The International Journal of Advanced Manufacturing Technology (2021) 114:2591–2615
https://doi.org/10.1007/s00170-021-06712-4

ORIGINAL ARTICLE

Artificial neural network for modeling and investigating the effects of forming tool characteristics on the accuracy and formability of thin aluminum alloy blanks when using SPIF

Sherwan Mohammed Najm[1,2] · Imre Paniti[1,3,4]

Received: 14 August 2020 / Accepted: 27 January 2021 / Published online: 12 April 2021

Abstract

Incremental Sheet Forming (ISF) has attracted attention due to its flexibility as far as its forming process and complexity in the deformation mode are concerned. Single Point Incremental Forming (SPIF) is one of the major types of ISF, which also constitutes the simplest type of ISF. If sufficient quality and accuracy without defects are desired, for the production of an ISF component, optimal parameters of the ISF process should be selected. In order to do that, an initial prediction of formability and geometric accuracy helps researchers select proper parameters when forming components using SPIF. In this process, selected parameters are tool materials and shapes. As evidenced by earlier studies, multiple forming tests with different process parameters have been conducted to experimentally explore such parameters when using SPIF. With regard to the range of these parameters, in the scope of this study, the influence of tool material, tool shape, tool-end corner radius, and tool surface roughness (*Ra*/*Rz*) were investigated experimentally on SPIF components: the studied factors include the formability and geometric accuracy of formed parts. In order to produce a well-established study, an appropriate modeling tool was needed. To this end, with the help of adopting the data collected from 108 components formed with the help of SPIF, Artificial Neural Network (ANN) was used to explore and determine proper materials and the geometry of forming tools: thus, ANN was applied to predict the formability and geometric accuracy as output. Process parameters were used as input data for the created ANN relying on actual values obtained from experimental components. In addition, an analytical equation was generated for each output based on the extracted weight and bias of the best network prediction. Compared to the experimental approach, analytical equations enable the researcher to estimate parameter values within a relatively short time and in a practicable way. Also, an estimate of Relative Importance (RI) of SPIF parameters (generated with the help of the partitioning weight method) concerning the expected output is also presented in the study. One of the key findings is that tool characteristics play an essential role in all predictions and fundamentally impact the final products.

Keywords SPIF · Incremental sheet forming · Single point · Flat tool · ANN · Predict formability · RI · Thin aluminum alloy blanks

✉ Sherwan Mohammed Najm
sherwan.mohammed@gpk.bme.hu

Imre Paniti
imre.paniti@sztaki.hu

[1] Department of Manufacturing Science and Engineering, Budapest University of Technology and Economics, H-1111, Műegyetem rkp. 3, Budapest, Hungary

[2] Kirkuk Technical Institute, Northern Technical University, Kirkuk, Iraq

[3] Centre of Excellence in Production Informatics and Control, Institute for Computer Science and Control (SZTAKI), H-1111, Kende u. 13-17, Budapest, Hungary

[4] Széchenyi István University, Research Center of Vehicle Industry, Egyetem Sq 1, Győr H-9026, Hungary

1 Introduction

Incremental Sheet Forming (ISF) is suitable for low-volume production and is ideal for complicated designs. ISF was patented in 1967 [1], and one of the crucial types of ISF is Single Point Incremental Forming (SPIF). Emerging manufacturing technologies like ISF developed in the past few decades. Researchers have shown that an unconventional sheet forming process like ISF is economically feasible for producing prototypes; ISF is also versatile and can produce custom and complex products [2, 3]. Given this, a comprehensive literature review about ISF is presented in [4]. Also, a brief review of the history of ISF with a focus on the technological progress involved is found in [1]. This review, however, focuses on the mechanism of the deformation, modeling techniques, forming force prediction, and an investigation of SPIF. Furthermore, the articles reviewed in this paper all state that ISF is suitable for economical prototyping and is suitable for preparing customized and complex sheet products.

Shrivastava and Tandon [5] investigated components experimentally formed by SPIF and analyzed them using a finite element analysis in order to understand the characteristics of sheet deformation, forming behavior, and dominant deformation mechanisms. Shrivastava and Tandon concluded that ISF is a process capable of fulfilling industry demand for highly complex, economic, and customized products. Duflou et al. [6] assert that one of the most significant factors which influence the geometric accuracy of SPIF is tool diameter. Maqbool and Bambach [7] investigated various SPIF process parameters, including tool diameter on a pyramidal frustum, and found that in relation to geometry, smaller tool diameter positively affects geometrical accuracy in the range (5 mm, 10 mm, and 20 mm) of the tool tip diameter. Brendan et al. [8] examined two tool tip types (parabolic and angle radius) and compared the results using hemispherical and flat-bottomed tool tips. They found that the angular profile tool tip improves formability but concluded that formability is highest when the contact surface is decreased in the parabolic tool tip. Najm and Paniti [9] studied the effect of flat-end tools on SPIF components of thin sheets and found that the smallest corner radius of the flat tool in a range of (0.1 mm, 0.3 mm, and 0.5 mm) gives the best results in terms of forming depth and geometric accuracy. Two different tool ends (flat and hemispherical) were used by Ziran et al. [10] to form an AA-3003O aluminum sheet. They found that better geometric accuracy and formability can be achieved by using a flat tool rather than applying a hemispherical one. Moreover, they also established that relatively low forming force is needed when flat ends are used as compared to hemispherical ends. On the other hand, Wu et al. [11] claimed that SPIF is a process that exhibits flexibility in sheet forming, which in turn enables the process to be used for producing customized complex dimensional shape parts utilizing different materials. Many studies have been conducted to understand SPIF, but a majority of them deal with a sheet thickness of over 0.5 mm. Similarly, many researchers have studied SPIF parameters, but there is less research related specifically to the class of process parameters. For instance, there is no research studying the prediction of forming depth when a flat tool and SPIF are applied with an initial sheet thickness of less than 0.5 mm. No satisfactory solution has been found to improve the geometric accuracy of sheets below 0.5 mm under various conditions in the case of SPIF. In fact, the effects of tool materials and shape on the final product quality have not been discussed in any of the above-mentioned studies. Even if SPIF is quite flexible, it has limitations: its drawbacks, for instance, include the accuracy of the geometry of components and the planned achievable depth.

Recently, various techniques of artificial intelligence have been used in many industries, including the metal forming industry: specifically, Artificial Neural Network (ANN) is used for developing predictive models for end-milling machining, powder metallurgy, and high-speed machining [12–14]. In addition, machine learning techniques have dominated manufacturing in an attempt to develop the most effective predictive models [15–19]. There are also different optimization algorithms commonly used in manufacturing processes. For example, the Johnson-Cook model constants (J-C constants) of ultra-fine-grained titanium were researched by Ning et al. [20]: based on the chip formation model, they identified such constants via enforcing the gradient search method using a Kalman filter. Later on, Ning and Liang [21] developed an inverse identification method for J-C constants by replacing the exhaustive search method with an iterative gradient search method in the Kalman Filter algorithm. They predicted machining forces using the modified chip formation model and the J-C constants. They found a close correspondence between predicted forces and experimental forces. In another study, Gok [22] introduced a new method for determining optimal cutting parameters by applying fuzzy TOPSIS and gray relational analysis. He found that the lowest values of cutting velocity, feed rate, and cutting depth produce the smallest Ra, Rt, Ff, and Fc values in terms of surface roughness. The results obtained using fuzzy TOPSIS are in accordance with the Gray relational analysis. Zuo et al. [23] presented a new approach to reduce design space and guarantee topology outcomes concerning manufacturability and engineering, i.e., they recommended a design that conforms to accepted principles, tests, or standards. They also introduced manufacturing and machining constraints to the topology optimization method formula. Their investigations suggest that modified topology optimization can solve non-manufacturing and non-machining problems related to engineering applications.

Kurra et al. [24] incrementally predicted surface roughness of Extra Deep Drawing (EDD) steel under various forming conditions. They evaluated the performance of ANN, SVR, and a model developed in the scope of the study (Genetic Programming) using an R-squared value. Using a feed-forward neural network with a Backpropagation algorithm, Nasrollahi and Arezoo [25] used training data for two different ANN models to predict the springback of bending in the case of sheet metals with holes. They found that data used for one type of hole, i.e., an oblong hole, and for three other types of holes (oblong, circle, square) in the bending area all affected springback. They also established that the use of all types of holes produces more accurate result for the prediction of springback: case errors are fewer than in the case of training each hole separately. Mekras [26] implemented an ANNs model for successful process set-up and used it in the scope of sheet metal forming theory. In the model, set-up parameters including aluminum alloy type, sheet thickness, pressing speed, and the tools' geometrical details were considered. Mekras found that even in the multi-input and multi-output process models, which included three inputs and four outputs, the models' accuracy was satisfactory. Kashid and Kumar [27] reviewed sixty-three published research articles and examined the applications of the ANN technique in sheet metal working; SPIF was not mentioned in any of the works cited in the study.

Nowadays, there are many articles concerned with modeling and optimizing different parameters in SPIF processes using an artificial neural network. Maji and Kumar [28] found that the Adaptive Neuro-Fuzzy Inference System (ANFIS) yields more accurate prediction when a hybrid algorithm is used, and even more so when a Backpropagation algorithm is applied. They developed a response surface methodology and ANFIS to predict the outcome of SPIF components; they considered different process parameters and dealt with inverse predictions of process parameters in SPIF. Furthermore, they utilized the desirability function and a non-dominated sorting genetic algorithm for performing multi-objective optimization in the scope of SPIF. Oraon and V. Sharma [29] predicted the surface quality of SPIF parts by adopting the ANN model using a feed-forward neural network along with a backpropagation learning algorithm. They reported a result of 94.744% for ANN simulation performance with a mean absolute error of 1.068%. Also, an ANN model was utilized by Mulay et al. [30] to predict the average surface roughness and the wall angle of AA5052-H3 parts manufactured using SPIF. Oraon et al.[31] trained feed-forward backpropagation (FFBP) in an ANN model with a structure 6-6-1 to predict the surface roughness of a brass Cu67Zn33 piece formed by way of SPIF. Radu et al. [32] evaluated the effectiveness of the Response Surface Method (RSM) and the Neural Network (NN) method for improving and controlling the accuracy of SPIF components. Basing their claims on the accuracy of their experiments, they suggest further research of a broader range of process parameters; they claim that such investigation will help to generate valid general empirical models. Behera et al. [33] analyzed the accuracy of truncated pyramids formed using SPIF. They suggested studying the effects of the interaction between diverse features of SPIF, and they made predictions for that purpose. In addition, they also investigated the effects of material properties and sheet thickness on accuracy profiles. In another study, McAnulty et al. [34] described the effects of forming tip diameter on formability, which is the focus parameter in their review paper: their efforts are underpinned by the fact that contradictory results were published about the impact of tool diameter on formability. Ten articles reported that a decrease in tip diameter causes a reduction in formability, whereas seven articles claimed the opposite. However, six of the studies claimed that the tip diameter should be optimized to reach maximum formability. Bayram and Koksal [35] found that a 0.5 mm step size offers better homogeneous distribution and geometrical accuracy than a 0.2 mm step size in the case of SPIF concerning a 1 mm thick AA2024 aluminum alloy. Nama et al. [36] found that a larger tool head, an increase in tool speed and feed rate lead to better surface roughness of an aluminum 1100 sheet with 0.6 mm thickness. Rattanachan and Chungchoo [37] investigated the formability of DIN 1.0037 steel and found that formability decreased as a consequence of an increase in tool speed. Based on a review published by Nimbalkar and Nandedkar [38], the most significant facet in SPIF is the forming tool. For the optimization of the SPIF process, the quality of formed components should be maximized. The ideal characteristics of the product formed using SPIF are geometric accuracy and maximum forming depth, which can be reached using a desired shape. Besides, formed part accuracy is one of the significant elements of the process capability of SPIF.

