# Prediction of surface roughness in the end milling machining using Artificial Neural Network

**Article** *in* Expert Systems with Applications · March 2010

DOI: 10.1016/j.eswa.2009.07.033 · Source: DBLP



**3 authors:**

Azlan Mohd Zain
Universiti Teknologi Malaysia
**123** PUBLICATIONS **2,556** CITATIONS

SEE PROFILE

Habibollah Haron
Universiti Teknologi Malaysia
**195** PUBLICATIONS **2,613** CITATIONS

SEE PROFILE

Safian Sharif
Universiti Teknologi Malaysia
**255** PUBLICATIONS **4,103** CITATIONS

SEE PROFILE

Expert Systems with Applications 37 (2010) 1755–1768

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

# Prediction of surface roughness in the end milling machining using Artificial Neural Network

Azlan Mohd Zain [a,*], Habibollah Haron [a], Safian Sharif [b]

[a] Faculty of Computer Science & Information System, Universiti Teknologi Malaysia, 81310 Skudai Johor, Malaysia
[b] Faculty of Mechanical Engineering, Universiti Teknologi Malaysia, 81310 Skudai Johor, Malaysia

ARTICLE INFO

*Keywords:*
ANN
Modelling
Predicting surface roughness

ABSTRACT

This paper presents the ANN model for predicting the surface roughness performance measure in the machining process by considering the Artificial Neural Network (ANN) as the essential technique for measuring surface roughness. A revision of several previous studies associated with the modelling issue is carried out to assess how capable ANN is as a technique to model the problem. Based on the studies conducted by previous researchers, the abilities and limitations of the ANN technique for predicting surface roughness are highlighted. Utilization of ANN-based modelling is also discussed to show the required basic elements for predicting surface roughness in the milling process. In order to investigate how capable the ANN technique is at estimating the prediction value for surface roughness, a real machining experiment is referred to in this study. In the experiment, 24 samples of data concerned with the milling operation are collected based on eight samples of data of a two-level DOE $2^k$ full factorial analysis, four samples of centre data, and 12 samples of axial data. All data samples are tested in real machining by using uncoated, TiAlN coated and $SN_{TR}$ coated cutting tools of titanium alloy (Ti-6A1-4V). The Matlab ANN toolbox is used for the modelling purpose with some justifications. Feedforward backpropagation is selected as the algorithm with *traingdx*, *learngdx*, *MSE*, *logsig* as the training, learning, performance and transfer functions, respectively. With three nodes in the input layer and one node in the output layer, eight networks are developed by using different numbers of nodes in the hidden layer which are 3–1–1, 3–3–1, 3–6–1, 3–7–1, 3–1–1–1, 3–3–3–1, 3–6–6–1 and 3–7–7–1 structures. It was found that the 3–1–1 network structure of the $SN_{TR}$ coated cutting tool gave the best ANN model in predicting the surface roughness value. This study concludes that the model for surface roughness in the milling process could be improved by modifying the number of layers and nodes in the hidden layers of the ANN network structure, particularly for predicting the value of the surface roughness performance measure. As a result of the prediction, the recommended combination of cutting conditions to obtain the best surface roughness value is a high speed with a low feed rate and radial rake angle.



## 1. Introduction

Surface finish is an important factor in evaluating the quality of products. Surface roughness ($R_a$) is mostly used as an index to determine the surface finish in the machining process. Modelling techniques for the prediction of $R_a$ can be classified into three groups which are experimental models, analytical models and Artificial Intelligence (AI)-based models (Benardos & Vosnaikos, 2003). Experimental and analytical models can be developed by using conventional approaches such as the Statistical Regression technique. On the other hand, AI-based models are developed using non-conventional approaches such as the Artificial Neural Network (ANN), Fuzzy Logic (FL) and Genetic Algorithm (GA) (Brezocnik, Kavocic, & Ficko, 2004; Cus & Zuperl, 2006; Dweiri, Al-Jarrah, & Al-Wedyan, 2003).

Recently, AI-based models have become the preferred trend and these are applied by most researchers to develop a model for near optimal conditions in machining. It is also considered as a successful approach to modelling the machining process for predicting performance measures through the development of an expert system. An expert system is an interactive intelligence program with an expert-like performance in solving a particular type of problem using knowledge base, inference engine and user interface. A model based on ANN is able to learn, adapt to changes and mimic the human thought process with little human interaction. The FL model deals with linguistic variables rather than operating on crisp values. The GA model, meanwhile, involves the coding of solution states in chromosomes as a series of binary elements zero and one. Similar to the conventional approaches which consist of

* Corresponding author. Tel.: +60 7 5532088; fax: +60 7 5565044/5574908.
*E-mail addresses:* azlanmz@utm.my, azlanmz6674@gmail.com (A.M. Zain).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.07.033

various numbers of alternative techniques such as Respond Surface Methodology (RSM) and Taguchi, AI also provides alternative techniques in modelling as mentioned above. Different techniques may be suitable for particular modelling problems in the machining process and may not be suitable for other ones. Thus, this study investigates how capable ANN is as a potential technique to be applied with the $R_a$ modelling problem.

The machining process is very complex and does not permit pure analytical physical modelling (Luttervelt, Childs, Jawahir, Klocke, & Venuvinod, 1998). Thus, experimental and analytical models, or those called explicit (empirical) models that are developed by using conventional approaches such as the Statistical Regression technique which is combined with the Response Surface Methodology (RSM), have remained as an alternative in the modelling of the machining process. Although the Statistical Regression technique may work well for modelling, this technique may not describe precisely the underlying nonlinear complex relationship between machining parameters and performance measures (Mukherjee & Ray, 2006). With reference to the published literature, the reviews show that milling is one of the most cutting processes in machining besides turning (Benardos & Vosnaikos, 2003). $R_a$ is one of the performance measures employed mostly by the researchers using the ANN technique, but most of the work carried out is related to the turning process and it was found that there is a very limited amount of work which deals with the milling process. The ability of ANN to be a good modelling technique for $R_a$ prediction was mentioned by Tsai, Chen, and Luu (1999) where the ANN model gave a high accuracy rate (96–99%) for predicting $R_a$ in the end milling cutting operations compared to the result of the Statistical Regression model. Erzurumlu and Oktem (2007) concluded that the ANN model led to slightly more accurate $R_a$ prediction values compared to the conventional model.

With the very limited resources available for the problem investigated in this study, namely, the influence of the $R_a$ value in the end milling of titanium alloy, the review also found that no study has yet focused on the issue of the radial rake angle cutting conditions applied with the ANN modelling technique. This issue could be assumed to be the contribution of this study to the machining area. With the consideration that the AI approach is able to handle the machine modelling problem, this paper outlines an understanding of how the ANN technique operates to develop the model for the prediction of $R_a$ in the milling process.

## 2. Previous studies on ANN for $R_a$ prediction

In order to see how capable ANN is as the technique for modelling the machining process, several studies related to the modelling issue are reviewed. Table 1 shows the previous studies that have dealt with the $R_a$ prediction model. The contents of the table include the machining process involved, the network structure applied, the decision variable affecting the $R_a$ value prediction and remarks that indicate how capable ANN is at modelling the machining process.

Previous studies listed in Table 1 obviously indicate that ANN has been successfully applied by researchers for modelling $R_a$ prediction in the machining process. In terms of the machining process for the $R_a$ prediction model, ANN has been applied mostly for turning and there are a very limited number of studies on the milling process. The decision variables that most influenced the $R_a$ prediction values by using the ANN modelling technique are cutting speed, feed rate and depth of cut. The remarks column indicates that ANN mostly gave a better $R_a$ prediction result compared to the result obtained with the real experiment.

Based on the previous studies listed, the ability and limitation of the ANN technique for the $R_a$ prediction values could be summarized. By looking at previous studies (Al-Ahmari, 2007; Davim, Gaitonde, & Karmik, 2008; Hans, Swarup, Srivastava, & Patvardhan, 2000; Kafkas, Karatas, Sozen, Arcaklioglu, & Saritas, 2007; Nalbant, Gokkaya, Toktas, & Sur, 2009; Sheikh-Ahmad & Twomey, 2007; Zuperl & Cus, 2003; Zuperl, Cus, Mursec, & Ploj, 2006) the abilities of ANN for modelling the machining process may include the following:

(i) ANN is able to handle a nonlinear form of modelling that learns the mapping of inputs to outputs.
(ii) ANN is more successful, when compared to conventional approaches, in terms of speed, simplicity and capacity to learn from examples, and also does not require much experimental data.

**Table 1**
Application of ANN in modelling related to $R_a$ prediction.

| | Author, Year | Process | Cutting conditions | Remarks |
|---|---|---|---|---|
| 1 | Abeesh et al. (2008) | Turning | Feed rate, particle size, tool nose radius | The predicted responses of the ANN model are in very good agreement with the experimental data |
| 2 | Erzurumlu and Oktem (2007) | Milling | Feed rate, cutting speed, axial–radial depth of cut, machining tolerance | The ANN model leads to a slightly more accurate surface roughness prediction than a conventional model |
| 3 | Davim et al. (2008) | Turning | Feed rate, cutting speed, depth of cut | ANN can capture any degree of non-linearity that exists between the process response and input parameters, and exhibits good generalization |
| 4 | Nalbant et al. (2009) | Turning | Coating tools, feed rate, cutting speed | ANN is very close to the experiment-based results with an acceptable accuracy |
| 5 | Al-Ahmari (2007) | Turning | Cutting speed, feed rate, depth of cut, tool nose radius | The ANN model is better than linear regression analysis techniques and RSM for predicting tool life and cutting force models |
| 6 | Sanjay and Jyothi (2006) | Drilling | Drill diameter, cutting speed, feed rate, machining time | ANN has shown that it is capable of generalization and is effective in surface roughness analysis |
| 7 | Cus and Zuperl (2006) | Turning | Cutting speed, feed rate, depth of cut | ANN provides a sufficient approximation to the true optimal solution |
| 8 | Kohli and Dixit (2005) | Turning | Cutting speed, feed rate, depth of cut, radial vibration | ANN is able to make an accurate prediction of surface roughness by utilizing a small-sized training and testing dataset |
| 9 | Ezugwu et al. (2005) | Turning | Cutting speed, feed rate, depth of cut, cutting time, coolant pressure | ANN gives a very good agreement between predicted and experimentally measured process parameters |
| 10 | Grzesick and Brol (2003) | Turning | Feed rate, spindle speed, depth of cut, cutting time, cutting force | ANN gives a close agreement between the real data and predicted values of surface finish within ±10% of the experimental values |
| 11 | Zuperl and Cus (2003) | Turning | Cutting speed, feed rate, depth of cut | ANN provides a sufficient approximation to the true optimal solution |
| 12 | Tansel et al. (2006) | Milling | Cutting speed, feed rate, radial depth of cut | The ANN model represents the data well and has a very small error for the training cases |
| 13 | Oktem et al. (2006) | Milling | Cutting speed, feed rate, axial–radial depth of cut, machining tolerance | ANN presents a very good performance for surface roughness response value |

(iii) The ANN model does not need any preliminary assumptions as to the underlying mechanisms in the modelled process.
(iv) Improvements in the behaviour of the experimental results are easy to understand in a short time from the neuronal model in ANN.
(v) The performance of the ANN prediction model could be further improved by defining more levels for the input process parameters that can be achieved by trial-and-error methods and repeated training simulation.
(vi) There exists the ANN toolbox software package such as MATLAB which can easily be used for training and testing the machining data modelled.
(vii) ANN allows for simple complementing of the model by new input parameters without modifying the existing model structures.
(viii) Researchers have the choice to use and compare different training algorithms such as BP and RB in ANN to obtain more accurate results of the prediction model.

