that the observed values are extremely close to the actual average value of 0.688 microns. Therefore, we can

Fig. 12 Predict x experiment

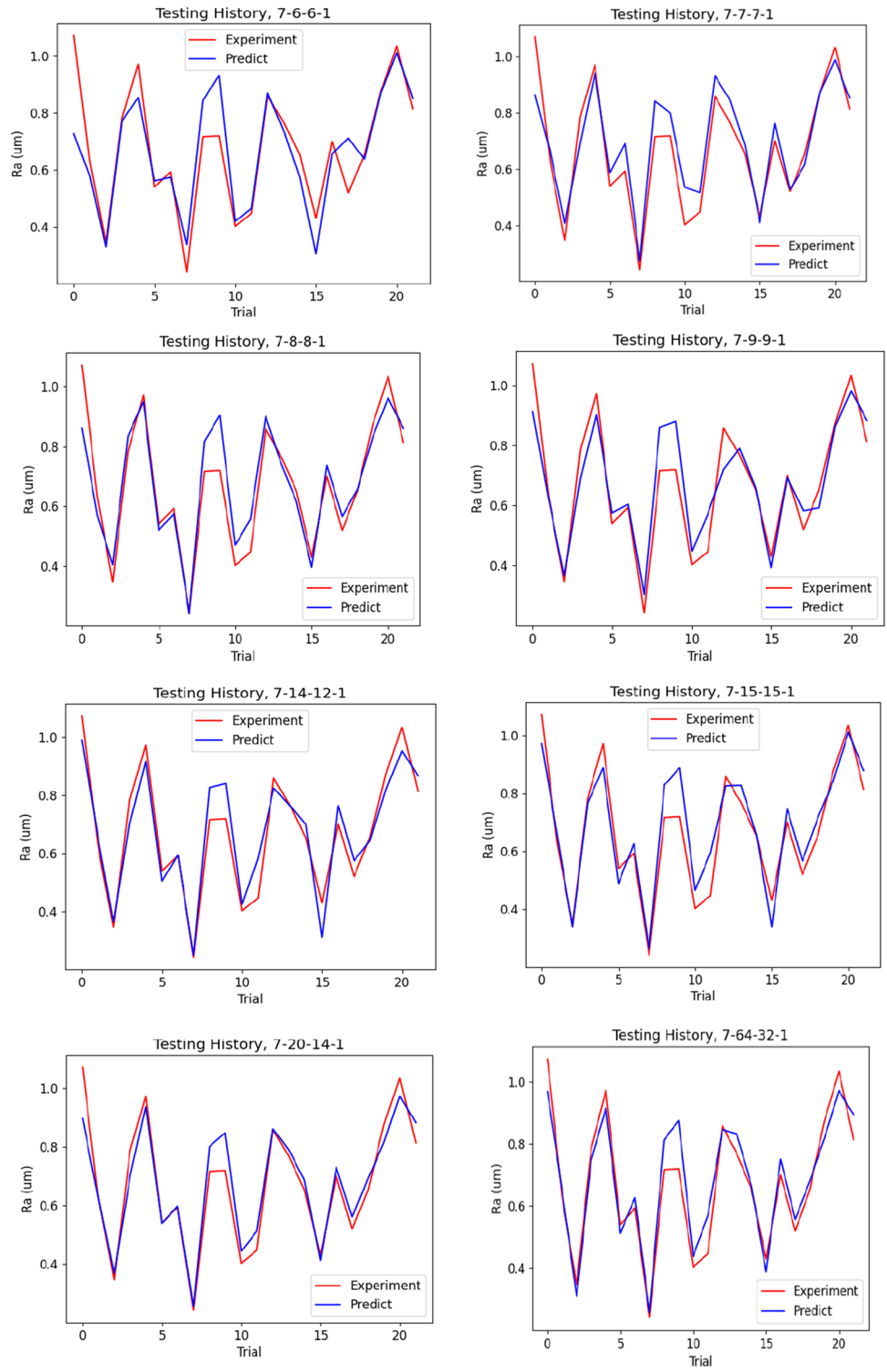


Table 8 Prevision of neural network

Ra (exp)	Ra (prev)							
	7-6-6-1	7-7-7-1	7-8-8-1	7-9-9-1	7-14-12-1	7-15-15-1	7-20-14-1	7-64-32-1
1.071	0.726	0.862	0.860	0.911	0.989	0.971	0.895	0.967
0.630	0.579	0.664	0.569	0.621	0.659	0.659	0.628	0.651
0.347	0.330	0.408	0.405	0.367	0.365	0.340	0.369	0.310
0.784	0.772	0.690	0.834	0.689	0.704	0.767	0.697	0.747
0.971	0.854	0.941	0.950	0.902	0.915	0.889	0.936	0.914
0.540	0.562	0.587	0.520	0.575	0.505	0.488	0.539	0.512
0.593	0.574	0.690	0.575	0.604	0.594	0.626	0.597	0.627
0.243	0.338	0.274	0.243	0.305	0.250	0.262	0.256	0.258
0.716	0.843	0.841	0.816	0.860	0.826	0.828	0.800	0.812
0.719	0.931	0.799	0.903	0.880	0.840	0.888	0.847	0.875
0.403	0.420	0.535	0.470	0.449	0.425	0.465	0.444	0.438
0.447	0.464	0.516	0.557	0.571	0.583	0.594	0.512	0.569
0.858	0.869	0.932	0.900	0.720	0.824	0.825	0.861	0.845
0.767	0.737	0.847	0.744	0.789	0.765	0.827	0.789	0.830
0.653	0.575	0.685	0.617	0.659	0.700	0.656	0.686	0.664
0.430	0.306	0.411	0.397	0.393	0.311	0.340	0.413	0.389
0.700	0.657	0.763	0.736	0.693	0.764	0.747	0.731	0.751
0.520	0.711	0.528	0.566	0.582	0.574	0.566	0.561	0.557
0.653	0.638	0.616	0.658	0.592	0.642	0.714	0.693	0.681
0.873	0.867	0.873	0.831	0.862	0.820	0.840	0.813	0.818
1.033	1.010	0.987	0.962	0.980	0.951	1.011	0.972	0.971
0.813	0.851	0.852	0.860	0.883	0.866	0.877	0.882	0.893
1.071	0.726	0.862	0.860	0.911	0.989	0.971	0.895	0.967
Mean exp	Mean prev							
0.688	0.667	0.703	0.688	0.687	0.690	0.702	0.688	0.698

conclude that, for the roughness analysis, the error found is insignificant and does not substantially affect the accuracy of the results, thus validating the agreement of the observed data with the actual mean value.