Based on the literature, it can be concluded that ANN could be a useful tool for result prediction and modeling prior to starting new experiments. The benefits of using machine learning based artificial neural networks before starting new experiments consist in reducing the time needed for preparing the experiments, minimizing the errors, and increasing efficiency. Furthermore, ANN is one of the most powerful tools to predict experimental data for solving engineering problems, and ANN can serve as a very useful tool to create and evaluate processes and to determine the final details of tools.

The above-detailed issues as well as the lack of well-defined requirements of the SPIF process and the absence of referent mathematical models have motivated the authors to examine the investigation and prediction of the formability and geometric accuracy of truncated frustums processed using SPIF. To the authors' knowledge, such an experimental process has not been tested or described in the literature. In fact, in the scope of the present project, forming depth was considered an indicator of formability: forming depth is deemed as an indicator of the formability of formed parts, as described in [39–46]. Furthermore, as an aim and novelty in the scope of this paper, a prediction equation for both accuracy and formability based on weights and biases was derived, as well as the joint partitioning weight of the neural network was adopted to assess the Relative Importance (RI) of SPIF parameters on the output. In view of this and in the scope of this paper, influence parameters are tool materials, tool shape, end corner radius, and the surface roughness (*Ra*/*Rz*) of the tool.

2 Material properties

In the experiments conducted in the scope of this study, the components' blank sheet was made of 0.22 mm AlMn1Mg1 aluminum alloy. By cutting the specimens from the sheet at 0°, 45°, and 90° of the rolling directions, tensile tests were

Table 1 Mechanical properties of blank material

Direction	0^0	45^0	90^0
Yield strength (MPa) 0.2	88.30	90	86.3
Ultimate tensile strength (MPa)	183	155.5	170.3
Elongation (%)	16.44	9.27	12.48
Elongation A50 (%)	16.88	10.45	12.95
n5	0.297	0.266	0.268
r10	0.554	0.580	0.594

carried out at room temperature using an INSTRON 5582 universal testing machine. In the research design, 3 samples were used for each direction regarding the rolling direction. As shown in Table 1, which presents average data of the mechanical properties of the sheet material, the relative standard deviation did not exceed 3%. In the scope of the research design, related tensile tests were carried out based on the EN ISO 6892-1:2010 standard, and an Advanced Video Extensometer (AVE) was used to measure the planar anisotropy values (r10).

Table 2 shows the chemical composition of the sheet material. In addition, in the SPIF experiments, different tool materials were used: tool materials (see Fig. 1) consisted of steel (C45), brass (CuZn39Pb3), bronze (CuSn12), copper (E-Cu57), aluminum (AlMgSi 0.5), and polymer VeroWhitePlus (RGD835). The hardness of the tools was tested experimentally with the help of a Wolpert Diatronic 2RC S hardness tester, and the measurement was carried out based on the ISO 6506-1:2014 standard. The materials were measured by a FOUNDRY-MASTER Pro2 Optical Emission Spectrometer in order to determine the ISO code of each type of tool material. Using the ISO code of each material, the mechanical properties of the metallic tool were listed. The properties were ensured by measurements executed by [SAARSTAHL, L. KLEIN SA, AURUBIS, PX PRECIMET SA, ALUMINCO S.A.] based on the sequence of the materials in Table 3. Table 4 presents the properties of the polymer tool provided by STRATASYS.

3 Experiments

Experimental tests were performed using forming a frustum geometry (see Fig. 2a). Each component was formed until failure: the crack criterion was defined as the end of forming. Given this, the crack happening during forming is the very criteria for establishing the forming limit. Fig. 3 shows a failed specimen. Two different tool tips were used (spherical and flat), with different tip diameters and corner radiuses. Fig. 2b shows the schematic drawing of the tools, and Table 5 specifies the dimensions of the spherical and flat tools. The experiments were performed on a SIEMENS Topper TMV-510T 4-axis CNC milling machine: Fig. 4 shows the CNC table with a rapid clamping rig. The full forming processes were carried out using the same parameters: a 1500 mm/min feed rate and a 2000 rpm spindle speed were applied. A constant step-down of a value of 0.05 mm was used, as the application of a smaller step-down would have resulted in better geometric accuracy and surface finish of the SPIF components [47, 48]. To increase the reliability of the measurements, each sample was formed three times experimentally; the total number of formed components were thus 108. The data collected from these samples (108) were used as an actual dataset (input and output) for prediction; process parameters were used as inputs, and the obtained results of geometric accuracy and formability (maximum depth) were used as output arguments of the ANN predicted model. In regard to the above conditions (see Table 13 in the Appendix), which lists the raw data of the 108 components formed using experimental SPIF. In each forming process, the surface roughness of the tool was measured prior to and after forming. The value of the surface roughness of the forming tool before the forming process served as the adopted input value of the formed product. As for this value, tool surface roughness following the forming process was taken as input for the subsequent forming process and so on. This method was applied to all tools used in this research. Nevertheless, due to the wear on the polymer tool surface caused by forming, a new polymer-forming tool was used in each forming process, and each polymer tool's surface roughness was measured before the start of the process. Formed part profiles were measured using a Mitutoyo

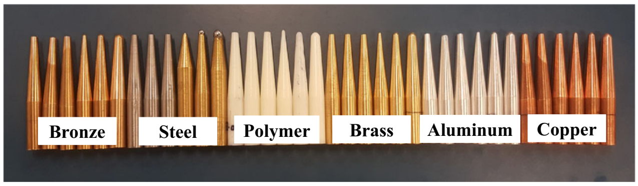

Fig. 1 Tool materials

Table 2 Chemical composition of blank material

Al	Si	Fe	Cu	Mn	Mg	Zn	Cr	Ni	Others
96.90	0.201	0.448	0.212	0.807	1.260	0.071	0.022	0.006	0.073

Table 3 Mechanical properties of the metallic tool

Material	Tensile strength Rm-MPa	Yield Strength Rp 0.2-MPa	Brinell hardness-HB
Steel (C45)	700	490	223
Brass (CuZn39Pb3)	500	390	186
Bronze (CuSn8)	450	300	135
Copper (E-Cu57)	395	365	88
Aluminum (AlMgSi 0.5)	215	160	73

Coordinate Machine (see Fig. 5). The average deviation along the wall was considered as the value of geometrical accuracy for every 3 SPIF components, and the components were formed under the same process conditions. Depth was measured using a Mitutoyo Digimatic Height Gauge with a maximum jaw distance of 12″/300 mm and an uncertainty value of x .0005″/0.01 mm. Forming depth was measured between the bottom part and the upper sheet surface and is expressed as the distance of the jaws, as shown in Fig. 6.

The measurement data were analyzed to obtain formability and geometric accuracy values. In the scope of this, formability constituted the forming depth of the component, while accuracy was understood as the deviation of the wall radius from the designed CAD model. Accuracy was thus expressed in the form of a comparison between the real wall radius obtained by the Mitutoyo Coordinate Machine and the wall radius of the CAD model, which was 25 mm. The wall radius is shown in Fig. 3.

4 Artificial neural networks

It has often been claimed that Warren McCulloch and Walter Pitts' seminal study of the 1940s introduced the neural network concept. Their original view of neural networks showed that neural networks could compute any function of logic or mathematical formula. In the late 1950s, the innovation of the perceptron network, i.e., the first practical application of artificial neural networks, was introduced [49]. Recently, thousands of papers have been published in connection with neural networks and have been used in various sciences. Such uses include applications by artists, filmmakers, musicians, scientists, and particularly by researchers in order to produce useful and often creative results. Artificial Neural Network (ANN) topology could be determined based on the number of layers (input and output layer(s)), as well as on the transfer function of these layers and the number of neurons in each layer [50]. Any ANN structure has input and output layers and also features a minimum of one hidden layer. There are several neurons in each layer, and they exhibit a transfer function: this allows the transfer of weight backward and forward [51]. In this study, for the ANN model, the backpropagation learning algorithm was used, which is called "multilayer perceptron" (MLP) or "multilayer feed-forward." The concept of the MLP originated from Werbos 1974, and Rumelhart et al. 1986 [52]. Equation 1 expresses the multilayer perceptron as follows:

$$y = f(\text{net}) = f\left(\sum_{i=1}^{n} w_i x + b\right) \tag{1}$$

where y is the output and x is input, w_i are the weights, and b is the bias [53].

In order to predict the actual data obtained from the components formed by SPIF, two different structures of the ANN model were built using the Neural Network Toolbox™ of MATLAB [54]. Both structures had the same number of inputs: ten (different tool materials, tool shapes, tool end/corner radiuses, and tool surface roughness values (*Ra* and *Rz*)). The tools were classified into two groups based on their shapes (flat and hemispherical) for the purpose of checking the effect of the tool shape on the accuracy and formability of components. Furthermore, each shape was divided into three sections based on the corner radius (*r*) of the flat tool and the tip radius (*R*) so that the effectiveness of these factors on the above-mentioned outputs could be assessed. Each structure had one hidden layer with ten neurons. For the experiments, different training and transfer functions were trained (see Sections 4.2 and 4.3). The main difference between the structures was the number of outputs, which also affected the number of neurons in the output layer, as shown in the pictorial representation of Fig. 7a and b. In the scope of the study, the learning rate was

Table 4 Polymer properties

Polymer	Density (g/cm^3)	Elastic Modulus (MPa)	Tensile strength (MPa)	Elongation at break %	Shore D Hardness
VeroWhitePlus, (RGD835)	1.19	2,500	58	25	85

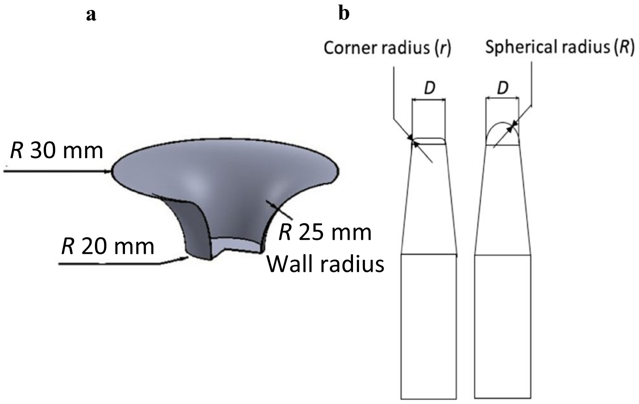

Fig. 2 **a** Frustum geometry. **b** Tool schematic

0.01, the performance goal was 0.001, and the number of epochs was 1000.