The limitations of ANN in the machining process modelling could be stated as follows (Davim et al., 2008; Zuperl et al., 2006):

(i) Empirical experiences are necessary in creating a realistic network.
(ii) It is costly and time consuming for further improved by defining more levels for the input process parameters that can be achieved by the trial-and-error method and repeated training simulation.
(iii) Repeatability of training for a further improved model is not assured.

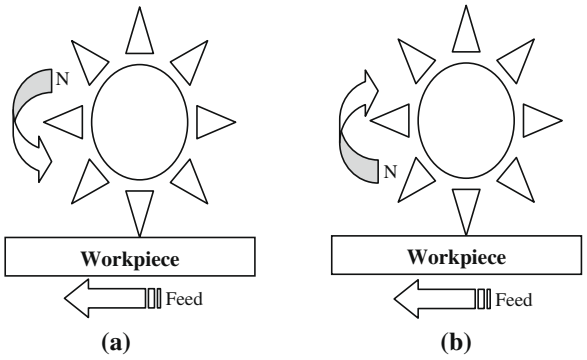


**Fig. 1.** (a) Up milling surface generation; (b) Down milling surface generation (Kalpakjian, 1991).

By considering the abilities and limitations of ANN for the modelling purpose, the focus of this study is stated as being related to the item (v) of the capabilities of ANN where the best prediction model for surface roughness could be achieved by the trial-and-error method, and item (iii) of the limitations of ANN where the trial and repeatability of ANN training for the improved model is not assured. Therefore, considering the concept of the trial-and-error method with modifying the ANN model structure, it is expected that improving the result of surface roughness is carried out in this study.

## 3. Surface roughness in the milling process

In the milling process, material is removed from the workpiece by a rotating cutter. The milling process can be classified into two parts; peripheral milling and face milling. Peripheral milling generates a surface parallel to the spindle rotation, while face milling generates a surface normal to the spindle rotation. End milling is a type of face milling, and is used for facing, profiling and slotting processes. The surface can be generated by two methods; up milling and down milling. Up milling is also called conventional milling; the cutter rotates against the direction of feed of the workpiece as is shown in Fig. 1a. Down milling is also called climb milling; the rotation is in the same direction as the feed as is shown in Fig. 1b.

Typically all surfaces have their own characteristics, which are referred to as surface texture. Surface texture is the pattern of the surface which deviates from a nominal surface. The deviations may be repetitive or random and may result from roughness, flaws and waviness. Therefore, the actual surface profile is the superposition of error of form, waviness and roughness. Surface roughness is defined as a closely spaced, irregular deviation on a scale smaller than waviness. Fig. 2 shows the standard terminology and symbols to describe surface roughness (Brezocnik et al., 2004).

In machining, surface roughness is generally specified mathematically in terms of the arithmetic average deviation from the mean using Eq. (1),

$$R_a = \frac{1}{L}\int_0^L |Y(x)|dx \tag{1}$$

where $L$ is the sampling length and $Y$ is the ordinate of the profile curve. In other words, $R_a$ is the area between the roughness profile and its mean line in μm, or the integral of the absolute profile height over the evaluation length which needs to be optimized as shown in Fig. 2. Generally, the $R_a$ value is influenced by many factors such as machining parameters, cutting phenomena, workpiece properties and cutting tool properties (Benardos & Vosnaikos, 2003). Cutting speed, feed rate and depth of cut are parameters that

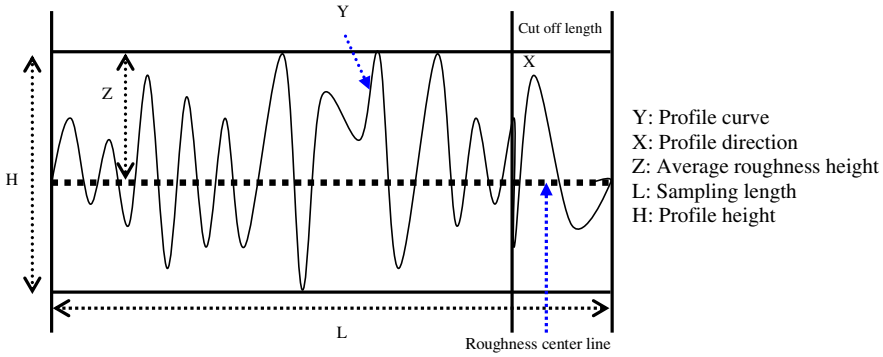


**Fig. 2.** Surface roughness profile (Yang & Chen, 2001).

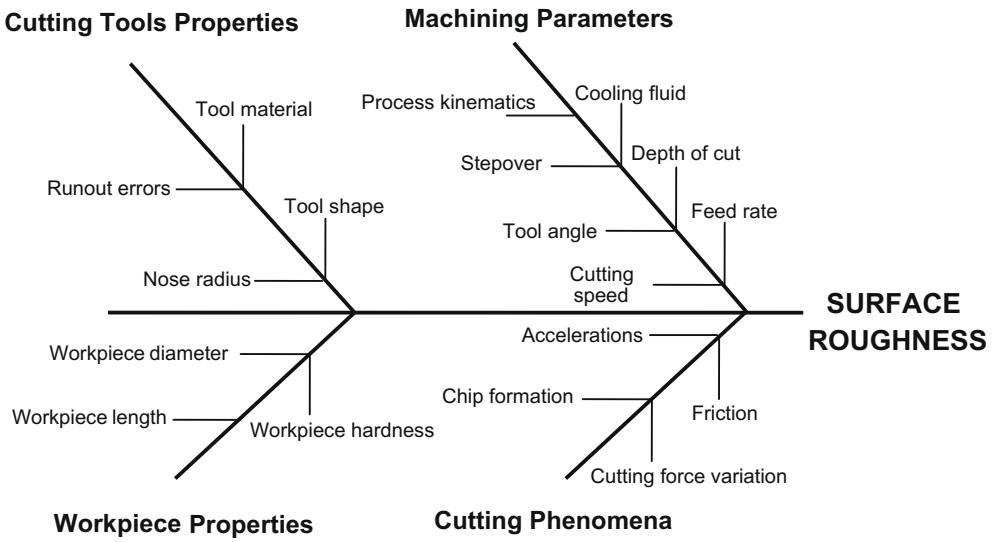


**Fig. 3.** Parameters that affect surface roughness (Benardos & Vosnaikos, 2003).

mostly influence the $R_a$ value of surface quality in machining, particularly in the milling machining process. The set of parameters that are thought to influence the $R_a$ value could be summarized in a fishbone diagram as shown in Fig. 3.

As shown in Fig. 3, tool angle is one of the main machining parameters that affect surface roughness. Rake angle is one of the most important tools that affect surface roughness performance measure measurement. There are various rake angles in machining such as the axial rake angle, the radial rake angle, and the side rake angle. By looking at previous studies, it was found that the application of ANN as a surface roughness prediction model in the end milling of titanium alloy involving radial rake angle machining parameters is still not given consideration by researchers. Therefore, the radial rake angle machining parameter is taken into consideration. This study is interested in observing the effect of this parameter in influencing the surface roughness prediction result when using the ANN technique.

## 4. $R_a$ modelling and the case study for the investigation of the problem

A model developed for the machining process is the relationship between two variables which are the decision variable (input parameter) and the response variable (output parameter) in terms of mathematical equations. Modelling can also be described as a scientific way to study the behaviours involved in the process. The modelling of machining processes is important for providing the basis mathematical model for the formulation of the objective function. Therefore, the minimization of the $R_a$ value must be formulated in the standard mathematical model. Normally, the model of the predicted $R_a$ value for the milling process in relation to the decision variables investigated can be expressed by Eq. (2)

$$\hat{R}_a = k v^{x_1} f^{x_2} a_a^{x_3} a_r^{x_4} m^{x_5} \tag{2}$$

In Eq. (2), $\hat{R}_a$ is the predicted surface roughness in μm, $v$ is the cutting speed, $f$ is the feed per tooth, $a_a$ is the axial depth of cut,$a_r$ is the radial depth of cut, $m_t$ is machining tolerance and $k$, $x_1, x_2, x_3, x_4$, and $x_5$ are the model parameters to be estimated using the experimental data. Then, all the parameters that affect the predicted $\hat{R}_a$ value obtained from the modelling phase must be optimized by using conventional or non-conventional approaches in the optimizing phase.

The $R_a$ value is expected to be as low as possible and could be achieved by adjusting the value of the decision variables with the assistance of an appropriate numerical optimization method. For example, if an attempt is made to optimize the value of the decision variables for $R_a$ which are given in Eq. (2), the optimization objective function could be written as Eq. (3) (Oktem, Erzurumlu, & Erzincanli, 2006; Zuperl & Cus, 2003),

$$R_a = \frac{1}{L}\int_0^L |Y(x)|dx = \min f(x_1, x_2, x_3, x_4, x_5) \tag{3}$$

where $x_1 = v_c, x_2 = f_t, x_3 = a_a, x_4 = a_r, x_5 = m_t \cdot v_c$ is the cutting speed, $f_t$ is the feed rate, $a_a$ is the axial depth of cut, $a_r$ is the radial depth of cut and $m_t$ is machining tolerance. $v_c, f_t, a_a, a_r, m_t$ are called values of coefficients. The objective function is principally to minimize surface roughness in order to define the standard optimization. Limitations for the coefficient values of the objective function for Eq. (3) are given in Eqs. (4a)–(4e)

$$v_{c\text{-}min} \leqslant v_c \leqslant v_{c\text{-}max} \tag{4a}$$

$$f_{t\text{-}min} \leqslant f_t \leqslant f_{t\text{-}max} \tag{4b}$$

$$a_{a\text{-}min} \leqslant a_a \leqslant a_{a\text{-}max} \tag{4c}$$

$$a_{r\text{-}min} \leqslant a_r \leqslant a_{r\text{-}max} \tag{4d}$$

$$m_{t\text{-}min} \leqslant m_t \leqslant m_{t\text{-}max} \tag{4e}$$

In order to investigate how capable the ANN technique is in predicting the surface roughness value the work of Mohruni (2008) is referred to as a case study. The work relates to the development of a mathematical model for surface roughness in the end milling of titanium alloy (Ti-6A1-4V) using uncoated solid carbide under flood conditions. The factorial design of the experiment integrated with the regression technique was used in developing the surface roughness models in relation to the primary machining decision variables which are cutting speed, feed rate and radial rake angle. RSM technique was applied in determining the optimum cutting conditions for a particular range of surface roughness values. The adequacy of the predictive models was verified by ANOVA. The selected decision variables or cutting conditions of the experiment for surface roughness, with their limits, units and notations, are given in Table 2.