5 Conclusions

In this paper, we explore the use of artificial neural networks (ANN) as an approach for predicting surface roughness in milling operations. We demonstrate the effectiveness of this technique in modeling the machining process, emphasizing the ability to predict roughness measurements. We also highlight the importance of tuning the ANN architecture, specifically the number of layers and neurons in the hidden layers, to achieve high-quality predictions.

Our results indicate that it is possible to obtain accurate predictions of surface roughness, even when considering the inherent noises of the process and when working with relatively small training sets. Selecting the proper network configuration is essential to ensure the quality of forecasts. In addition, our research highlights

the relevance of considering noise when training ANN models, providing a more accurate understanding of how real processes work.

In summary, this study contributes significantly to the modeling of machining processes, with important implications for the manufacturing industry. It highlights the importance of considering noise when training ANN models and offers an innovative approach to predicting surface roughness. These results have practical relevance and can be applied in a variety of industrial applications.

Author contribution The work is divided into (i) literature review, (ii) background, (iii) statistical analysis, and (iv) writing and review. The work was divided as follows: Guilherme Vasconcelos: literature review, background, statistical analysis, and writing and review; Matheus Brendon: literature review, statistical analysis, and writing and review; Lucas Ribeiro: background, statistical analysis, and review; Ronny Junior: literature review, background, and review; and Miriam Motta: literature review, background, and review.