4.1 One-hot encoding

One-hot encoding is the common method of describing categorical variables, also known as dummy variables [55]. The concept behind one-hot encoding is to substitute a categorical variable with one or more new features. Through the replacement of the categorical inputs with values 0 and 1, these categorical inputs will sparse-binarize and can be included as a feature for training the ANN model. In this study, two sets of data were encoded. Furthermore, tool materials and tool shapes were binarized as sparse matrices of 0 and 1. It must be considered that when one material is active as number 1, all other materials are encoded as 0, and so on.

4.2 Training function

In a Neural Network (NN), optimization is a procedure used for training a dataset to tune and for finding a set of network weights to create a good map for prediction. Different optimization algorithms (training functions) can be used in the training process to predict the output from a given input. The training algorithm relies on many factors including, and not limited to, the data set, the number of weights and biases, and the performance goal. Hence, selecting the proper training algorithm to be the fastest and best in the scope of a prediction is a challenging task. With this in mind, various types of training-function "learning algorithms" were implemented in the scope of this paper for mapping output parameters. Training functions used for that purpose were Levenberg-Marquardt (Trainlm), Conjugate Gradient Backpropagation with Powell-Beale Restarts (Traincgb), Resilient Backpropagation (Trainrp), Bayesian Regularization Backpropagation (Trainbr), BFGS Quasi-Newton (Trainbfg), and Scaled Conjugate Gradient (Trainscg). Levenberg-Marquardt is faster compared to other training functions and more agile; and likewise, the BFGS Quasi-Newton algorithm is also quite fast [54].

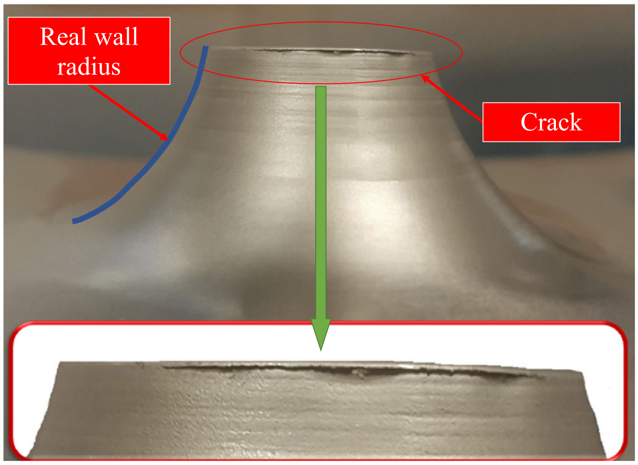

Fig. 3 Failed specimen

Table 5 Tool geometry details

Tools	Geometry Corner radius r (mm) Spherical radius R (mm)	D (mm)
Flat end	$r = 0.1$	4
	$r = 0.3$	4
	$r = 0.5$	4
Hemispherical end	$R = 1$	2
	$R = 2$	4
	$R = 3$	6

4.3 Transfer function

There are different types of transfer functions, and selecting an appropriate one depends on many factors: particularly the type

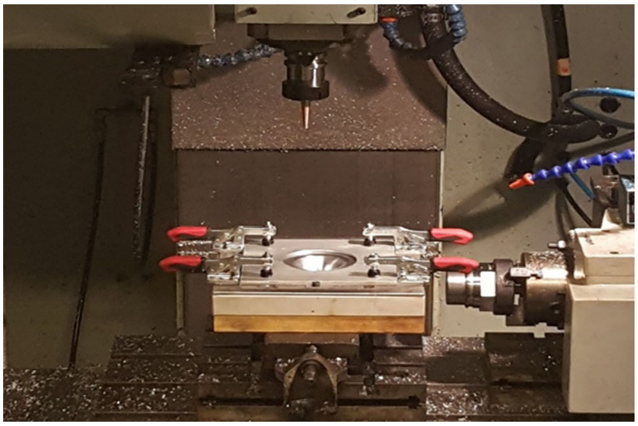

Fig. 4 Rapid clamping rig on the CNC milling table

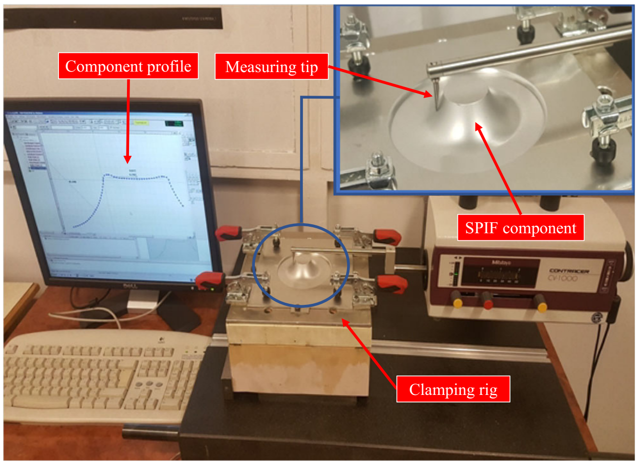

Fig. 5 Mitutoyo coordinate machine

of ANN. In a NN, the sums of each layer are weighted, and the summed weights undergo a transfer function. Finally, transfer functions calculate a layer's output from the summed weights that entered a layer. Usually, Log-sigmoid (Logsig) is used in multilayer networks; other functions such as Tan-sigmoid (Tansig) can be an alternative, and this latter is usually used for pattern recognition problems [54]. Nevertheless, in this study, various types of transfer functions were executed, and different training functions were conducted to improve prediction accuracy. Ultimately, the Purelin transfer function was selected for the output layer. Table 6 lists the algorithm of the transfer function used for the purpose of this study.

4.4 Dataset distribution

The historical data of formed components can be used as inputs in order to predict the expected outcome of forming without performing any new process of forming. Actual data must be divided into different subsets: i.e., training, validation, and testing datasets. The performance of any model can be significantly affected by the splitting of the dataset into training and testing data. Shahin [56] claimed that there is no clear relationship between the ratio of the data of different subsets. Zhang et al. [57] describe that one of the primary dataset problems is the dividing ratio, and this problem has no general setting as a solution. Based on their survey, the researchers divided their datasets in line with a different ratio of subsets. The most extensively used ratios are 90% vs. 10%, 80% vs. 20%, or 70% vs. 30%. In fact, unbalanced subsets negatively affect model performance. In the trials conducted in the scope of this paper, optimal prediction resulted from the data of subsets 80% vs. 20% of the actual data concerning training and testing datasets, respectively. To assure that the model learned and assessed all data samples, the dataset of the training had to be divided into validation and test subsets. Accordingly, the training dataset, which is 80% of the whole dataset, was divided into 90% for training, 5% for validation and the remaining 5% for testing. It should be noted that the testing dataset (20%) did not include the training dataset, which was stored for final testing purposes. Concerning the actual dataset, there is 108 rows extracted from the experiments of forming the given sheet using SPIF, and these rows were used as training and testing datasets.

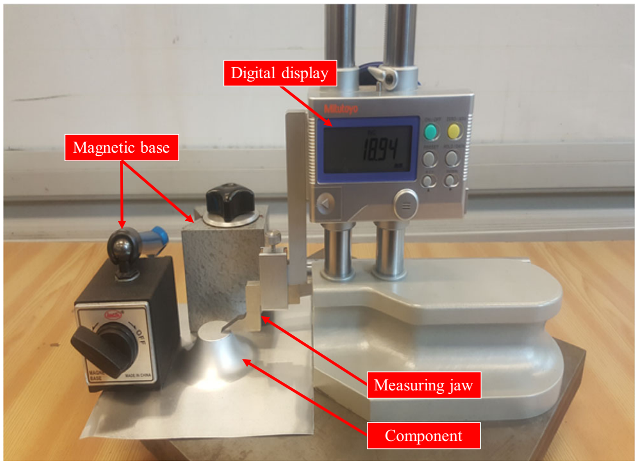

Fig. 6 Mitutoyo digimatic height gage

4.5 Overfitting

Overfitting happens if the model takes into consideration variables such as the noise or random fluctuations of the trained data as learning data and considers it as one of the model's concepts. As a result, this function affects new data saved for testing purposes. Consequently, this concept produces a model that yields good performance on the training dataset but does not perform so effectively on a new sample dataset used as a test sample. In order to ensure that the trained model did not exhibit features of overfitting, 20% of the real data was saved for the model's final testing. In this set-up and for the purpose of preventing overfitting, regularization discouraged the learning of a more complex or flexible model. Another solution to reduce overfitting is to reduce the complexity of a NN model. One possible method of improving network generalization is to adjust the value of weights by changing network parameters. As a consequence, controlling the complexity of a model is achieved through the use of regularization [58]. The second method of improving generalization is called early stopping. Using this technique, training data, which constitutes 80% of the entire actual data set, is divided into three subsets: training, validation, and test subsets. The training data set is employed to compute the gradient and modernizing weights and biases of the network, whereas the validation set is for monitoring the error as the training process runs. The third subset is the test set, which plots the error

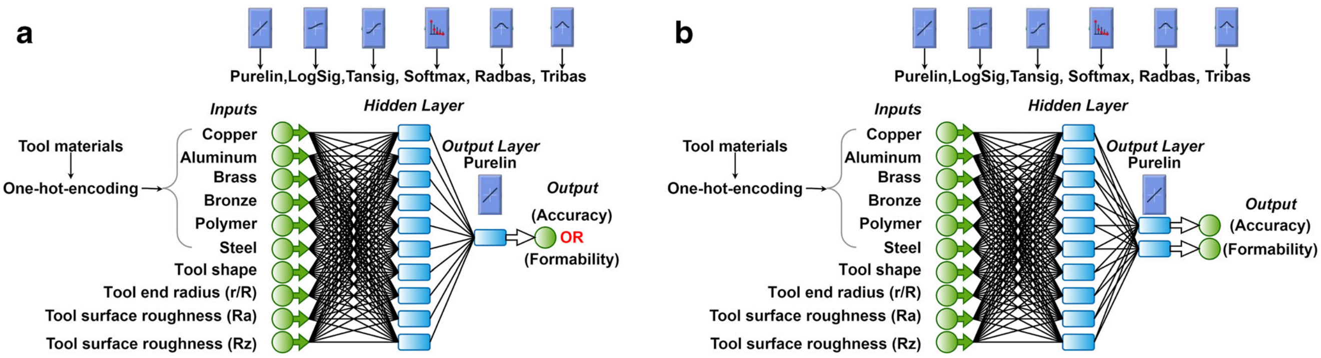

Fig. 7 Different ANN structures: **a** one output; **b** two outputs

during the training process. However, if the overfitting of data commences during the training, the number of errors will increase in the validation set. If the validation error rises above a specified number of iterations, the training stops, and the weights and biases return to the smallest validation error [59].