To handle the experiment, 24 samples of data are collected based on eight samples of data of a two-level DOE $2^k$ full factorial analysis, four samples of centre data, and 12 samples of axial data.

**Table 2**
Setting of cutting condition values for real machining experiments.

| Decision variables | Notation | Units | Limits | | | | |
|---|---|---|---|---|---|---|---|
| | | | −1.4142 | −1 | 0 | +1 | +1.4142 |
| Cutting speed | $V$ | m/min | 124.53 | 130.00 | 144.22 | 160.00 | 167.03 |
| Feed rate | $F$ | mm/tooth | 0.025 | 0.03 | 0.046 | 0.07 | 0.083 |
| Radial rake angle | $\gamma$ | ° | 6.2 | 7.0 | 9.5 | 13.0 | 14.8 |

All the samples are tested in real machining for three different cutting tools which are uncoated, TiAlN coated and $SN_{TR}$ coated cutting tools to show the actual value (experiment result) of $R_a$. The $R_a$ value of each type of cutting tool which has been observed is shown in Table 3.

## 5. Justification of ANN for $R_a$ modelling

Principally, to obtain the successful model of ANN, it totally depends on the process of trial and error with some factors to consider. The application of the ANN model for the modelling purpose in various different areas including machining is used very widely by researchers; however, until now there have been no clear rules that could serve as a basis to be followed by this study in producing the perfect model. Due to this fact, namely that the effectiveness of the ANN model developed is fully dependent on the trial-and-error process in some factors, this study considers the factors which could be influencing the effectiveness of the ANN model developed based on the item required by the Matlab ANN toolbox in order to develop the ANN model.

Based on the ANN toolbox of Matlab software, five influencing factors are as follows:

(i) Network algorithm.
(ii) Transfer function.
(iii) Training function.
(iv) Learning function.
(v) Performance function.

Four other factors are also considered that can influence the effectiveness of the model and these are:

(i) Network structure.
(ii) Number of training data.
(iii) Number of testing data.
(iv) Normalization of data input.

To justify selecting ANN as the methodology that depends on the process of trial and error with some considerations to give the best predictive result, Table 4 shows the example item or value used by several previous studies for the nine factors listed above in order to develop the ANN model for $R_a$ value prediction in the machining cutting process.

### 5.1. Network structure

An ANN network structure principally consists of layers and nodes. Nodes are also known as neurons. An illustration of an ANN network with layers and nodes is given in Fig. 4; this is known as an implicit model.

Based on Fig. 4, the example of an ANN network structure consists of three layers which are the input layer, hidden layer and output layer. It is also possible to have an ANN structure with no hidden layers. The network structure has three nodes in the input layer, $j$ nodes in the first hidden layer, $k$ nodes in the second hidden layer, $l$ nodes in the $m$th hidden layer and one node in the output layer. Three nodes for the input layer stand for the three decision values of the case study which are cutting speed ($v$), feed

**Table 3**
$R_a$ values for real machining experiments.

| No. | Data source | Setting values of process parameters | | | $R_a$ value (μm) | | |
|---|---|---|---|---|---|---|---|
| | | $V$ (m/min) | $F$ (mm/tooth) | $\gamma$ (°) | Uncoated | TiAlN coated | $SN_{TR}$ coated |
| 1 | DOE $2^k$ | 130 | 0.03 | 7 | 0.365 | 0.32 | 0.284 |
| 2 | | 160 | 0.03 | 7 | 0.256 | 0.266 | 0.196 |
| 3 | | 130 | 0.07 | 7 | 0.498 | 0.606 | 0.668 |
| 4 | | 160 | 0.07 | 7 | 0.464 | 0.476 | 0.624 |
| 5 | | 130 | 0.03 | 13 | 0.428 | 0.26 | 0.28 |
| 6 | | 160 | 0.03 | 13 | 0.252 | 0.232 | 0.19 |
| 7 | | 130 | 0.07 | 13 | 0.561 | 0.412 | 0.612 |
| 8 | | 160 | 0.07 | 13 | 0.512 | 0.392 | 0.576 |
| 9 | Centre | 144.22 | 0.046 | 9.5 | 0.464 | 0.324 | 0.329 |
| 10 | | 144.22 | 0.046 | 9.5 | 0.444 | 0.38 | 0.416 |
| 11 | | 144.22 | 0.046 | 9.5 | 0.448 | 0.46 | 0.352 |
| 12 | | 144.22 | 0.046 | 9.5 | 0.424 | 0.304 | 0.4 |
| 13 | Axial | 124.53 | 0.046 | 9.5 | 0.328 | 0.36 | 0.344 |
| 14 | | 124.53 | 0.046 | 9.5 | 0.324 | 0.308 | 0.32 |
| 15 | | 167.03 | 0.046 | 9.5 | 0.236 | 0.34 | 0.272 |
| 16 | | 167.03 | 0.046 | 9.5 | 0.24 | 0.356 | 0.288 |
| 17 | | 144.22 | 0.025 | 9.5 | 0.252 | 0.308 | 0.23 |
| 18 | | 144.22 | 0.025 | 9.5 | 0.262 | 0.328 | 0.234 |
| 19 | | 144.22 | 0.083 | 9.5 | 0.584 | 0.656 | 0.64 |
| 20 | | 144.22 | 0.083 | 9.5 | 0.656 | 0.584 | 0.696 |
| 21 | | 144.22 | 0.046 | 6.2 | 0.304 | 0.3 | 0.361 |
| 22 | | 144.22 | 0.046 | 6.2 | 0.288 | 0.316 | 0.36 |
| 23 | | 144.22 | 0.046 | 14.8 | 0.316 | 0.324 | 0.368 |
| 24 | | 144.22 | 0.046 | 14.8 | 0.348 | 0.396 | 0.36 |

**Table 4**
Influencing factors of ANN modelling for $R_a$ prediction.

| | Author, Year | Network struc. | # Train. data | # Test data | Norm. input | Network algorithm | Trans. func. | Train. func. | Learn. func. | Perf. func. |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Abeesh et al. (2008) | 5–8–1 | Not stated | Not stated | Not stated | Feedforward BP | tansig: tansig | trainbr & trainlm | learngd | MAE |
| 2 | Erzurumlu and Oktem (2007) | 5–42–42–1 | 243 | 7 | Not stated | Feedforward BP | tansig: tansig | traingda, $\eta = 0.0001$ | learngd | MSE |
| 3 | Davim et al. (2008) | 3–16–2 | 27 | 3 | Not stated | Feedforward BP | Not given | traingdx, $\eta = 0.05, \alpha = 0.85$ | learngdm | MSE |
| 4 | Nalbant et al. (2009) | 3–9–1 | 55 | 5 | Not stated | Feedforward BP | logsig: logsig | training & trainlm | learng & learangd | RMSE |
| 5 | Al-Ahmari (2007) | 4–73–45–3 | 28 | Not stated | Not stated | Feedforward BP | tansig: purelin | trainscg | learngd | SSE |
| 6 | Sanjay and Jyothi (2006) | 4–1–1, 4–5–1, 4–10–1, 4–15–1, 4–20–1 | Not stated | Not stated | Given | Feedforward BP | logsig: logsig | traingda, $\eta = 0.01$ | learngd | Not stated |
| 7 | Cus and Zuperl (2006) | 3–3–6–1 | 20 | 10 | Not stated | Feedforward BP&RB | tansig: tansig | traingda | learngd | MAE |
| 8 | Kohli and Dixit (2005) | 4–3–1 | 21 | 10 | Not stated | Feedforward BP | logsig: logsig | traingda | learngd | RMSE |
| 9 | Ezugwu et al. (2005) | 4–10–10–1, 4–15–15–1, 4–10–15–1, 4–15–10–1 | Not stated | Not stated | Given | Feedforward BP | tansig: purelin | trainbr & trainlm | learngd | Not stated |
| 10 | Grzesick and Brol (2003) | 7–72–72–72–7 | Not stated | Not stated | Not stated | Feedforward BP | logsig: logsig | traingdx, $\eta = 0.4, \alpha = 0.5$ | learngd | MSE |
| 11 | Zuperl and Cus (2003) | 3–3–6–1 | 20 | 20 | Not stated | Feedforward BP&RB | logsig: logsig | traingdx, $\eta = 0.3, \alpha = 0.004$ | learngd | MAE |
| 12 | Tansel et al. (2006) | Not stated | 81 | Not stated | Not stated | Feedforward BP | logsig: logsig | traingdx | Not stated | Not stated |
| 13 | Oktem et al. (2006) | 5–42–42–1 | 236 | 7 | Not stated | Feedforward BP | tansig: tansig | traingda, $\eta = 0.0001$ | learngd | MSE |

rate ($f$), and rake angle ($\gamma$). One node for the output layer stands for the predicted surface roughness value of the case study. When considering that a multilayer feedforward network is applied at the $m$th hidden layer with $j$, $k$ and $l$ nodes for each hidden layer, the example network given in Fig. 4 could be defined as a 3–$j$–$k$–l–1 structure.

From Table 4 it can be seen that the researchers have applied different structures for the model developed. For example, the work of Abeesh, Dabade, Joshi, Bhanuprasad, and Gadre (2008) used the very basic, 5–8–1 structure, meaning that it has five nodes for the input layer, eight nodes for hidden layers, and one node for the output layer. Grzesick and Brol (2003) applied the 7–72–72–72–7 structure meaning that it has three hidden layers with 72 nodes for each layer. Sanjay and Jyothi (2006), for example, applied five different structures, 4–1–1, 4–5–1, 4–10–1, 4–15–1, and 4–20–1, to predict the surface roughness value. The result showed

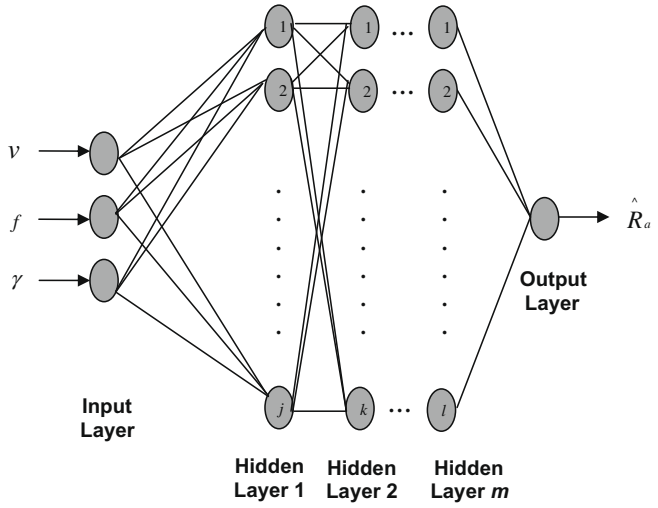


**Fig. 4.** Example illustration of an ANN network structure with layers and node.

that the 4–1–1 network structure is more accurate and reliable for the prediction of the surface roughness performance measure.