Funding I want to thank the Foundation for Research Support of the State of Minas Gerais (FAPEMIG) for the funding in projects APQ 02290-23 and APQ-03036-23.

Declarations

Conflict of interest The authors declare no competing interests.

References

- Marques DC, Suyama DI, Antunes RA, Hassui A (2023) Influence of machining parameters on tool wear, residual stresses, and corrosion resistance after milling super duplex stainless steel UNS S32750. *Int J Adv Manuf Technol* 129:801–814
- Phokobye SN, Desai DA, Tlhabadira I et al (2024) Comparative investigation and optimization of cutting tools performance during milling machining of titanium alloy (Ti6Al4V) using response surface methodology. *Int J Adv Manuf Technol*. <https://doi.org/10.1007/s00170-024-13225-3>
- Gowthaman PS, Jeyakumar S, Saravanan BA (2020) Machinability and wear mechanism of duplex stainless steel tools – a review. *Mater Today: Proc* 26:21423–1429. <https://doi.org/10.1016/j.matpr.2020.02.295>
- Zain AM, Haron H, Sharif S (2010) Prediction of surface roughness in the end milling machining using artificial neural network. *Expert Syst Appl* 37(2):1755–1768. <https://doi.org/10.1016/j.eswa.2009.07.033>
- Guo M, Xia W, Wu C et al (2024) A surface quality prediction model considering the machine-tool-material interactions. *Int J Adv Manuf Technol*. <https://doi.org/10.1007/s00170-024-13072-2>
- Thangarasu SS, Mohanraj T, Devendran K (2019) Tool wear prediction in hard turning of EN8 steel using cutting force and surface roughness with artificial neural network. *Proc Inst Mech Eng Part C J Mech Eng Sci* 234:329–342
- Yeganefar A, Niknam SA, Asadi R (2019) The use of support vector machine, neural network, and regression analysis to predict and optimize surface roughness and cutting forces in milling. *Int J Adv Manuf Technol* 105:951–965. <https://doi.org/10.1007/s00170-019-04227-7>
- Huang Z, Zhu J, Lei J et al (2020) Tool wear predicting based on multi-domain feature fusion by deep convolutional neural network in milling operations. *J Intell Manuf* 31:953–966. <https://doi.org/10.1007/s10845-019-01488-7>
- Wang J, Zou B, Liu M et al (2021) Milling force prediction model based on transfer learning and neural network. *J Intell Manuf* 32:947–956. <https://doi.org/10.1007/s10845-020-01595-w>
- Chen C-H, Jeng S-Y, Lin C-J (2022) Prediction and analysis of the Surface roughness in CNC end milling using neural networks. *Appl Sci* 12:393. <https://doi.org/10.3390/app12010393>
- Xie J, Zhao P, Hu P et al (2021) Multi-objective feed rate optimization of three-axis rough milling based on artificial neural network. *Int J Adv Manuf Technol* 114:1323–1339. <https://doi.org/10.1007/s00170-021-06902-0>
- Rodrigues A, Silva FJG, Sousa VFC, Pinto AG, Ferreira LP, Pereira T (2022) Using an artificial neural network approach to predict machining time. *Metals* 12(10):1709. <https://doi.org/10.3390/met12101709>
- Sharma R, Jha BK, Pahuja V (2022) Application of RSM and ANN for the predication and optimization of the circularity error of DSS 2205 under hybrid cryo-MQL process. *Journal of Xi'an Shiyou University, Natural Science Edition* (June, 2022)
- Sivarajan S, Elango M, Sasikumar M, Doss Arockia Selvakumar Arockia (2022) Prediction of surface roughness in hard machining of EN31 steel with TiAlN coated cutting tool using fuzzy logic. *Mater Today: Proc* 65:35–41. <https://doi.org/10.1016/j.matpr.2022.04.161>