5 Investigation of accuracy

There are many different metrics for validation but using the proper validation metric is an important consideration in the evaluation and improvement of model performance. In this study, different structures and various training and transferring algorithms were compared and validated. The criteria of validation consist in minimizing error. The coefficient of determination (R^2) and adjusted determination (*adj.* R^2) are used for checking the models and structures in question since an (R^2) value close to 1 implies good performance. Moreover, Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) were also used for validation. In fact, RMSE is more sensitive to error if MAE is more stable. In fact, RMSE and MAE have better evaluation metrics compared to (R^2) due to the latter's limitation listed in [60]. Better performance of the model is indicated by a situation where MAE and RMSE values are close to 0. Even so, the significant variance between RMSE and MAE values means large variations in error distribution. In this study, Mean Relative Error (MRE) was used to measure the precision of the model, as it is clear that absolute error is the magnitude of the difference between the actual and predicted

Table 6 Details of the transfer function

Transfer function	Abbreviation	Graph	Algorithms
Linear	Purelin		$f(x) = purelin(x) = x$
Log-sigmoid	Logsig		$f(x) = logsig(x) = 1 / (1 + exp(-x))$
Hyperbolic tangent sigmoid	Tansig		$f(x) = tansig(x) = 2/ (1+exp(-2* x))-1$
Softmax	Softmax		$f(x) = softmax(x) = exp(x)/sum(exp(x))$
Radial basis	Radbas		$f(x) = radbas(x) = exp (-x \text{^}2)$
Triangular basis	Tribas		$f(x) = tribas(x) = 1 - abs(x), \text{ if } -1 <= x <= 1$ $= 0, \text{ otherwise}$

where x is the weighted sum of w_i, b, and y of Eq. 1.

values; hence, the relative error is the absolute error divided by the magnitude of the actual value. The distribution of prediction values was validated via Standard Error Mean (SEM), where SEM is the standard deviation of the sampling distribution of the sample mean. In other words, the variance of the sample means is inversely proportionate to the sample size, and SEM is the original Standard Deviation (SD) of the sample size over the square root of the sample size. It is worth pointing out that the sample size is 108 for the entire dataset, while 86 for training, and 22 for testing. Error (E), Mean Error (ME), Mean Square Error (MSE), and Standard Deviation (SD) were applied for deriving the validation equations, as mentioned earlier; (R^2) and (*adj.* R^2) were derived from the Total Sum of Square (SS_{tot}) and the Sum of the Square of Residuals (SS_{res}). The validation equations can be represented as follows:

$$E = \left(y_i^{\text{actual}} - y_i^{\text{predict}}\right) \tag{2}$$

$$ME = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\text{actual}} - y_i^{\text{predict}}\right) \text{ or } \mathrm{ME} = \frac{1}{n}\sum_{i=1}^{n}(E) \tag{3}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left(\left|y_i^{\text{actual}} - y_i^{\text{predict}}\right|\right) \text{or } \mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}(|E|) \tag{4}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\text{actual}} - y_i^{\text{predict}}\right)^2 \text{ or } \mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(E)^2 \tag{5}$$

$$RMSE = \sqrt{\mathrm{MSE}} \tag{6}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left(\left|\frac{y_i^{\text{actual}} - y_i^{\text{predict}}}{y_i^{\text{actual}}}\right|\right) \text{ or } \mathrm{MRE} = \frac{1}{n}\sum_{i=1}^{n}\left(\left|\frac{E}{y_i^{\text{actual}}}\right|\right) \tag{7}$$

$$\overline{E} = \left(y_i^{\text{predict}} - y_i^{\text{actual}}\right) \tag{8}$$

$$\overline{ME} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\text{predict}} - y_i^{\text{actual}}\right) \tag{9}$$

$$SD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\overline{E} - \overline{\mathrm{ME}}\right)^2} \tag{10}$$

$$SEM = \frac{SD}{\sqrt{n}} \tag{11}$$

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\text{actual}}\right) \tag{12}$$

$$SS_{tot} = \sum_{i=1}^{n}\left(y_i^{\text{actual}} - \overline{y}\right)^2 \tag{13}$$

$$\mathrm{SS}_{\text{res}} = \sum_{i=1}^{n}\left(y_i^{\text{actual}} - y_i^{\text{predict}}\right)^2 \text{ or } \mathrm{SS}_{\text{res}} = \sum_{i=1}^{n}(E)^2 \tag{14}$$

$$R^2 = \frac{\mathrm{SS}_{tot} - \mathrm{SS}_{res}}{\mathrm{SS}_{\text{tot}}} \tag{15}$$

thus:

$$R^2 = \frac{\sum_{i=1}^{n}\left(y_i^{\text{actual}} - \overline{y}\right)^2 - \sum_{i=1}^{n}\left(y_i^{\text{actual}} - y_i^{\text{predict}}\right)^2}{\sum_{i=1}^{n}\left(y_i^{\text{actual}} - \overline{y}\right)^2} \tag{16}$$

$$\overline{\overline{y}} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{\text{predict}}\right) \tag{17}$$

$$adj.R^2 = 1 - \left(\frac{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{actual} - y_i^{predict}\right)^2}{\left(\frac{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{predict} - y\right)^2}{n-1}\right)}\right) or\ adj.R^2 = 1 - \left(\frac{\mathrm{MSE}}{\left(\frac{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{predict} - y\right)^2}{n-1}\right)}\right) \tag{18}$$

6 Contribution analysis of input variables

There are different methods to assess the contribution of each input variable on ANN outputs. For the generation of a dependent variable, this paper utilized Garson's algorithm [61] to determine the relative importance (RI) of the various inputs as a predictor of predicted outputs. The Garson method was also used in different studies as underpinned by [62–67]. The algorithm, as shown in Eq. 19, is based on the connection weights of the neurons. Goh [68] applied the Garson algorithm and claimed that RI estimation requires partitioning of the hidden output weights into elements connected to each neuron in the input layers.

$$RI(\%) = \left[\sum_{j=1}^{n_h}\left(\frac{y_{vj}/\sum_{k=1}^{n_v} y_{kj}\Big)\, O_j\Big]}{\sum_{y=1}^{n_v}\left[\sum_{j=1}^{n_h}\left(y_{vj}/\sum_{k=1}^{n_v} y_{kj}\right) O_j\right]}\right.\right. \tag{19}$$

where:

n_v number of neurons in the input layer,

n_h number of neurons in the hidden layer,

y_j absolute value of connection weights between the input and the hidden layers,

O_j absolute value of connection weights between the hidden and the output layers.

7 Results and discussion

For the purpose of inspecting the accuracy of different ANN models and for producing comparisons between the two structures, errors of predicted results were analyzed to determine the ANN's performance. The errors extracted from the results were subjected to various validation metrics.

7.1 One-output structure

7.1.1 Prediction of accuracy

Table 7 illustrates two validation metrics used for checking the one-output ANN structure (the prediction of the accuracy of

Table 7 Two validation metrics for checking the ANN structure used for predicting accuracy (one-output structure)

Training function		BFGS Quasi-Newton (Trainbfg)					
Transfer function		Logsig	Purelin	Radbas	Softmax	Tansig	Tribas
Validation metrics	MSE	0.2734	3.6239	0.3960	0.5797	0.3889	2.8790
	R^2	0.9834	0.7796	0.9759	0.9648	0.9764	0.8249
Training function		Bayesian Regularization Backpropagation (Ttrainbr)					
Validation metrics	MSE	0.3804	3.6181	0.4926	0.7858	0.4278	1.6239
	R^2	0.9769	0.7800	0.9700	0.9522	0.9740	0.9013
Training function		Conjugate Gradient Backpropagation (Traincgb)					
Validation metrics	MSE	0.2067	3.6055	0.2992	0.4847	0.2877	1.1252
	R^2	0.9874	0.7807	0.9818	0.9705	0.9825	0.9316
Training function		Levenberg-Marquardt (Trainlm)					
Validation metrics	MSE	0.1968	3.5999	0.2602	0.4523	0.2162	1.3350
	R^2	0.9880	0.7811	0.9842	0.9725	0.9869	0.9188
Training function		Resilient Backpropagation (Trainrp)					
Validation metrics	MSE	0.4162	3.6117	0.3062	0.4429	0.3992	0.6687
	R^2	0.9747	0.7804	0.9814	0.9731	0.9757	0.9593
Training function		Scaled Conjugate Gradient (Trainscg)					
Validation metrics	MSE	0.3558	3.6056	0.1503	0.4779	0.3172	0.9182
	R^2	0.9784	0.7807	0.9909	0.9709	0.9807	0.9442

SPIF components). It can be seen from the data that Scaled Conjugate Gradient (Trainscg) achieved the best values: near to zero concerning MSE and up to 1 concerning R^2.

From the data shown in Fig. 8a, it is clear that there is a significant disparity ($MSE = 0.1503$, $R^2 = 0.9909$) between the different transfer functions in favor of Radbas when the Trainscg training function is run. It can be seen from Fig. 8b that R^2 and MSE differed slightly when the Radbas transfer function with various training functions was used, but the highest R^2 and smallest MSE were found in the case of Trainscg. With reference to this, Fig. 8 exhibits a one-output structure of the ANN used for predicting the accuracy value of SPIF components.