From Table 4, it could be summarized that researchers have tried different model structures in order to get the best prediction result. In other words, the ANN model is designed on a trial-and-error basis to obtain the best result. The process of trial and error is carried out by adjusting the number of layers and the number of nodes of hidden layer(s) of the network structure. Researchers are free to test at any number of hidden layers with any number of nodes for each hidden layer. However, the number of hidden layers and the nodes in each hidden layer area are subject to the complexity of the mapping, computer memory, computation time and the desired data control effect. Too many nodes result in a waste of computer memory and computation time, while too few nodes may not provide the desired data control effect (Al-Ahmari, 2007). Due to this statement the following question relating to the network structure could be raised: "*What ANN network structure should be applied to give the best prediction result?*"

In relation to this study, a possible answer for the question above is to adjust the hidden structure of the ANN network. As a suggested solution, this study prefers to apply different network structures and compare the results by following the guidelines given by Zhang, Patuwo, and Hu (1998) where the recommended number of nodes for the hidden layer are "$n/2$", "$1n$", "$2n$", and "$2n + 1$" where $n$ is the number of input nodes. Since the number of decision variables is three for the case study, the recommended number of nodes in the hidden layer are $(3)/2 = 1.5 \approx 1$, $1(3) = 3$, $2(3) = 6$, and $2(3) + 1 = 7$. Therefore, by limiting the trial-and-error process with two hidden layers, this study applies eight network structures, which are 3–1–1, 3–3–1, 3–6–1, 3–7–1, 3–1–1–1, 3–3–3–1, 3–6–6–1 and 3–7–7–1, as described in Figs. 5 and 6.

### 5.2. Amount of training and testing data

An issue arising from previous studies given in Table 4 which could be highlighted is the amount of training and testing data used by researchers. With reference to the concept of ANN, an increment in the amount of training data will increase the chance of getting a

more accurate model. In machining, data used to be trained is taken from the actual experimental trials. Many constraints such as the cost and time used in conducting the actual experiment might be a problem for researchers to get more data for the modelling purpose. With reference to Table 4, it can be seen that the amount of training and testing samples of data collected by researchers varies from as low as 20 samples of data for training and 10 samples for testing in developing machining models using ANN in relation to predicting the value of surface roughness. Cus and Zuperl (2006), for example, have obtained an accurate result for predicting surface roughness with a small amount of training and testing samples. Since the sample size for this case study is 24 that could be classified as a small amount of data. The question to be asked is: "*Could an accurate predictive result be obtained with the small amount of training and testing data available for this experiment?*"

The possible answer is could be, since it was proven by previous work. By referring to Table 4, it can be seen that several studies have been conducted using a small amount of modelling samples (less than 30 when training and testing are combined), and ANN presented a good performance for the response value for surface roughness. Therefore, with a total experimental sample size of 24 available, by applying the ANN model this study is expected to give an accurate predictive result for surface roughness.

### 5.3. Ratio of training and testing data

Based on Table 4, it can be seen that the number of testing samples used by researchers are various and it is smaller than the number of training samples. The issue here is to separate the available experimental samples into training and testing. Basically, there is

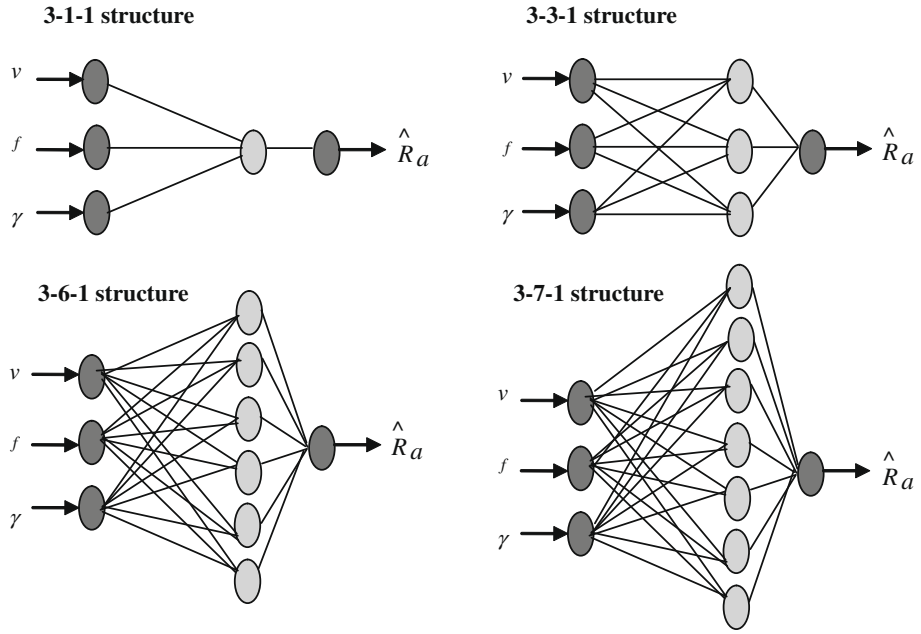


**Fig. 5.** Network structure model with a single hidden layer.

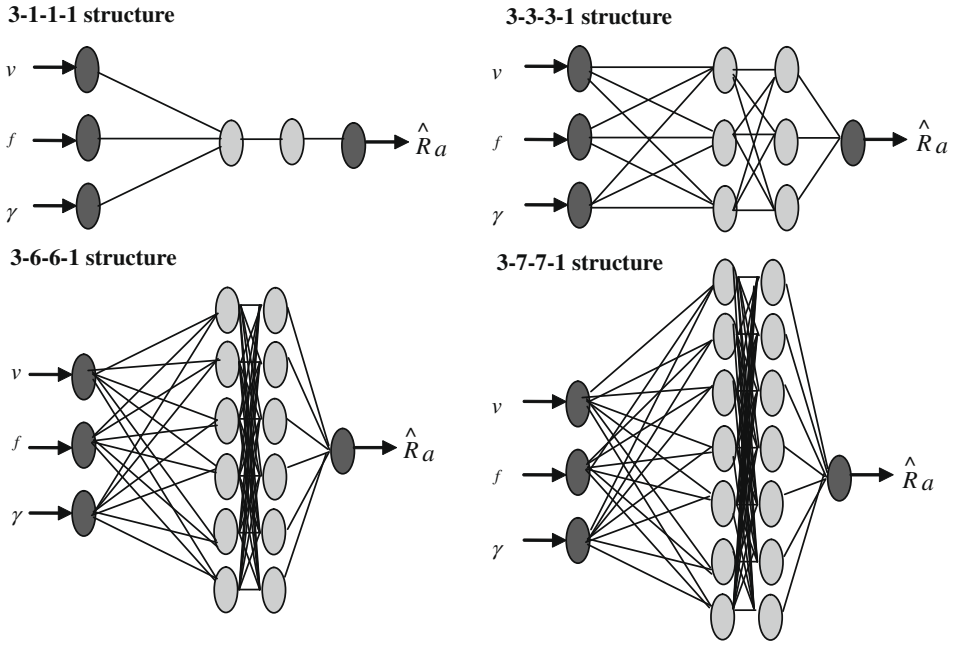


**Fig. 6.** Network structure model with two hidden layers.

no general guideline which could be followed to measure the ratio between the amount of training and the amount of testing samples. In relation to this issue, the question to be asked is: "*With the available experimental sample size of 24 in the case study how much data should be collected for training and how much for testing?*"

The possible answer is self-determination, as long as the amount of training samples is more than the amount of testing samples. A suggested solution is to follow the guidelines given by Zhang et al. (1998) where the recommended ratio of training and testing samples could be given as percent, such as 90%:10%, 85%:15% and 80%:20% with a total of 100% for the combined ratio. To fit in with the available experimental sample size of 24, the preferred ratio is selected as 70%:30%. So the recommended amount of training and testing samples is:

1. $(85/100) \times 24 = 16$–$17$ training samples,
2. $(15/100) \times 24 = 7$–$8$ data testing samples.

As is given in Table 3, with 24 available experimental samples, the separate samples for training and testing are suggested as follows:

1. Four centre sample sets (ninth to 12th experimental sample) and 12 axial sample sets (13th to last experimental samples) with a total of 16 sample sets being chosen as training samples.
2. DOE $2^k$ samples (the first eight experimental samples) with total of eight sample sets being chosen as testing samples.

### 5.4. Normalization of data input/output

Data normalization is often performed before the training and testing process begins. It is possible to normalize the quantitative variable to some standard range such as 0 to 1 or −1 to 1. Referring to Table 4, two authors discuss the requirement for normalizing data (Ezugwu, Fadare, Bonneya, Silva, & Sales, 2005; Sanjay & Jyothi, 2006). Basically, when nonlinear transfer functions such as the logistic sigmoid function are used at the output nodes, the desired output values must be transformed to the range of the actual outputs of the network. Even if a linear output transfer function is used, it may still be advantageous to standardize the outputs as well as the inputs to avoid computational problems. ANN is applied for the modelling process in this study, where speed, feed rate and rake angle are the inputs, and surface roughness is the output, and the values of input and output are given in Table 3. Here, the first step in developing the model is to normalize all the raw data. In relation to the issue about the normalization of data input/output, the question to be asked is "*How is normalization of the input/output data made with the available experimental sample size of 24 for the case study?*"

To normalize the raw data of input and output, two potential normalization equations to be used are:

$$x_i = \frac{2}{d_{\max} - d_{\min}}(d_i - d_{\min}) - 1 \tag{5}$$

$$x_i = \frac{0.8}{d_{\max} - d_{\min}}(d_i - d_{\min}) + 0.1 \tag{6}$$

In Eq. (5) by Ezugwu et al. (2005) and Eq. (6) by Sanjay and Jyothi (2006), $d_{\max}$ is the maximum value of the input/output data, $d_{\min}$ is the minimum value of the input/output data, and $d_i$ is the $i$th input/output data. As a suggested solution, Eq. (6) is chosen for the normalization of the input/output data for this study.

### 5.5. Network algorithm

Many different ANN network algorithms have been proposed by researchers for the modelling purpose such as Cascade-forward BP, Elman BP, Time-delay BP, Perceptron, Radial Basis, and Self-Organizing Map. Looking at Table 4, it can be seen that the feedforward backpropagation (BP) algorithm is mostly applied by researchers for predicting surface roughness. In related to the ANN network algorithm, the question to be asked is: "*What ANN algorithm should be applied to give the best prediction result?*"

There are several choices for a possible answer, but the review showed that the feedforward BP network algorithm is the most widely used by researchers. Zuperl and Cus (2003), for example, have developed the model by using the feedforward BP and radial basis network algorithm, and it was found that the feedforward BP gave the more accurate results, but it required more time for training and testing. The radial basis network is preferred since it is very fast and reliable but it is worse in terms of the accuracy of the prediction result for the surface roughness response variable. Therefore, the BP algorithm is recommended for this study.