- Arunadevi M, Shreeram PB, Kumar T, Gowda K, Deepika UM (2023) Performance enhancement of CNC milling process using different machine learning techniques. *J Mines Met Fuels* 71(2):149–156
- Outemsa O, El Farissi O, Hamouti L (2022) Optimization of cutting parameters and prediction of surface roughness in turning of duplex stainless steel (DSS) using a BPNN and GA. *Int J Tech Phys Probl Eng (IJTPE)* 14(2):234–239
- Knap P, Jachymczyk U, Balazy P (2023) Tool wear detection in milling processes using long short-term memory networks: an industry 4.0 approach machines. *Technol Mater* 17(4):148–151
- Ponnusamy P, Tamilperuvalathan S (2023) Performance evaluation and prediction based on a hybrid deep recurrent neural network of SS304 stainless steel properties using water-emulsified MQL with added nanoparticles. *Biomass Bioenergy Convention* 13:7349–7373. <https://doi.org/10.1007/s13399-023-04106-y>
- Kumar R et al (2023) Application of artificial neural network for prediction of MRR and surface finish in milling operation. *AIP Conference Proceedings*, Vol. 2535. No. 1. AIP Publishing
- Cheng Y-N, Jin Y-B, Gai X-Y, Guan R, Lu M-D (2023) Prediction of tool wear in milling process based on BP neural network optimized by firefly algorithm. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*. <https://doi.org/10.1177/09544089231160492>
- Bai L, Cheng X, Yang Q et al (2023) Predictive model of surface roughness in milling of 7075Al based on vibration stability analysis and backpropagation neural network. *Int J Adv Manuf Technol* 126:1347–1361. <https://doi.org/10.1007/s00170-023-11133-6>
- Montgomery DC (2017) *Designs and analysis of experiments*, 9th edn. Wiley, USA
- McCulloch W, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. *Bull Math Biophys* 5:115–133
- Akinwekomi AD, Lawal AI (2021) Neural network-based model for predicting particle size of AZ61 powder during high-energy mechanical milling. *Neural Comput Appl* 33:17611–17619
- ROSENBLATT F (1958) The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol Rev* 65(6):386–408
- Minsky ML, Papert SA, *Perceptrons* (1969) MIT Press, Cambridge, MA
- Haykin S (2001) *Neural networks - principles and practices*, 2nd edn. Bookman, São Paulo, p 900
- Machado W, Fonseca C, Júnior ES (2013) Artificial neural networks applied to the prediction of VTEC in Brazil. *Bulletin of Geodetic Sciences* 19(2):227–246
- Dhobale N, Mulik S, Jegdeeshwaran R, Ganer K (2021) Multipoint milling tool supervision using artificial neural network approach. *Mater Today: Proc* 45(2):1898–1903
- Chai, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE) arguments against avoiding RMSE in the literature. *Geosci Model Dev* 7(3):1247–1250
- Ramírez-Gallego S, Krawczyk B, García S, Woźniak M, Herrera F (2017) A survey on data preprocessing for data stream mining: current status and future directions. *Neurocomputing* 239:39–57
- Boukerche A, Zheng L, Alfandi OD (2020) Outlier detection: methods, models, and classification. *Comput ACM Survive* 53:1–37
- Nguyen N-T, Tien DH, Tung NT, Luan ND (2021) Analysis of tool wear and surface roughness in high-speed milling process of aluminum alloy Al6061. *EUREKA: Phys Eng*, (3), 71–84
- Wu X-Y, Chen H, Zhang L-D, Xiong H, Lei, Deng S-H (2019) Hyperparameter optimization for machine learning models based on bayesian optimization. *J Electron Sci Technol* 17:26–40
- Injadat M, Moubayed A, Nassif AB, Shami A (2020) Systematic ensemble model selection approach for educational data mining. *Knowl-Based Syst* 200:105992

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.