7.1.2 Prediction of formability

Regarding the prediction of formability via the one-output ANN, the differences between training and transfer functions are highlighted in Table 8. There is a significant positive correlation between Trainbfg and Logsig. Therefore, the results indicate that the smallest (*MSE*) is 0.0351, and the largest (R^2) is 0.9860. Concerning formability, the same values of (*MSE*)

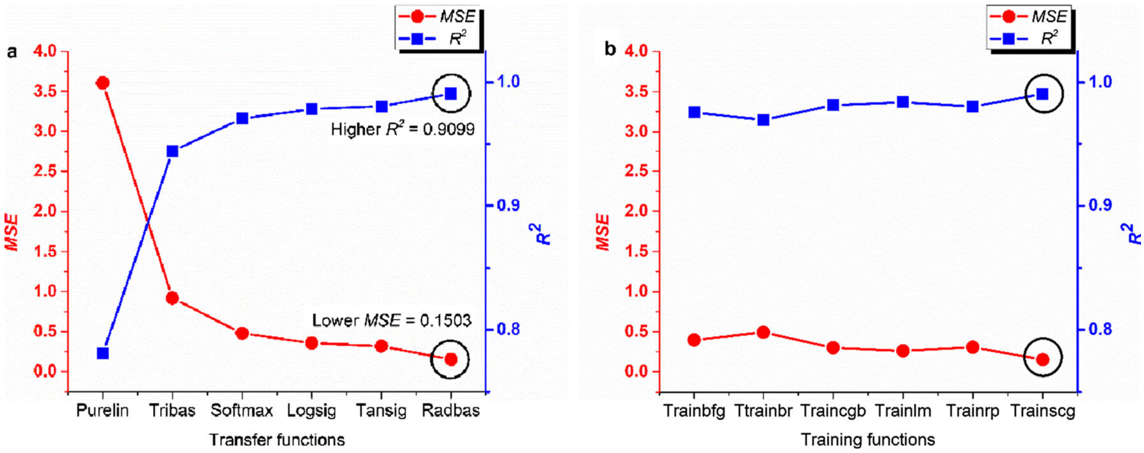

Fig. 8 MSE and R^2 values for predicted results in the case of the ANN model: one-output structure for predicting accuracy of SPIF components. **a** Various transfer functions using Trainscg. **b** Various training functions using Radbas

Table 8 Two validation metrics for checking the ANN structure used for predicting formability (one-output structure)

Training function		BFGS Quasi-Newton (Trainbfg)					
Transfer function		Logsig	Purelin	Radbas	Softmax	Tansig	Tribas
Validation metrics	MSE	0.0351	1.0904	0.4904	0.4595	0.0988	1.2607
	R^2	0.9860	0.5648	0.8043	0.8166	0.9606	0.4968
Training function		Bayesian Regularization Backpropagation (Ttrainbr)					
Validation metrics	MSE	0.6014	1.1936	0.5658	0.5646	0.5604	0.6901
	R^2	0.7600	0.5236	0.7742	0.7747	0.7763	0.7246
Training function		Conjugate Gradient Backpropagation (Traincgb)					
Validation metrics	MSE	0.1029	1.0892	0.5076	0.3966	0.1423	0.6756
	R^2	0.9589	0.5653	0.7974	0.8417	0.9432	0.7304
Training function		Levenberg-Marquardt (Trainlm)					
Validation metrics	MSE	0.0972	1.0871	0.0821	0.2877	0.3431	0.5794
	R^2	0.9612	0.5661	0.9672	0.8852	0.8631	0.7687
Training function		Resilient Backpropagation (Trainrp)					
Validation metrics	MSE	0.2660	1.0887	0.4272	0.3933	0.3815	0.5292
	R^2	0.8938	0.5655	0.8295	0.8430	0.8477	0.7888
Training function		Scaled Conjugate Gradient (Trainscg)					
Validation metrics	MSE	0.0626	1.0901	0.3778	0.4728	0.1586	0.6666
	R^2	0.9750	0.5649	0.8492	0.8113	0.9367	0.7340

and (R^2) as the ones in Fig. 8 can be established based on the values listed in Table 8.

7.1.3 Assessment of the best ANN models for predicting accuracy and formability in the case of the one-output structure

Table 9 presents the setout scenario for validating the ANN model: the use of Trainscg for predicting accuracy and the use of Trainbfg for predicting formability vis a vis the application of the Radbas and Logsig transfer functions, respectively. An overview of all metrics values of the validation was used for comparing the various training and transfer functions; this process is listed in the Appendix as Tables 14 and 15. Such values are related to the one-output ANN structure and were used to predict the accuracy and formability of SPIF parts. Together these results provide valuable insights and suggest the following: positive error means that the predicted value is larger than the actual value, and a negative error means that the predicted value is lower than the actual value.

Figure 9a and b compares two trials of prediction concerning accuracy and formability in addition to their variation with the actual data obtained from SPIF experiments. There is a clear trend of fitting between predicted values and real data. Moreover, Fig. 10a and b clearly shows a significant positive correlation between predicted and actual datasets.

7.2 Two-output structure

Two discrete analyses emerge by comparing the values of ANN in terms of output numbers. First, the rate of the smallest (MSE) was obtained during accuracy prediction: 99.8611% vs. 0.1389% for one and two outputs, respectively. Second, regarding formability, 99.7778% vs. 0.2222% was the (MSE)

Table 9 Assessment of the best ANN models in the case of different validation metrics (one-output structure)

Trainscg training function with Radbas transfer function for predicting accuracy								
Validation metrics	ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj*-R^2
Accuracy	− 0.0027	0.2005	0.1503	0.3876	0.0367	0.3894	0.0375	0.9909	0.9909
Trainbfg training function with Logsig transfer function for predicting Formability								
Validation metrics	ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj*-R^2
Formability	− 0.0131	0.1077	0.0351	0.1874	0.0066	0.1878	0.0181	0.9860	0.9859

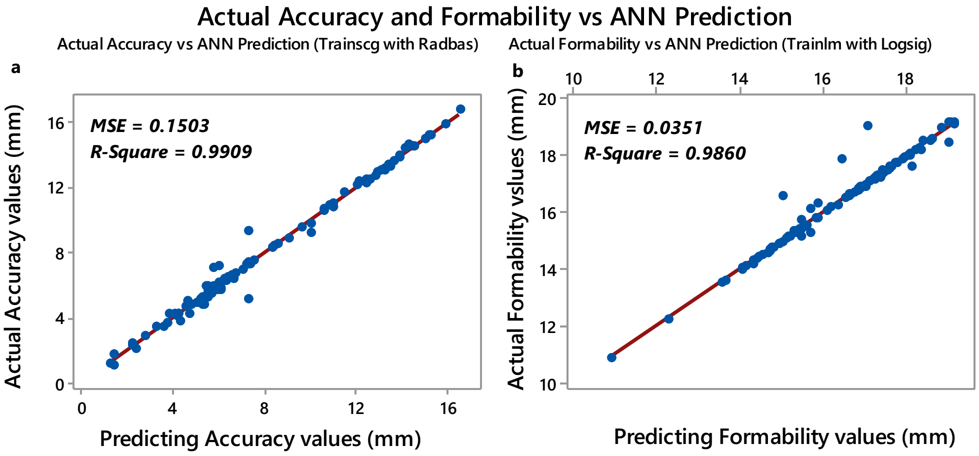

Fig. 9 Actual accuracy and formability vs. ANN prediction accuracy

rate for one and two-output structures, respectively. However, there are several possible explanations of the poor results that originate from two-output structures as compared to the one-output structure. One main reason for these results may be the fluctuation of each output, which are located far from one another, and there is also an enormous difference between the inputs. This is explained by the fact that accuracy values are between 1 and 17 mm, and formability values (maximum depth) are between 10 and 20 mm. The most striking result emerging from the data is that the best prediction is obtained in the two-output structure of Trainlim with Logsig. For the sake of clarity, Table 10 shows the errors of this two-output structure model. Furthermore, the full results are shown in the Appendix as Table 16 and Table 17

7.3 Training and testing assessment of the best ANN models for the one-output structure

A comparison of the results reveals that the suggested structure of ANN is the one-output argument structure, which concurrently utilizes the Trainscg training function and the Radbas transfer function. This scenario offered the best prediction of accuracy. It is important to note that Trainbfg and Logsig emerged as a reliable method of the prediction of formability. Therefore, these results were to be interpreted in the light of numerous analyses and details. Consequently, the actual data were divided into two major sets: training and testing, with values of 80% with 20%, respectively, as discussed in Section 4.4. The suggested structure and model were run on the training datasets and were tested for prediction using a test dataset. Table 11 lists the errors and validation metrics of training and testing prediction resulting from the model, as described above.

The accuracy model was also successful in prediction, but its operation was coupled with an insignificant decrease in performance. Here, one unanticipated finding was that the accuracy of the prediction of formability decreased considerably. An emerging issue from these findings is that a reduction in the sample size leads to an increase in error and causes a decrease in performance. Moreover, the one-output ANN

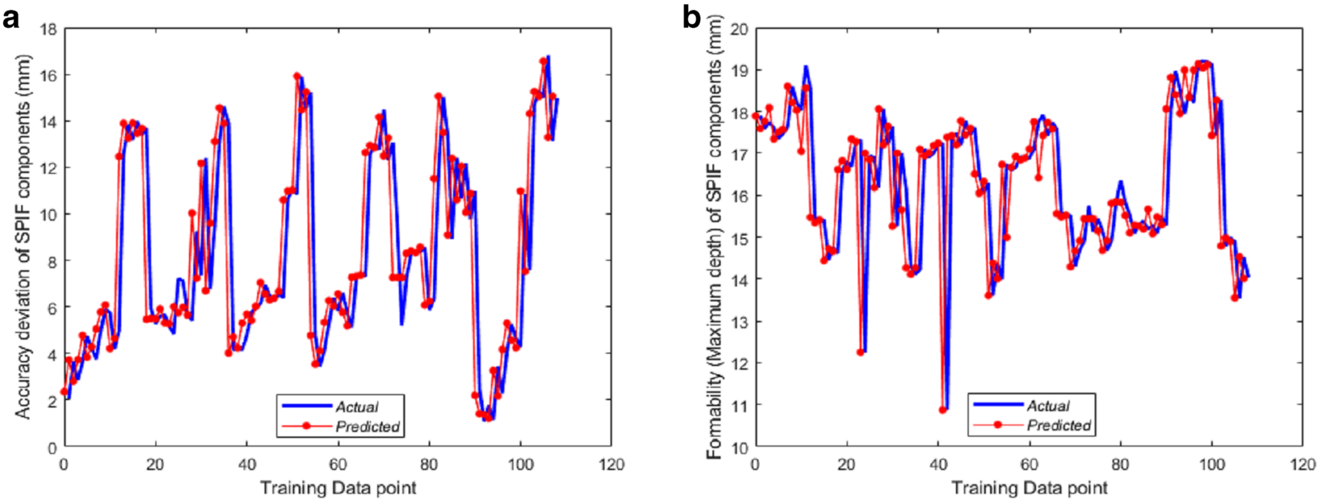

Fig. 10 Actual and predicted datasets, **a** accuracy; **b** formability

Table 10 Assessment of the best ANN models in the case of different validation metrics (two-output structure)

Two-output structure—Trainlim training function with Logsig transfer function

Validation metrics	ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj*-R^2
Accuracy	0.0268	0.3123	0.2868	0.5355	0.0656	0.5373	0.0517	0.9826	0.9824
Formability	− 0.0098	0.2403	0.2964	0.5444	0.0158	0.5469	0.0526	0.8817	0.8643

model cannot be extrapolated to different sample sizes. In addition, it is important to bear in mind the presence of possible bias in these responses. Consequently, various models were investigated in an attempt to find an alternative model capable of predicting formability more accurately. As a result, the alternative model using Traincgb with Logsig was found to be capable of successfully predicting formability, as shown in Table 12.

7.4 Analytical equations to predict the accuracy and formability of SPIF

For finding an alternative method of predicting formability and accuracy in an easy and more accurate way, rather than having to build a NN model each time, analytical equations that could predict the accuracy and formability of SPIF were envisaged to be used in place of the approved network. Therefore, two equations (23 and 27) were established, which required constant weights and biases extracted from the recommended ANN network. The ANN network tuning provided the weights and biases necessary for achieving the best prediction. In this study, due to the fact that only one hidden layer was applied, there was only one set of input weight (IW) and layer weight (LW). The IW is between the inputs and the hidden layer, and the LW is situated between the hidden layer and the output layer. The biases for each layer are b1 and b2.