Basically, a feedforward network based on BP is a multilayered architecture made up of one or more hidden layers placed between the input and output layers. Each layer consists of units which recover their input from units a layer directly below and send their output to units a layer directly above the unit. Fig. 7 illustrates an example of a multilayer feedforward ANN structure.

### 5.6. Transfer function, training function, learning function and performance function

Based on Fig. 7, and considering that the multilayer feedforward training network with one hidden layer is applied, the net input to unit $k$ in the hidden layer is expressed in Eq. (7),

$$net\_hidden = \sum_{j=1}^{J} C_{j,k} i_j + \theta_k \tag{7}$$

where $C_{j,k}$ is the weight between the input neurons and hidden neurons, $i_j$ is the value of the input which consists of speed, feed rate and rake angle of the experimental sample, and $\theta_k$ is the biases on the hidden nodes. The net input to unit $z$ in the output layer is expressed in Eq. (8),

$$net\_output = \sum_{k=1}^{K} D_{k,z} h_k + \phi_z \tag{8}$$

where $D_{k,z}$ is the weight between hidden and output neurons, $h_k$ is the value of the output for hidden nodes, and $\phi_z$ is the biases on the output nodes.

From Eqs. (7) and (8), the output for hidden nodes can be given as Eq. (9), and the output for output nodes can be given as Eq. (10),

$$h_k = f(net\_hidden) \tag{9}$$

$$o_z = f(net\_output) = \hat{R}_a \tag{10}$$

where $f$ is the transfer function. By referring to Table 4, it can be seen that the previous studies applied different transfer functions for predicting surface roughness. There are some transfer functions that could be applied, such as log-sigmoid transfer function (*logsig*), linear transfer function (*purelin*), hyperbolic tangent sigmoid transfer function (*tansig*), and hard limit transfer function (*hardlim*). Therefore, in relation to the choice of the transfer function for the multilayer feedforward training network, the question to be asked is: "*What transfer function should be applied to give the best prediction result?*"

There are several choices for a possible answer, and there is no clear statement given by previous studies as to whether different transfer functions have major effects on the performance of the networks. By referring to the User's Guide for Neural Network Toolbox 6 written by Demuth, Beale, and Hagan (2008), three transfer functions most commonly used with the feedforward BP algorithm are the *logsig*, *tansig* and *linear* transfer functions.

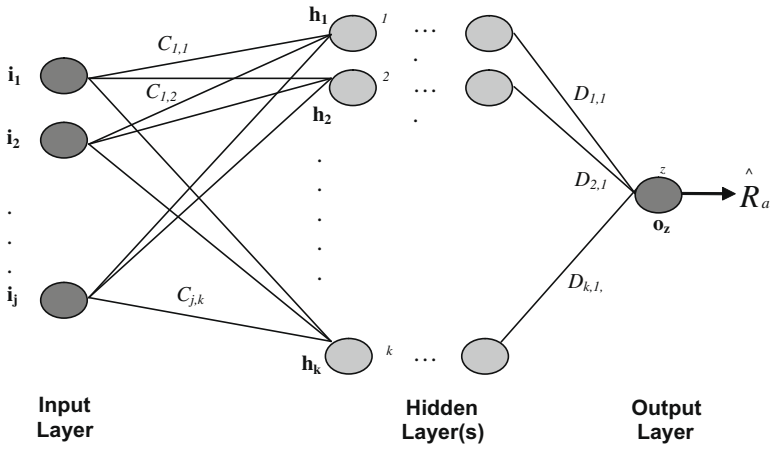


**Fig. 7.** Multilayer feedforward ANN structure.

Nalbant et al. (2009) concluded that the determination of transfer function depends on the nature of the problem. The use of a nonlinear transfer function makes a network capable of a nonlinear relationship between the input and the output. A commonly used function is the *sigmoid* function because it is self-limiting and has a simple derivative. An advantage of the *sigmoid* function is that the output cannot grow infinitely large or small. Kohli and Dixit (2005), for example, have applied two different transfer functions, *logsig* and *tansig*, and it was observed that both these transfer functions produced almost the same performance. It is recommended that the *logsig* transfer function is applied in this study since it is mostly applied by researchers as is shown in Table 4. It is used in determining the output for hidden and output nodes as given in Eqs. (9) and (10). The *logsig* (log-sigmoid) transfer function is basically written as Eq. (11),

$$f = \frac{1}{(1+e^{-net})} \tag{11}$$

where *net* could be either the net of hidden nodes (Eq. (7)) with its output (Eq. (9)) or the net of output nodes (Eq. (8)) with its output (Eq. (10)). Therefore, the output for hidden nodes with the *sigmoid* function could be written as Eq. (12). The output nodes with the *sigmoid* function could be written as Eq. (13)

$$h_k = f(net\_input) = \frac{1}{1+e^{-\sum_{j=1}^{J} C_{j,k} i_j + \theta_k}} \tag{12}$$

$$o_z = f(net\_output) = \frac{1}{1+e^{-\sum_{k=1}^{K} D_{k,z} h_k + \phi_z}} = \hat{R}_a \tag{13}$$

With reference to the $\hat{R}_a$ value as illustrated in Fig. 7 which was obtained by using Eq. (13), it is noticed that this value possible contains the error. The error value is obtained by using Eq. (14),

$$error = ERR = \frac{1}{2}(R_a - \hat{R}_a)^2 \tag{14}$$

where $R_a$ is the surface roughness value of experimental (ANN target), and $\hat{R}_a$ is predicted surface roughness value of the ANN model (ANN output). Therefore, the error measure (*ERR*) must be reduced as low as possible by applying the BP algorithm. The error given in Eq. (14) is easily determined by MATLAB software which is known as a performance function. By referring to Table 4, it can be seen that the mean square error (MSE) performance function is mostly applied by researchers. Some other performance functions are mean absolute error (MAE), sum square error (SSE), root mean squared error (RMSE) and mean absolute percentage error (MAPE). In relation to the selection of the performance function, the question to be asked is: "*What performance functions should be applied to give a low error rate in the predicted value?*"

There are several choices for a possible answer, and just as with the issue of transfer function, there is no clear statement given by previous studies as to whether different performance functions have major effects on the performance of the networks. Since most of the previous studies applied the *MSE* performance function, it is therefore recommended that this performance function is applied for determining the error in the predicted value for surface roughness as is given in Eq. (14). Basically, the BP algorithm is applied in the multilayer feedforward ANN structure to reduce the error presented in Eq. (14). The BP algorithm is applied to find the lowest value of error possible. The BP algorithm generally proceeds as follows: firstly, inputs are presented to the network and the error is calculated; secondly, sensitivities are propagated from the output layer to the first layer; then, weights and biases are updated. The weights of the connections between input and hidden nodes, $C_{j,k,}$ and biases on the hidden nodes, $\theta_k$, are updated by Eqs. (15) and (16), respectively

$$\Delta C_{j,k} = -\eta \frac{\partial ERR}{\partial C_{j,k}} \tag{15}$$

$$\Delta \theta_k = -\eta \frac{\partial ERR}{\partial \theta_k} \tag{16}$$

The weights of the connection between hidden and output nodes, $D_{k,z,}$ and biases on the output nodes, $\phi_z$, are updated by Eqs. (17) and (18), respectively

$$\Delta D_{k,z} = -\eta \frac{\partial ERR}{\partial D_{k,z}} \tag{17}$$

$$\Delta \phi_z = -\eta \frac{\partial ERR}{\partial \phi_z} \tag{18}$$

In Eqs. (15)–(18), $\eta$ is the learning rate which should be selected to be as small as possible for a true approximation and, at the same time, as large as possible to speed up convergence. Smaller learning rates tend to slow the learning process while larger learning rates may cause network oscillation in the weight space. An option to allow for larger learning rates resulting in faster convergence while minimizing the tendency to oscillation is to include an additional momentum parameter, denoted by the $\alpha$ symbol. The effect of the momentum factor for the updated weights is given in Eqs. (19) and (20)

$$\Delta C_{j,k} = -\eta \frac{\partial ERR}{\partial C_{j,k}} + \alpha \Delta C_{j,k} \tag{19}$$

$$\Delta D_{k,z} = -\eta \frac{\partial ERR}{\partial D_{k,z}} + \alpha \Delta D_{k,z} \tag{20}$$

The process of updating the value of weights and biases of the feedforward multilayer network based on the BP algorithm could easily be made with the assistance of the Matlab ANN toolbox. To reduce the value of error based on the BP algorithm, two important parts in the Matlab ANN toolbox need to be considered; these are the training function and the learning function. Referring to Table 4, there are various training functions that have been applied by the previous studies, such as *trainbr* (Bayesian regularization), *traingd* (gradient descent BP), *traingda* (gradient descent with adaptive learning rule BP), *traingdm* (gradient descent with momentum BP), traingdx (gradient descent with momentum and adaptive learning rule BP), and *trainlm* (Levenberg–Marquardt BP). On the other hand,, examples of learning functions that are shown in Table 4 are *learngd* (gradient descent weight/bias learning function), *learngdm* (gradient descent with momentum weight/bias learning function), etc. In relation to the choice of the performance function, the question to be asked is: "*What training function and learning function should be applied to give a low error rate in the response value for the ANN model developed?*"

There are several choices for a possible answer, and just as with the issue of the performance function, there is no clear statement given by previous studies as to whether different training and learning functions have major effects on the performance of the networks. Since, most previous studies applied the *traingdx* training function and *learngd* learning function as is shown in Table 4, it is therefore suggested that these two functions be applied in this study. To get a successful model, the learning rate $(\eta)$ and momentum $(\alpha)$ where the value is in the range of 0–1 will be determined during the training process by the assistance of the Matlab ANN tool box.