Tables 18 and 19 in the Appendix provide (b_1), (b_2), (IW), and (LW) obtained from the best trained ANN model regarding accuracy (as shown in Table 18 in the Appendix) and formability (as shown in Table 19).

$$f(x) = Radbas(x) = \exp\left(-x^2\right) \tag{20}$$

$$Accuracy_i^{\text{predict}} = b_2 + LW \times Radbas(b_1 + IW \times x) \tag{21}$$

$$Accuracy_i^{\text{predict}} = b_2 + LW \times \exp\left(-(b_1 + IW \times x)^2\right) \tag{22}$$

$$\text{Accuracy}_i^{\text{predict}} = 3.9206 + \quad [LW] \times exp\left(- \left(\begin{bmatrix} -5.1413 \\ 1.2616 \\ 0.6642 \\ 2.7010 \\ 1.5268 \\ -2.5527 \\ 0.9677 \\ 0.9160 \\ 2.3191 \\ -4.4413 \end{bmatrix} + \begin{bmatrix} -1.1773 & -0.7032 & -1.7944 & -1.8489 & 1.2188 & 2.0108 & -0.6296 & 1.9369 & 0.7717 & 0.3779 \\ -1.7346 & -1.1106 & 0.1240 & 0.7292 & 0.0289 & -0.3600 & 0.6209 & -2.3675 & -0.6774 & -1.6394 \\ -0.4661 & 1.3360 & -0.1381 & -1.3452 & -2.0399 & -0.3382 & -1.2443 & 3.8528 & 0.3032 & 1.1052 \\ -0.0229 & 0.2218 & -1.2916 & 2.9920 & -4.6045 & 3.0664 & 1.2266 & -2.3559 & 2.6646 & 4.1350 \\ 1.9534 & -2.5503 & 1.0572 & 2.3674 & -1.2938 & 2.2008 & -0.7986 & -2.0592 & 0.8307 & 0.2103 \\ -2.3043 & -1.3088 & -1.9797 & 2.4409 & 0.0598 & 0.7101 & -0.6936 & 0.8617 & -1.1759 & 0.2284 \\ 0.0759 & 1.0141 & 1.6768 & -0.7190 & 1.8965 & -1.7534 & -0.7307 & 2.1473 & -1.5098 & 0.2528 \\ -0.2273 & -0.7655 & -0.1006 & -0.3139 & 1.1390 & 2.3990 & -0.9744 & -1.6789 & -0.5358 & -2.2392 \\ 0.3222 & -0.2106 & -0.1155 & 1.1132 & 2.3911 & -0.5010 & -1.7080 & 0.0188 & 0.4250 & -0.6822 \\ 0.1749 & -0.6833 & -0.8397 & 1.7357 & -1.2086 & -0.6854 & -0.8242 & 1.5373 & -1.4357 & 0.2100 \end{bmatrix} \times \begin{bmatrix} \text{x} \\ \text{Tool Materials} \\ \text{Tool Shapes} \\ \text{Tool end Radius} \\ \text{Tool surface Ra} \\ \text{Tool surface Rz} \end{bmatrix} \right)^2 \right) \tag{23}$$

Table 11 Assessment of the best ANN models (training and testing) in the case of different validation metrics (one-output structure)

Accuracy (Trainscg with Radbas)

Validation metrics	ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj*-R^2
Training	− 0.0230	0.4111	0.3486	0.5904	0.0788	0.5934	0.0640	0.9795	0.9792
Testing	0.0393	0.7714	1.1081	1.0527	0.1104	1.0767	0.2295	0.9171	0.9370
Formability (Trainbfg with Logsig)									
Validation metrics	ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj*-R^2
Training	0.0139	0.3338	0.4439	0.6663	0.0220	0.6700	0.0723	0.8162	0.7783
Testing	− 0.4356	0.5891	0.8301	0.9111	0.0333	0.8191	0.1746	0.7080	0.4874

Table 12 Assessment of the best alternative ANN models (training and testing) in the case of different validation metrics for predicting formability (one-output structure)

Formability (Traincgb with Logsig)

Validation metrics	ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj*-R^2
Training	− 0.0108	0.2217	0.1570	0.3962	0.0139	0.3984	0.0430	0.9372	0.9327
Testing	0.0854	0.4757	0.3664	0.6053	0.0300	0.6134	0.1308	0.8530	0.8375

where: *LW* = [6.4552 -2.1276 2.7984 7.8625 4.7908 4.3514 -6.4366 -4.2600 2.3087 4.7381]

$$f(x) = Logsig(x) = \frac{1}{(1+\exp(-x))} \tag{24}$$

$$\text{Formability}_i^{\text{predict}} = b_2 + LW \times Logsig(b_1 + IW \times x) \tag{25}$$

$$\text{Formability}_i^{\text{predict}} = b_2 + LW \times \frac{1}{1+\exp(-(b_1 + IW \times x))} \tag{26}$$

$$\text{Formability}_i^{\text{predict}} = -15.9818 + [LW] \times \frac{1}{1+\exp\left(-\left(\begin{bmatrix} -6.2268 \\ 23.0398 \\ 4.0820 \\ 6.1367 \\ -5.2423 \\ 5.0917 \\ -2.2796 \\ -30.7090 \\ 1.6168 \\ 19.9338 \end{bmatrix} + \begin{bmatrix} -3.4709 & 13.2876 & -7.2619 & -16.9449 & -13.0659 & 15.1323 & -2.0860 & 22.3754 & 14.3554 & -1.4070 \\ -12.3566 & 13.6008 & 3.3758 & 0.6860 & 5.9896 & 7.1386 & 6.7221 & -11.7328 & 4.2203 & -26.1003 \\ -8.0432 & 3.0246 & 0.5947 & 5.3230 & -7.8297 & 6.9763 & -1.2762 & -15.8632 & -1.6303 & -0.2870 \\ 1.7346 & 6.3500 & 8.5891 & 2.2943 & -18.7526 & 3.2920 & 15.8267 & -5.0288 & -1.2114 & -0.8662 \\ -0.8079 & -1.3571 & 3.6600 & -1.0886 & -2.3603 & 1.3466 & -10.1771 & -0.6398 & -0.0152 & -0.0121 \\ -3.0961 & -4.1506 & -6.1267 & 4.1379 & 4.7903 & 12.5706 & -8.8120 & -16.4002 & 3.2859 & -8.6912 \\ 8.2305 & -9.0868 & -0.7259 & 4.8521 & -4.1178 & 4.7707 & -6.4077 & 13.9174 & 21.7924 & 9.8806 \\ -1.0452 & 23.8309 & -28.0332 & -11.3384 & -3.0887 & -6.9537 & -9.0903 & -27.3553 & 14.7710 & 37.7944 \\ 17.0348 & -20.7669 & 11.0464 & 4.7830 & -3.2273 & 2.6294 & 16.1645 & 21.9736 & 19.5971 & -24.7539 \\ 9.7667 & -8.6318 & -2.005 & 9.1440 & 18.1230 & -9.7823 & 6.2870 & -9.6843 & -6.1344 & 0.0711 \end{bmatrix} \times \begin{bmatrix} \text{x} \\ \text{Tool Materials} \\ \text{Tool Shapes} \\ \text{Tool end Radius} \\ \text{Tool surface Ra} \\ \text{Tool surface Rz} \end{bmatrix}\right)\right)} \tag{27}$$

where *LW* = [1.7159 1.0223 -1.0223 3.9185 -17.628 16.1231 3.1300 25.4021 25.2140 1.1739]

7.5 Relative importance

With regards to the relative importance and weights analysis of the neural network, it can be seen that the most significant factor in Fig. 11a is the tool-end radius with a RI of 29% concerning accuracy, while tool surface roughness (*Ra* and *Rz*) has the lowest effect. Another interesting observation, based on Fig. 11b, is that the tool-end radius with a RI of 54% is the most influential factor with respect to formability. Secondly, in the prediction process, tool shape and tool surface roughness (*Ra*) with a relative importance of 24% and 28%, respectively, were as important as effects on accuracy and formability. However, concerning accuracy and formability, a striking difference in the RI ratios of the tool-end radius can be noticed. In addition, tool material can also be considered as a powerful factor in determining accuracy.

In the scope of this study, the tools were created by a turning machine under identical machining conditions. However, as mentioned in Section 2, different materials with varying values of hardness were used for producing the tools. Due to that scenario, this resulted in differing values of surface roughness of the

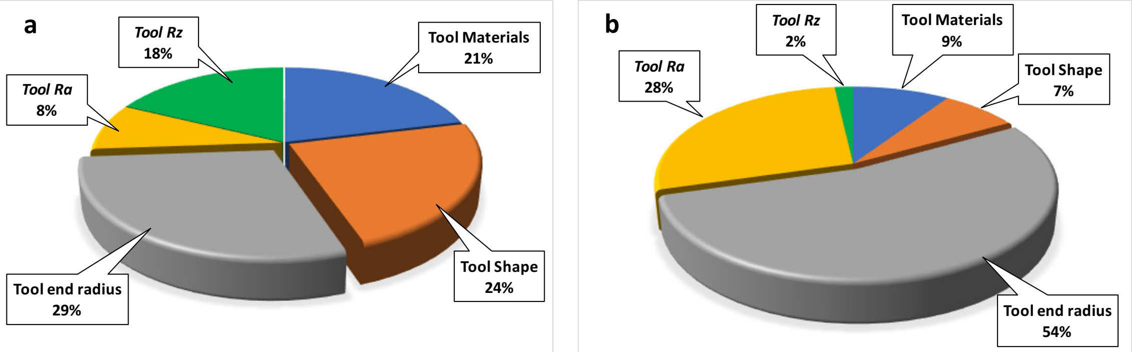

Fig. 11 Relative importance of different input variables, **a** accuracy; **b** formability

tools; and this, in turn, impacted the surface roughness of the final products formed using SPIF. High roughness of the tool tears the wall of the components and causes low depth: this depth is considered a value of formability, as previously mentioned. Similarly to the research presented in the scope of this paper, Hagan and Jeswiet [69] analyzed the influence of several forming variables, such as step-down size and spindle speed on surface roughness in the scope of an ISF process. They claim that tool hardness and its polished surface not only affect depth increment tests but concurrently also result in impacts on the surface roughness of the final product. What Hagan and Jeswiet found explains the change in tool surface roughness identified in the current study: this phenomenon can be explained by changes in formability. As a result of differences in the roughness of the forming tools, differing surface roughness of the products was observed. The latter causes differences in forming depth, which is considered an attribute of formability.

8 Conclusion

This study presented different models and structures of ANN for predicting the accuracy and formability of SPIF components produced from thin aluminum alloy blanks. The main goal of the study was to determine the best model and architecture for that purpose. The most significant finding from the study was that the structure of a one-output solution showed better results than a network with two outputs. The second significant finding was that Trainscg and Trainbfg, as a training function together with Radbas and Logsig as a transfer function, achieved the best prediction concerning accuracy and formability. In the scope of the experiments, the results were assessed using different validation metrics; the highest R^2 values were 0.9909 and 0.9860, and the lowest MSE values were 0.1503 and 0.0351 for accuracy and formability, respectively. In fact, this research project marks the first time that the relative importance (RI) method was used to assess SPIF factors on outputs. RI revealed that the tool end radius is an effective factor that impacts accuracy and formability with values of 29% and 54%, respectively. Tools with different materials exhibit diverse hardness, which causes different surface strains. This also holds true regarding tool geometry. Nevertheless, it is a noteworthy finding to point out that two tools from two different materials with identical geometry will cause different surface strains; consequently, they will have different effects on components. Differing values of hardness result in varying tool tip surface roughness, which thereby affect sheet accuracy and formability. Tool shapes together with tool materials are also influential factors, which impact accuracy by 24% and 21%, respectively. Following the tool radius with a value of 28%, tool surface roughness (*Ra*) is the second most effective factor concerning formability output. Finally, tool surface roughness (*Ra*) and (*Rz*) with values of 8% and 2% for each factor, respectively, were found to be the least influential factor on accuracy and formability.

Author contribution Sherwan Mohammed Najm carried out the experiments, model building, data analysis, and interpretation, and played the lead role in drafting the paper. Imre Paniti provided the concept of the work and carried out supervision, including the critical revision of the paper.

Funding Open access funding provided by Budapest University of Technology and Economics. The research reported in this paper and carried out at Budapest University of Technology and Economics has been supported by the NRDI Fund (TKP2020 NC, Grant No. BME-NC) based on the charter of bolster issued by the NRDI Office under the auspices of the Hungarian Ministry for Innovation and Technology and by the European Commission through the H2020 project EPIC (https://www.centre-epic.eu/) under grant No. 739592. Furthermore, this research has been supported by Hungary's TEMPUS Public Foundation and Stipendium Hungaricum Scholarship.

Data availability Not applicable.

Declarations

Ethical approval and consent to participate Not applicable.

Consent for publication Not applicable.

Conflicts of interest The authors declare no conflict of interest.

Appendix

Table 13 Raw data of the experiments (input and output data)

Tool materials	Tool shapes	Tool radius	Tool arithmetical mean roughness (*Ra*)	Tool ten-point mean roughness (*Rz*)	Responses	
	Flat = 1 Hemispherical = 0	Corner radius *r* (mm) Spherical radius *R* (mm)			Maximum Depth (mm)	Accuracy Deviation (mm)
Copper	1	0.1	0.23	3.76	17.89	2.053
	1	0.1	0.06	0.73	17.59	3.653

Table 13 (continued)

Tool materials	Tool shapes	Tool radius	Tool arithmetical mean roughness (*Ra*)	Tool ten-point mean roughness (*Rz*)	Responses	
	Flat = 1 Hemispherical = 0	Corner radius *r* (mm) Spherical radius *R* (mm)			Maximum Depth (mm)	Accuracy Deviation (mm)
	1	0.1	0.14	2.24	17.74	2.853
	1	0.3	0.32	5.54	17.60	3.645
	1	0.3	0.05	0.65	17.34	4.760
	1	0.3	0.19	3.10	17.47	4.203
	1	0.5	0.13	3.34	17.62	3.763
	1	0.5	0.05	0.45	18.60	4.954
	1	0.5	0.09	1.29	18.21	5.878
	0	1	0.12	3.50	18.03	5.752
	0	1	0.10	1.08	19.09	4.195
	0	1	0.11	2.29	18.56	4.974
	0	2	0.09	0.66	15.47	12.508
	0	2	0.04	0.64	15.35	13.830
	0	2	0.06	0.65	15.41	13.169
	0	3	0.21	1.84	14.45	13.977
	0	3	0.12	2.02	14.76	13.379
	0	3	0.16	1.93	14.61	13.678
Aluminum	1	0.1	0.12	1.16	16.56	5.910
	1	0.1	0.13	1.22	16.80	5.291
	1	0.1	0.12	1.19	16.68	5.601
	1	0.3	0.09	1.04	17.25	5.650
	1	0.3	0.08	0.61	17.32	5.166
	1	0.3	0.06	0.60	12.25	4.837
	1	0.5	0.08	0.64	16.93	7.203
	1	0.5	0.07	0.49	16.90	7.110
	1	0.5	0.04	0.63	16.18	5.707
	0	1	0.47	5.76	18.06	5.424
	0	1	0.22	8.18	17.20	9.248
	0	1	0.34	6.97	17.63	7.336
	0	2	0.33	3.51	15.26	12.396
	0	2	0.06	1.82	17.00	6.760
	0	2	0.20	2.66	16.13	9.578
	0	3	0.36	3.92	14.32	13.101
	0	3	0.12	2.08	14.11	14.602
	0	3	0.24	3.00	14.22	13.852
Brass	1	0.1	0.22	3.50	17.10	4.172
	1	0.1	0.20	2.38	16.90	4.181
	1	0.1	0.21	2.94	17.00	4.177
	1	0.3	0.11	2.04	17.18	4.776
	1	0.3	0.12	1.39	17.24	5.769
	1	0.3	0.12	1.95	10.87	5.917
	1	0.5	0.16	2.00	17.38	6.126
	1	0.5	0.07	1.41	17.47	6.942
	1	0.5	0.02	0.24	17.20	6.468
	0	1	0.40	3.96	17.77	6.246
	0	1	0.13	1.66	17.44	6.533
	0	1	0.27	2.81	17.61	6.390

Table 13 (continued)

Tool materials	Tool shapes	Tool radius	Tool arithmetical mean roughness (*Ra*)	Tool ten-point mean roughness (*Rz*)	Responses	
	Flat = 1 Hemispherical = 0	Corner radius *r* (mm) Spherical radius *R* (mm)			Maximum Depth (mm)	Accuracy Deviation (mm)
	0	2	0.43	3.24	16.52	10.629
	0	2	0.16	4.38	16.06	11.069
	0	2	0.30	3.81	16.29	10.849
	0	3	0.34	3.25	13.60	15.900
	0	3	0.06	1.07	14.38	14.529
	0	3	0.20	2.16	13.99	15.215
Bronze	1	0.1	0.93	11.22	16.73	4.798
	1	0.1	0.27	3.78	16.59	3.445
	1	0.1	0.60	7.50	16.66	4.122
	1	0.3	0.23	8.61	16.92	5.277
	1	0.3	0.18	4.21	16.84	6.386
	1	0.3	0.20	6.41	16.88	5.832
	1	0.5	0.38	6.20	17.10	6.584
	1	0.5	0.13	1.57	17.75	5.753
	1	0.5	0.03	0.24	17.91	5.125
	0	1	0.20	2.06	17.42	7.240
	0	1	0.08	0.67	17.73	7.405
	0	1	0.14	1.36	17.58	7.323
	0	2	0.16	1.43	15.56	12.544
	0	2	0.05	0.57	15.48	12.992
	0	2	0.10	1.00	15.52	12.768
	0	3	0.03	0.32	14.29	14.478
	0	3	0.02	0.21	14.67	12.292
	0	3	0.04	0.40	14.91	13.059
Polymer	1	0.1	0.03	0.36	15.75	9.345
	1	0.1	0.03	0.36	15.13	5.189
	1	0.1	0.03	0.36	15.44	7.267
	1	0.3	0.12	2.35	15.15	8.308
	1	0.3	0.02	0.30	14.68	8.424
	1	0.3	0.07	1.33	14.92	8.366
	1	0.5	0.04	0.61	15.81	8.556
	1	0.5	0.07	0.91	16.35	5.865
	1	0.5	0.05	1.13	15.83	6.324
	0	1	0.27	3.17	15.52	11.687
	0	1	0.32	4.33	15.09	15.020
	0	1	0.30	3.75	15.31	13.354
	0	2	0.21	3.21	15.38	8.927
	0	2	0.09	0.91	15.19	12.400
	0	2	0.15	2.06	15.29	10.664
	0	3	0.19	2.33	15.09	12.166
	0	3	0.08	1.11	15.48	9.760
	0	3	0.13	1.72	15.29	10.963
Steel	1	0.1	0.47	3.71	18.04	2.438
	1	0.1	0.43	5.92	18.97	1.07
	1	0.1	0.45	4.82	18.51	1.754
	1	0.3	0.49	4.94	17.95	1.144

Table 13 (continued)

Tool materials	Tool shapes	Tool radius	Tool arithmetical mean roughness (*Ra*)	Tool ten-point mean roughness (*Rz*)	Responses	
	Flat = 1 Hemispherical = 0	Corner radius *r* (mm) Spherical radius *R* (mm)			Maximum Depth (mm)	Accuracy Deviation (mm)
	1	0.3	0.58	5.92	18.46	3.435
	1	0.3	0.53	5.43	18.21	2.2895
	1	0.5	0.74	5.37	19.17	4.063
	1	0.5	0.61	7.91	19.19	5.256
	1	0.5	0.67	6.64	19.18	4.6595
	0	1	0.07	0.70	19.11	4.279
	0	1	0.08	1.39	17.42	10.863
	0	1	0.07	1.05	18.27	7.571
	0	2	0.04	0.45	14.79	14.688
	0	2	0.06	0.82	14.98	15.294
	0	2	0.05	0.64	14.89	14.991
	0	3	0.04	0.40	13.53	16.8
	0	3	0.06	0.51	14.52	13.13
	0	3	0.05	0.45	14.03	14.965

Table 14 Validation metrics of all ANN models for predicting accuracy (one-output structure)