## 6. Development of the ANN prediction model

The modelling process is conducted to produce the outputs which are the set values of the predicted surface roughness. The prediction value of $R_a$ is expected to give the better result compared to the result of the real machining experiment. The justification for ANN as the methodology for this study to develop the $R_a$

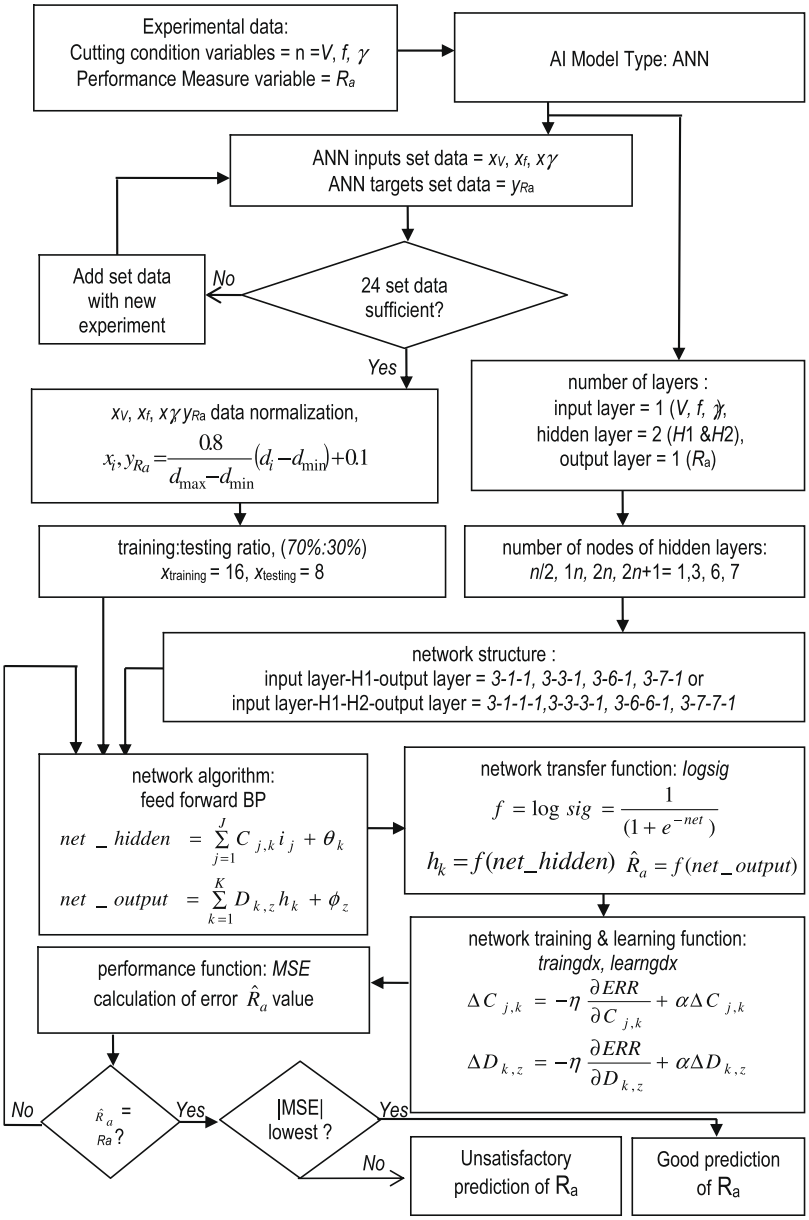


**Fig. 8.** Flow of the ANN-based model development for the $R_a$ response prediction.

**Table 5**
Normalized values of ANN inputs and targets.

| No. | Data source | ANN inputs (normalized values of cutting condition) | | | ANN targets (normalized values of experimental surface roughness) | | |
|---|---|---|---|---|---|---|---|
| | | $x_v$ | $x_f$ | $x_\gamma$ | $y_{R_a}$ uncoated | $y_{R_a}$ TiAlN | $y_{R_a}$ SNTR |
| 1 | DOE $2^k$ | 0.203 | 0.169 | 0.174 | 0.346 | 0.266 | 0.249 |
| 2 | | 0.768 | 0.169 | 0.174 | 0.138 | 0.164 | 0.109 |
| 3 | | 0.203 | 0.721 | 0.174 | 0.599 | 0.806 | 0.856 |
| 4 | | 0.768 | 0.721 | 0.174 | 0.534 | 0.560 | 0.786 |
| 5 | | 0.203 | 0.169 | 0.733 | 0.466 | 0.153 | 0.242 |
| 6 | | 0.768 | 0.169 | 0.733 | 0.130 | 0.100 | 0.100 |
| 7 | | 0.203 | 0.721 | 0.733 | 0.719 | 0.440 | 0.767 |
| 8 | | 0.768 | 0.721 | 0.733 | 0.626 | 0.402 | 0.710 |
| 9 | Centre | 0.471 | 0.390 | 0.407 | 0.534 | 0.274 | 0.320 |
| 10 | | 0.471 | 0.390 | 0.407 | 0.496 | 0.379 | 0.457 |
| 11 | | 0.471 | 0.390 | 0.407 | 0.504 | 0.530 | 0.356 |
| 12 | | 0.471 | 0.390 | 0.407 | 0.458 | 0.236 | 0.432 |
| 13 | Axial | 0.100 | 0.390 | 0.407 | 0.275 | 0.342 | 0.343 |
| 14 | | 0.100 | 0.390 | 0.407 | 0.268 | 0.243 | 0.306 |
| 15 | | 0.900 | 0.390 | 0.407 | 0.100 | 0.304 | 0.230 |
| 16 | | 0.900 | 0.390 | 0.407 | 0.108 | 0.334 | 0.255 |
| 17 | | 0.471 | 0.100 | 0.407 | 0.130 | 0.243 | 0.163 |
| 18 | | 0.471 | 0.100 | 0.407 | 0.150 | 0.281 | 0.170 |
| 19 | | 0.471 | 0.900 | 0.407 | 0.763 | 0.900 | 0.811 |
| 20 | | 0.471 | 0.900 | 0.407 | 0.900 | 0.764 | 0.900 |
| 21 | | 0.471 | 0.390 | 0.100 | 0.230 | 0.228 | 0.370 |
| 22 | | 0.471 | 0.390 | 0.100 | 0.199 | 0.258 | 0.369 |
| 23 | | 0.471 | 0.390 | 0.900 | 0.252 | 0.274 | 0.381 |
| 24 | | 0.471 | 0.390 | 0.900 | 0.313 | 0.409 | 0.369 |

prediction model could be summarized in Fig. 8. Since the important step in developing the model is to normalize all the raw input and output data, by using the normalization formula shown in Eq. (6), the normalized values of the cutting condition (ANN inputs) and $R_a$ of the experiment (ANN targets) are given in Table 5.

With the learning rate = 0.01, and momentum rate=0.05, the modelling result of $\hat{R}_a$ and the MSE value of $\hat{R}_a$ are generated by using the ANN toolbox of the MATLAB software. The justification for separating the experimental sample sets into trained and tested is mentioned in Section 5.3. With four centre sample sets and twelve axial sample sets, with a total of sixteen sample sets selected as training sample sets, the modelling result of the$\hat{R}_a$ training phase is presented in Table 6. The MSE value of $\hat{R}_a$ for eight DOE $2^k$ sample sets which are chosen as testing samples is presented in Table 7. Furthermore, the results of Tables 6 and 7 are used to determine the best network structure of the ANN prediction model discussed in the next section.

## 7. Determination of the best ANN model

In order to determine the best network structure of the ANN prediction model, two criteria are given consideration. The first criterion is the consideration of the line pattern data between ANN targets ($Y_{R_a}$) and ANN outputs ($\hat{R}_a$) of the training sample set. In this way, these two lines are generated on the same graph. If the patterns of the two lines of the graph look similar, then an assumption can be made that there is a good agreement observed between ANN targets and ANN outputs. The second criterion is the consideration of the smallest value for the absolute average error value of the testing sample set to justify the network structure that gives the best prediction for the surface roughness value. In this way, the absolute average value of MSE is calculated for each network structure. The aim is for the MSE absolute average to be as small as possible (approaching zero) to justify that the network structure has given the best prediction.

In relation to the first criterion, Fig. 9 presents the graph that shows the pattern of the data between the ANN target data and ANN output data. This graph is generated by using the surface roughness predicted values of the ANN structure in the training phase given in Table 6. On assessing the graph in Fig. 9, it could be summarized that six network structures have given quite a similar form of line pattern between ANN targets ($Y_{R_a}$) and ANN outputs ($\hat{R}_a$) which are 3–7–1 uncoated, 3–1–1–1 uncoated, 3–1–1 TiAlN coated, 3–7–7–1 TiAin coated, 3–1–1 $SN_{TR}$ coated and 3–7–7–1 $SN_{TR}$ coated network structures. In relation to the second criterion, the absolute average value of MSE for the surface roughness predicted values of the ANN structure in the testing phase is given in Table 7. On assessing the absolute average value of MSE, it could be summarized that the 3–1–1 network structure of the $SN_{TR}$ coated cutting tool has given the lowest absolute MSE average value which is 0.023 for the six best potential models.

With the two facts listed above, the assumption could be made that the 3–1–1 network structure of the $SN_{TR}$ coated cutting tool has given the best prediction for the surface roughness value. Next, to justify whether the 3–1–1 network structure of the $SN_{TR}$ coated cutting tool will give a better result than the result for the real experiment for surface roughness prediction, the following condition is considered:

$$\hat{R}_a \min\_SNTR < Y_{R_a} \min\_SNTR \tag{21}$$

In Eq. (21), $Y_{R_a} \min\_SNTR$ is the lowest normalized surface roughness value (ANN target lowest value) for the $SN_{TR}$ cutting tool which is given in Table 5. $\hat{R}_a \min_SNTR$is the lowest ANN predicted surface roughness value of the 3–1–1 network structure (ANN output lowest value) for the $SN_{TR}$ coated cutting tool which could be obtained from Table 6. The perception of this study is that, if the $\hat{R}_a \min_SNTR < Y_{R_a} \min_SNTR$condition is fulfilled, it could be concluded that the implicit ANN model of the 3–1–1 network structure of the $SN_{TR}$ coated cutting tool has created a better result than the result of real experiment.

A value of $Y_{R_a} \min\_SNTR$= 0.100 can be obtained from the sixth row of Table 5. A value of $\hat{R}_a \min\_SNTR = 0.088$ can be obtained from the seventh and eighth rows of Table 6. Since the value of $\hat{R}_a \min\_SNTR$ (0.088) $< Y_{R_a} \min\_SNTR$ (0.100) is obtained, it could be concluded that the ANN model using the 3–1–1 network structure of the $SN_{TR}$ coated cutting tool has given a better result for

**Table 6**
Surface roughness predicted values of the ANN structure in the training phase.

| No. | Data source | $\hat{R}_{a\_}$ uncoated | | | | | | | | $\hat{R}_{a\_}$ TiAIN coated | | | | | | | | $\hat{R}_{a\_}SN_{TR}$ coated | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 3–1–1 | 3–3–1 | 3–6–1 | 3–7–1 | 3–1–1–1 | 3–3–3–1 | 3–6–6–1 | 3–7–7–1 | 3–1–1 | 3–3–1 | 3–6–1 | 3–7–1 | 3–1–1–1 | 3–3–3–1 | 3–6–6–1 | 3–7–7–1 | 3–1–1 | 3–3–1 | 3–6–1 | 3–7–1 | 3–1–1–1 | 3–3–3–1 | 3–6–6–1 | 3–7–7–1 |
| 1 | Centre | 0.069 | 0.185 | 0.348 | 0.291 | 0.224 | 0.406 | 0.417 | 0.409 | 0.255 | 0.334 | 0.160 | 0.342 | 0.229 | 0.387 | 0.372 | 0.296 | 0.308 | 0.366 | 0.519 | 0.317 | 0.081 | 0.387 | 0.385 | 0.412 |
| 2 | | 0.069 | 0.185 | 0.348 | 0.291 | 0.224 | 0.406 | 0.417 | 0.409 | 0.255 | 0.334 | 0.160 | 0.342 | 0.229 | 0.387 | 0.372 | 0.296 | 0.308 | 0.366 | 0.519 | 0.317 | 0.081 | 0.387 | 0.385 | 0.412 |
| 3 | | 0.069 | 0.185 | 0.348 | 0.291 | 0.224 | 0.406 | 0.417 | 0.409 | 0.255 | 0.334 | 0.160 | 0.342 | 0.229 | 0.387 | 0.372 | 0.296 | 0.308 | 0.366 | 0.519 | 0.317 | 0.081 | 0.387 | 0.385 | 0.412 |
| 4 | | 0.069 | 0.185 | 0.348 | 0.291 | 0.224 | 0.406 | 0.417 | 0.409 | 0.255 | 0.334 | 0.160 | 0.342 | 0.229 | 0.387 | 0.372 | 0.296 | 0.308 | 0.366 | 0.519 | 0.317 | 0.081 | 0.387 | 0.385 | 0.412 |
| 5 | Axial | 0.033 | 0.328 | 0.277 | 0.225 | 0.197 | 0.481 | 0.346 | 0.441 | 0.565 | 0.576 | 0.449 | 0.043 | 0.474 | 0.390 | 0.463 | 0.157 | 0.769 | 0.425 | 0.057 | 0.006 | 0.057 | 0.457 | 0.317 | 0.297 |
| 6 | | 0.033 | 0.328 | 0.277 | 0.225 | 0.197 | 0.481 | 0.346 | 0.441 | 0.565 | 0.576 | 0.449 | 0.043 | 0.474 | 0.390 | 0.463 | 0.157 | 0.769 | 0.425 | 0.057 | 0.006 | 0.057 | 0.457 | 0.317 | 0.297 |
| 7 | | 0.362 | 0.454 | 0.295 | 0.276 | 0.357 | 0.241 | 0.095 | 0.031 | 0.118 | 0.284 | 0.333 | 0.609 | 0.130 | 0.390 | 0.247 | 0.438 | 0.088 | 0.302 | 0.150 | 0.648 | 0.267 | 0.355 | 0.509 | 0.412 |
| 8 | | 0.362 | 0.454 | 0.295 | 0.276 | 0.357 | 0.241 | 0.095 | 0.031 | 0.118 | 0.284 | 0.333 | 0.609 | 0.130 | 0.390 | 0.247 | 0.438 | 0.088 | 0.302 | 0.150 | 0.648 | 0.267 | 0.355 | 0.509 | 0.412 |
| 9 | | 0.262 | 0.240 | 0.027 | 0.255 | 0.189 | 0.283 | 0.247 | 0.456 | 0.098 | 0.232 | 0.157 | 0.319 | 0.127 | 0.457 | 0.374 | 0.135 | 0.104 | 0.640 | 0.159 | 0.254 | 0.427 | 0.386 | 0.330 | 0.097 |
| 10 | | 0.262 | 0.240 | 0.027 | 0.255 | 0.189 | 0.283 | 0.247 | 0.456 | 0.098 | 0.232 | 0.157 | 0.319 | 0.127 | 0.457 | 0.374 | 0.135 | 0.104 | 0.640 | 0.159 | 0.254 | 0.427 | 0.386 | 0.330 | 0.097 |
| 11 | | 0.028 | 0.524 | 0.648 | 0.653 | 0.889 | 0.383 | 0.615 | 0.337 | 0.902 | 0.469 | 0.670 | 0.432 | 0.762 | 0.325 | 0.406 | 0.801 | 0.877 | 0.220 | 0.461 | 0.567 | 0.051 | 0.333 | 0.464 | 0.759 |
| 12 | | 0.028 | 0.524 | 0.648 | 0.653 | 0.889 | 0.383 | 0.615 | 0.337 | 0.902 | 0.469 | 0.670 | 0.432 | 0.762 | 0.325 | 0.406 | 0.801 | 0.877 | 0.220 | 0.461 | 0.567 | 0.051 | 0.333 | 0.464 | 0.759 |
| 13 | | 0.127 | 0.101 | 0.578 | 0.528 | 0.211 | 0.404 | 0.013 | 0.436 | 0.227 | 0.388 | 0.640 | 0.138 | 0.101 | 0.406 | 0.540 | 0.218 | 0.422 | 0.294 | 0.731 | 0.117 | 0.238 | 0.409 | 0.174 | 0.231 |
| 14 | | 0.127 | 0.101 | 0.578 | 0.528 | 0.211 | 0.404 | 0.013 | 0.436 | 0.227 | 0.388 | 0.640 | 0.138 | 0.101 | 0.406 | 0.540 | 0.218 | 0.422 | 0.294 | 0.731 | 0.117 | 0.238 | 0.409 | 0.174 | 0.231 |
| 15 | | 0.038 | 0.624 | 0.066 | 0.076 | 0.262 | 0.266 | 0.890 | 0.059 | 0.310 | 0.100 | 0.277 | 0.447 | 0.876 | 0.305 | 0.330 | 0.525 | 0.183 | 0.219 | 0.009 | 0.834 | 0.053 | 0.332 | 0.468 | 0.432 |
| 16 | | 0.038 | 0.624 | 0.066 | 0.076 | 0.262 | 0.266 | 0.890 | 0.059 | 0.310 | 0.100 | 0.277 | 0.447 | 0.876 | 0.305 | 0.330 | 0.525 | 0.183 | 0.219 | 0.009 | 0.834 | 0.053 | 0.332 | 0.468 | 0.432 |

**Table 7**
MSE of surface roughness predicted values of the ANN structure in the testing phase.

| No. | Data source | MSE of $\hat{R}_a$ uncoated | | | | | | | | MSE of $\hat{R}_a$ TiAIN coated | | | | | | | | MSE of $\hat{R}_a$ $SN_{TR}$ coated | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 3–1–1 | 3–3–1 | 3–6–1 | 3–7–1 | 3–1–1–1 | 3–3–3–1 | 3–6–6–1 | 3–7–7–1 | 3–1–1 | 3–3–1 | 3–6–1 | 3–7–1 | 3–1–1–1 | 3–3–3–1 | 3–6–6–1 | 3–7–7–1 | 3–1–1 | 3–3–1 | 3–6–1 | 3–7–1 | 3–1–1–1 | 3–3–3–1 | 3–6–6–1 | 3–7–7–1 |
| 1 | DOE $2^k$ | −0.271 | −0.177 | 0.320 | 0.315 | −0.181 | −0.008 | −0.161 | 0.287 | 0.240 | −0.541 | 0.192 | 0.237 | 0.001 | 0.176 | 0.004 | 0.171 | −0.048 | −0.100 | −0.483 | 0.201 | −0.462 | −0.180 | 0.199 | −0.021 |
| 2 | | −0.236 | 0.005 | −0.218 | −0.144 | 0.034 | 0.078 | −0.021 | −0.084 | 0.103 | −0.033 | −0.038 | −0.094 | 0.119 | 0.079 | −0.096 | 0.041 | −0.729 | 0.033 | −0.464 | 0.084 | −0.620 | −0.538 | −0.349 | −0.067 |
| 3 | | −0.182 | 0.002 | 0.563 | 0.549 | −0.153 | 0.541 | −0.132 | 0.104 | 0.777 | 0.295 | 0.665 | 0.100 | −0.028 | 0.755 | 0.035 | 0.177 | 0.058 | −0.037 | 0.214 | 0.078 | 0.306 | 0.198 | 0.163 | 0.846 |
| 4 | | −0.234 | 0.425 | −0.214 | 0.031 | −0.204 | 0.473 | 0.453 | −0.064 | 0.143 | 0.474 | −0.016 | −0.329 | 0.066 | 0.502 | 0.411 | −0.016 | −0.088 | 0.322 | 0.378 | −0.013 | 0.075 | 0.137 | 0.022 | 0.775 |
| 5 | | 0.417 | 0.303 | 0.276 | 0.183 | 0.408 | −0.012 | −0.051 | −0.058 | 0.125 | 0.021 | −0.268 | 0.087 | 0.103 | −0.205 | −0.152 | 0.046 | 0.191 | 0.067 | −0.210 | −0.049 | 0.179 | −0.113 | −0.036 | 0.228 |
| 6 | | 0.089 | −0.223 | 0.024 | −0.246 | 0.077 | −0.148 | 0.030 | −0.005 | −0.162 | −0.327 | −0.872 | 0.030 | 0.056 | −0.185 | −0.176 | −0.117 | −0.010 | −0.243 | 0.003 | −0.208 | −0.079 | −0.235 | −0.236 | 0.093 |
| 7 | | 0.303 | 0.210 | 0.654 | 0.135 | 0.060 | 0.674 | −0.034 | −0.009 | 0.397 | 0.140 | 0.073 | 0.063 | −0.321 | 0.146 | 0.314 | −0.158 | 0.690 | −0.118 | −0.138 | 0.095 | 0.713 | 0.454 | 0.239 | 0.758 |
| 8 | | 0.452 | −0.183 | 0.100 | −0.146 | 0.410 | 0.484 | 0.285 | −0.003 | −0.344 | −0.029 | −0.124 | −0.009 | 0.345 | 0.327 | 0.283 | −0.559 | 0.116 | −0.014 | 0.195 | 0.006 | 0.648 | 0.399 | 0.200 | 0.702 |
| \|Average\| | | 0.042 | 0.045 | 0.188 | 0.085 | 0.056 | 0.260 | 0.046 | 0.021 | 0.160 | 0.000 | 0.049 | 0.011 | 0.043 | 0.199 | 0.078 | 0.052 | 0.023 | 0.011 | 0.063 | 0.024 | 0.095 | 0.015 | 0.025 | 0.414 |

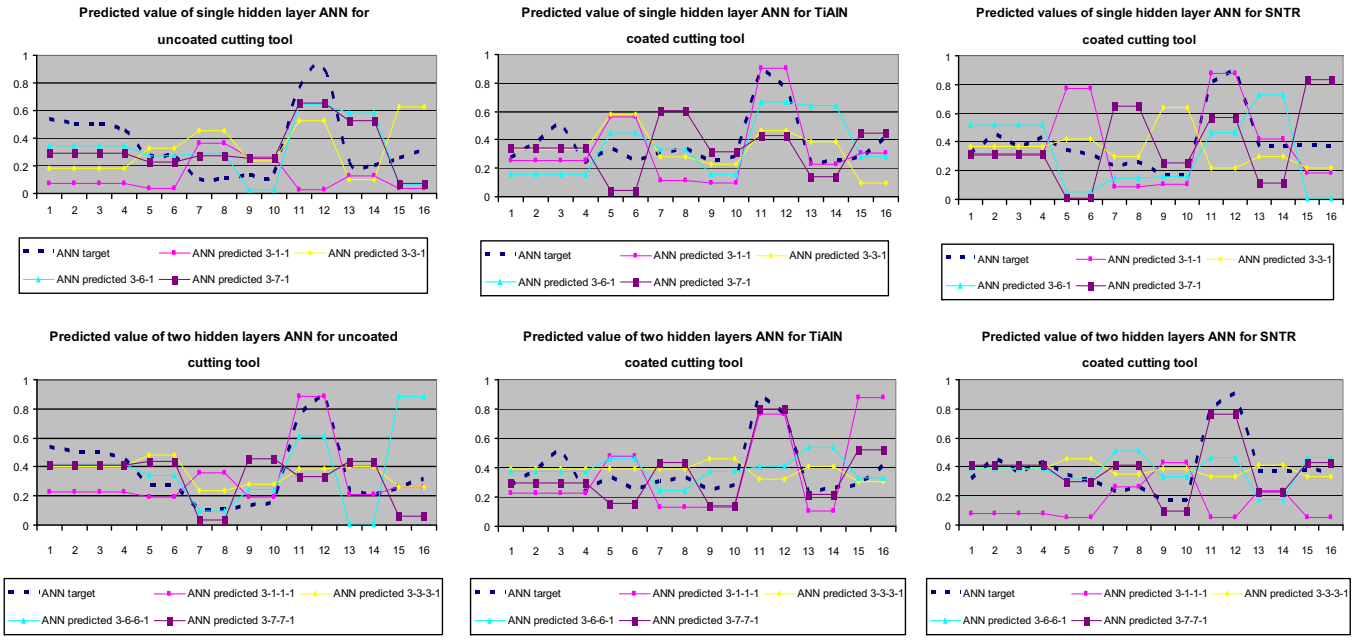


**Fig. 9.** Investigation of the similarity of line pattern between $Y_{R_a}$ and $\hat{R}_a$ in the training ANN phase.

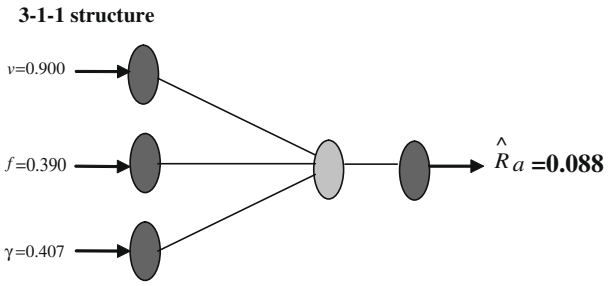


**Fig. 10.** The best ANN network structure for surface roughness prediction.

surface roughness prediction than the result for the real experiment. This is summarized in Fig. 10 in the network structure form.

The predicted result is that, by applying the $SN_{TR}$ coated cutting tool, the recommended combination of cutting conditions to obtain the best surface roughness value is a high rate of speed with a low rate of feed and radial rake angle.

## 8. Conclusion

In this paper, a review of the ANN technique to develop the prediction model for surface roughness has been discussed. Examples of studies are given, with their relative abilities and limitations, in relation to the modelling of the machining process focusing on the prediction of a surface roughness measurement by using ANN approaches. A discussion also focuses on the methodology of ANN as a potential modelling technique for the prediction of surface roughness in machining.

An important issue of concern in this study is the basic idea of ANN being applied for modelling purposes, particularly for a performance measure for surface roughness in the milling process. In our opinion, the very basic idea relates to the understanding of the network structure of ANN. The determination of the number of layers and nodes in the hidden layers using the trial-and-error method is important for the researcher to give a good prediction. This study has proven that by modifying the number of layers and nodes in the hidden layer, with the same ANN training algorithm and number of training samples used, it could give a different $\hat{R}_a$ value for surface roughness prediction. This study has applied eight different network structures which are *3–1–1, 3–3–1, 3–6–1, 3–7–1, 3–1–1–1, 3–3–3–1, 3–6–6–1, and 3–7–7–1*. Based on three different cutting tools, twenty four models were developed and it was found that the 3–1–1 structure of the $SN_{TR}$ cutting tool has given the best prediction for the performance measure for surface roughness.

This study also has proven that the accurate resulting value for the prediction performance measure could be obtained with the small training and testing samples by using the available experimental data. With a total sample size of 24, separated into 16 samples for training and 8 samples for testing, it was found that by applying the ANN technique an improvement in the resulting value for surface roughness prediction could be obtained when compared to the real machining result. In other words, ANN is still capable of generating accurate prediction values for the performance measure using a small number of training samples. It can be concluded that the small number of modelled samples is not the main issue in obtaining a good prediction. The result of the prediction depends on how the number of layers and nodes is modified in the hidden layer of the ANN network structure. Creating a different model by modifying the number of layers and neurons in the hidden layer could increase the cost and time of the modelling process, but it is beneficial to the researchers in order to give some alternatives for prediction values to be selected as the best prediction.

To conclude, this study has been involved with the ANN technique for the model development to predict the values of the performance measure, focusing on surface roughness in the end milling machining operation. It can also be defined as the modelling phase of the machining process which is important for providing the basic model for the formulation of the objective function. In the next phase, consideration should be given to studying the determination of conditions for optimization for the objective function. It is also called the optimizing phase which is important for obtaining the optimal solution of the predicted value obtained from the modelling phase. Several optimization techniques such as the Genetic Algorithm (GA), Simulated Annealing (SA), Tabu Search (TS), Ant Colony Algorithm (ACO) and Particle Swarm Optimization

(PSO), which can be classified as the Artificial Intelligent approach, have the potential to be applied for cutting parameter optimization problems. When considering future areas of work, it is recommended that these techniques be chosen as the optimizing techniques to be hybridized with ANN to optimize all the cutting parameter values that have affected the predicted surface roughness value.

## References


Abeesh, B. C., Dabade, U. A., Joshi, S. S., Bhanuprasad, V. V., & Gadre, V. M. (2008). Modeling of surface roughness in precision machining of metal matrix composites using ANN. *Journal of Material Processing Technology, 197*, 439–444.

Al-Ahmari, A. M. A. (2007). Predictive machinability models for a selected hard material in turning operations. *Journal of Material Processing Technology, 190*, 305–311.

Benardos, P. G., & Vosnaikos, G. C. (2003). Predicting surface roughness in machining: a review. *International Journal of Machine Tools and Manufacture, 43*, 833–844.

Brezocnik, M., Kavocic, M., & Ficko, M. (2004). Prediction of surface roughness with genetic programming. *Journal of Material Processing Technology*, 28–36.

Cus, F., & Zuperl, U. (2006). Approach to optimization of cutting conditions by using artificial neural networks. *Journal of Material Processing Technology, 173*, 281–290.

Davim, J. P., Gaitonde, V. N., & Karmik, S. R. (2008). Investigations into the effect of cutting conditions on surface roughness in turning of free machining steel by ANN models. *Journal of Material Processing, 205*, 16–23.

Demuth, H., Beale, M., & Hagan, M. (2008). Neural Network Toolbox™ 6 User's Guide. The MathWorks, Inc. 3 Apple Hill Drive Natick, MA 01760-2098.

Dweiri, F., Al-Jarrah, M., & Al-Wedyan, H. (2003). Fuzzy surface roughness modeling of CNC down milling of Alumic-79. *Journal of Material Processing Technology, 133*, 266–275.

Erzurumlu, T., & Oktem, H. (2007). Comparison of response surface model with neural network in determining the surface quality of moulded parts. *Materials and Design, 28*, 459–465.

Ezugwu, E. O., Fadare, D. A., Bonneya, J., Silva, R. B. D., & Sales, W. F. (2005). Modelling the correlation between cutting and process parameters in high-speed machining of Inconel 718 alloy using an artificial neural network. *International Journal of Machine Tools and Manufacture, 45*, 1375–1385.

Grzesick, W., & Brol, S. (2003). Hybrid approach to surface roughness evaluation in multistage machining processes. *Journal of Material Processing Technology, 134*, 265–272.

Hans, K., Swarup, R., Srivastava, S., & Patvardhan, C. (2000). Modeling of manufacturing processes with ANNs for intelligent manufacturing. *International Journal of Machine Tools and Manufacture, 40*, 851–868.

Kafkas, F., Karatas, C., Sozen, A., Arcaklioglu, E., & Saritas, S. (2007). Determination of residual stresses based on heat treatment conditions and densities on a hybrid (FLN2-4405) powder metallurgy steel using artificial neural network. *Materials and Design, 28*, 2431–2442.

Kalpakjian, S. (1991). *Manufacturing engineering and technology* (2nd ed.). Reading, MA: Addision-Wesley.

Kohli, A., & Dixit, U. S. (2005). A neural-network-based methodology for the prediction of surface roughness in a turning process. *International Journal of Advanced Manufacturing Technology, 25*, 118–129.

Luttervelt, C. A. V., Childs, T. H. C., Jawahir, I. S., Klocke, F., & Venuvinod, P. K. (1998). Keynote papers: Present situation and future trends in modelling of machining operations. *Annals of the ClRP, 47*(2).

Mohruni, A. S. (2008). *Performance evaluation of uncoated and coated carbide tools when end milling of titanium alloy using response surface methodology*. Thesis Doctor of Philosophy, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia.

Mukherjee, I., & Ray, P. K. (2006). A review of optimization techniques in metal cutting processes. *Computer and Industrial Engineering, 50*, 15–34.

Nalbant, M., Gokkaya, H., Toktas, I., & Sur, G. (2009). The experimental investigation of the effects of uncoated, PVD- and CVD-coated cemented carbide inserts and cutting parameters on surface roughness in CNC turning and its prediction using artificial neural networks. *Robotics and Computer-Integrated Manufacturing, 25*, 211–223.

Oktem, H., Erzurumlu, T., & Erzincanli, F. (2006). Prediction of minimum surface roughness in end milling mold parts using neutral network and genetic algorithm. *Journal of Material and Design, 27*, 735–744.

Sanjay, C., & Jyothi, C. (2006). A study of surface roughness in drilling using mathematical analysis and neural networks. *International Journal of Advanced Manufacturing Technology, 29*, 846–852.

Sheikh-Ahmad, J., & Twomey, J. (2007). ANN constitutive model for high strain-rate deformation of Al 7075-T6. *Journal of Material Processing Technology, 186*, 339–345.

Tansel, I. N., Ozcelik, B., Bao, W. Y., Chen, P., Rincon, D., & Yang, S. Y. (2006). Selection of optimal cutting conditions by using GONNS. *International Journal of Machine Tools and Manufacture, 46*, 26–35.

Tsai, Y. H., Chen, J. C., & Luu, S. J. (1999). An in-process surface recognition system based on neural networks in end milling cutting operations. *International Journal of Machine Tools and Manufacture, 39*, 583–605.

Yang, J. L., & Chen, J. C. (2001). A systematic approach for identifying optimum surface roughness performance in end-milling operation. *Journal of Industrial Technology, 45*, 110–120.

Zhang, G., Patuwo, B. E., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting, 14*, 35–62.

Zuperl, U., & Cus, F. (2003). Optimization of cutting conditions during cutting by using neural networks. *Robotics and Computer-Integrated Manufacturing, 19*, 189–199.

Zuperl, U., Cus, F., Mursec, B., & Ploj, T. (2006). A generalized neural network model of ball-end milling force system. *Journal of Material Processing Technology, 175*, 98–108.