Training function		BFGS Quasi-Newton (Trainbfg)								
Validation metrics		ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj*-R^2
Transfer function	Logsig	0.0018	0.2982	0.2734	0.5229	0.0509	0.5254	0.0506	0.9834	0.9834
	Purelin	0.0789	1.4223	3.6239	1.9037	0.2423	1.9109	0.1839	0.7796	0.7215
	Radbas	0.0445	0.3702	0.3960	0.6293	0.0699	0.6307	0.0607	0.9759	0.9754
	Softmax	0.0036	0.5027	0.5797	0.7613	0.0794	0.7649	0.0736	0.9648	0.9632
	Tansig	0.0068	0.3640	0.3889	0.6236	0.0621	0.6265	0.0603	0.9764	0.9763
	Tribas	− 0.0362	1.3145	2.8790	1.6968	0.2784	1.7043	0.1640	0.8249	0.7964
Training function		Bayesian Regularization Backpropagation (Ttrainbr)								
Transfer function	Logsig	− 0.0436	0.4199	0.3804	0.6168	0.0719	0.6181	0.0595	0.9769	0.9756
	Purelin	− 0.0052	1.4344	3.6181	1.9021	0.2426	1.9110	0.1839	0.7800	0.7271
	Radbas	− 0.0244	0.4958	0.4926	0.7019	0.0862	0.7047	0.0678	0.9700	0.9685
	Softmax	− 0.0241	0.6550	0.7858	0.8865	0.1061	0.8903	0.0857	0.9522	0.9490
	Tansig	− 0.0143	0.4541	0.4278	0.6541	0.0771	0.6570	0.0632	0.9740	0.9728
	Tribas	− 0.0393	0.9323	1.6239	1.2743	0.1555	1.2796	0.1231	0.9013	0.8797
Training function		Conjugate Gradient Backpropagation (Traincgb)								
Transfer function	Logsig	− 0.0278	0.2600	0.2067	0.4546	0.0412	0.4559	0.0439	0.9874	0.9873
	Purelin	− 0.0337	1.4442	3.6055	1.8988	0.2451	1.9074	0.1835	0.7807	0.7156
	Radbas	0.0126	0.3245	0.2992	0.5470	0.0689	0.5494	0.0529	0.9818	0.9816
	Softmax	0.0313	0.4749	0.4847	0.6962	0.0789	0.6988	0.0672	0.9705	0.9698
	Tansig	0.0198	0.3000	0.2877	0.5364	0.0518	0.5385	0.0518	0.9825	0.9824
	Tribas	0.1015	0.7469	1.1252	1.0608	0.1661	1.0608	0.1021	0.9316	0.9239
Training function		Levenberg-Marquardt (Trainlm)								
Transfer function	Logsig	0.0753	0.1401	0.1968	0.4436	0.0277	0.4392	0.0423	0.9880	0.9878

Table 14 (continued)

Training function		BFGS Quasi-Newton (Trainbfg)								
Validation metrics		ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj-*R^2
	Purelin	− 0.0187	1.4314	3.5999	1.8973	0.2418	1.9061	0.1834	0.7811	0.7222
	Radbas	0.0038	0.3081	0.2602	0.5101	0.0505	0.5125	0.0493	0.9842	0.9844
	Softmax	0.0056	0.4105	0.4523	0.6725	0.0678	0.6756	0.0650	0.9725	0.9722
	Tansig	0.0204	0.1510	0.2162	0.4650	0.0194	0.4667	0.0449	0.9869	0.9870
	Tribas	− 0.0985	0.8055	1.3350	1.1554	0.1314	1.1566	0.1113	0.9188	0.9118
Training function		Resilient Backpropagation (Trainrp)								
Transfer function	Logsig	− 0.0444	0.4074	0.4162	0.6451	0.0655	0.6466	0.0622	0.9747	0.9739
	Purelin	0.0142	1.4284	3.6117	1.9004	0.2399	1.9092	0.1837	0.7804	0.7349
	Radbas	0.0581	0.3355	0.3062	0.5533	0.0639	0.5529	0.0532	0.9814	0.9812
	Softmax	− 0.0032	0.4480	0.4429	0.6655	0.0775	0.6686	0.0643	0.9731	0.9722
	Tansig	− 0.0301	0.3853	0.3992	0.6318	0.0633	0.6340	0.0610	0.9757	0.9746
	Tribas	0.0241	0.5264	0.6687	0.8177	0.0903	0.8212	0.0790	0.9593	0.9582
Training function		Scaled Conjugate Gradient (Trainscg)								
Transfer function	Logsig	0.0208	0.4009	0.3558	0.5965	0.0779	0.5989	0.0576	0.9784	0.9778
	Purelin	− 0.0190	1.4304	3.6056	1.8989	0.2392	1.9076	0.1836	0.7807	0.7314
	Radbas	− 0.0027	0.2005	0.1503	0.3876	0.0367	0.3894	0.0375	0.9909	0.9909
	Softmax	− 0.0274	0.4693	0.4779	0.6913	0.0777	0.6940	0.0668	0.9709	0.9702
	Tansig	0.0030	0.3495	0.3172	0.5632	0.0609	0.5658	0.0544	0.9807	0.9804
	Tribas	0.0012	0.7222	0.9182	0.9582	0.1510	0.9627	0.0926	0.9442	0.9372

Table 15 Validation metrics of all ANN models for predicting formability (one-output structure)

Training function		BFGS Quasi-Newton (Trainbfg)								
Validation metrics		ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj-*R^2
Transfer function	Logsig	− 0.0131	0.1077	0.0351	0.1874	0.0066	0.1878	0.0181	0.9860	0.9859
	Purelin	0.0374	0.7101	1.0904	1.0442	0.0462	1.0484	0.1009	0.5648	0.2314
	Radbas	− 0.0360	0.3395	0.4904	0.7003	0.0218	0.7026	0.0676	0.8043	0.7756
	Softmax	− 0.0289	0.3588	0.4595	0.6778	0.0235	0.6804	0.0655	0.8166	0.7714
	Tansig	− 0.0021	0.2251	0.0988	0.3143	0.0139	0.3157	0.0304	0.9606	0.9584
	Tribas	− 0.0185	0.8515	1.2607	1.1228	0.0542	1.1279	0.1085	0.4968	− 0.1010
Training function		Bayesian Regularization Backpropagation (Ttrainbr)								
Transfer function	Logsig	0.0031	0.4246	0.6014	0.7755	0.0280	0.7791	0.0750	0.7600	0.6613
	Purelin	− 0.3194	0.8159	1.1936	1.0925	0.0513	1.0496	0.1010	0.5236	0.0948
	Radbas	− 0.0451	0.3951	0.5658	0.7522	0.0261	0.7543	0.0726	0.7742	0.6979
	Softmax	− 0.0406	0.4131	0.5646	0.7514	0.0272	0.7538	0.0725	0.7747	0.7103
	Tansig	0.0069	0.3992	0.5604	0.7486	0.0264	0.7521	0.0724	0.7763	0.6903
	Tribas	− 0.0648	0.4767	0.6901	0.8307	0.0311	0.8321	0.0801	0.7246	0.5580
Training function		Conjugate Gradient Backpropagation (Traincgb)								
Transfer function	Logsig	0.0300	0.1619	0.1029	0.3208	0.0098	0.3209	0.0309	0.9589	0.9596
	Purelin	− 0.0408	0.7124	1.0892	1.0436	0.0460	1.0477	0.1008	0.5653	0.2257
	Radbas	− 0.0308	0.3364	0.5076	0.7125	0.0228	0.7151	0.0688	0.7974	0.7442
	Softmax	− 0.0120	0.3283	0.3966	0.6297	0.0214	0.6325	0.0609	0.8417	0.8118
	Tansig	0.0240	0.2632	0.1423	0.3773	0.0162	0.3783	0.0364	0.9432	0.9402

Table 15 (continued)

Training function		BFGS Quasi-Newton (Trainbfg)								
Validation metrics		ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj-R^2*
	Tribas	− 0.0456	0.5358	0.6756	0.8220	0.0347	0.8245	0.0793	0.7304	0.6433
Training function		Levenberg-Marquardt (Trainlm)								
Transfer function	Logsig	− 0.0498	0.0939	0.0972	0.3118	0.0054	0.3093	0.0298	0.9612	0.9606
	Purelin	− 0.0005	0.7116	1.0871	1.0426	0.0461	1.0475	0.1008	0.5661	0.2301
	Radbas	− 0.0126	0.1673	0.0821	0.2866	0.0104	0.2877	0.0277	0.9672	0.9664
	Softmax	− 0.0269	0.2711	0.2877	0.5364	0.0177	0.5382	0.0518	0.8852	0.8694
	Tansig	0.0363	0.2446	0.3431	0.5857	0.0166	0.5873	0.0565	0.8631	0.8408
	Tribas	0.0624	0.3139	0.5794	0.7612	0.0215	0.7622	0.0733	0.7687	0.6987
Training function		Resilient Backpropagation (Trainrp)								
Transfer function	Logsig	0.0096	0.2330	0.2660	0.5158	0.0154	0.5181	0.0499	0.8938	0.8761
	Purelin	0.0280	0.7127	1.0887	1.0434	0.0463	1.0479	0.1008	0.5655	0.2221
	Radbas	− 0.0063	0.2939	0.4272	0.6536	0.0197	0.6566	0.0632	0.8295	0.7969
	Softmax	0.0128	0.2987	0.3933	0.6271	0.0196	0.6299	0.0606	0.8430	0.8129
	Tansig	− 0.0062	0.3209	0.3815	0.6177	0.0211	0.6205	0.0597	0.8477	0.8207
	Tribas	− 0.0078	0.3778	0.5292	0.7275	0.0250	0.7308	0.0703	0.7888	0.7402
Training function		Scaled Conjugate Gradient (Trainscg)								
Transfer function	Logsig	− 0.0091	0.1631	0.0626	0.2503	0.0101	0.2513	0.0242	0.9750	0.9746
	Purelin	0.0145	0.7149	1.0901	1.0441	0.0463	1.0488	0.1009	0.5649	0.2705
	Radbas	0.0009	0.2505	0.3778	0.6147	0.0168	0.6175	0.0594	0.8492	0.8300
	Softmax	0.0023	0.3637	0.4728	0.6876	0.0239	0.6908	0.0665	0.8113	0.7728
	Tansig	− 0.0135	0.2483	0.1586	0.3982	0.0155	0.3999	0.0385	0.9367	0.9330
	Tribas	− 0.0180	0.4850	0.6666	0.8164	0.0316	0.8200	0.0789	0.7340	0.6275

Table 16 Validation metrics of all ANN models for predicting accuracy (two-output structure)

Training function		BFGS Quasi-Newton (Trainbfg)								
Validation metrics		ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj-R^2*
Transfer function	Logsig	− 0.0264	0.4857	0.5342	0.7309	0.0846	0.7338	0.0706	0.9675	0.9672
	Purelin	− 0.0330	1.4310	3.6194	1.9025	0.2398	1.9111	0.1839	0.7799	0.7141
	Radbas	0.1040	0.7524	0.9998	0.9999	0.1253	0.9991	0.0961	0.9392	0.9345
	Softmax	0.0366	0.5874	0.6628	0.8141	0.1031	0.8171	0.0786	0.9597	0.9587
	Tansig	0.0291	0.4627	0.4118	0.6417	0.0834	0.6440	0.0620	0.9750	0.9743
	Tribas	− 0.1011	1.3406	3.5981	1.8969	0.2412	1.9030	0.1831	0.7812	0.7334
Training function		Bayesian Regularization Backpropagation (Ttrainbr)								
Transfer function	Logsig	0.0136	0.5431	0.6031	0.7766	0.0982	0.7801	0.0751	0.9633	0.9614
	Purelin	− 0.0752	1.4590	3.6286	1.9049	0.2455	1.9123	0.1840	0.7793	0.7064
	Radbas	− 0.0491	0.5885	0.6789	0.8240	0.1055	0.8263	0.0795	0.9587	0.9564
	Softmax	− 0.1383	0.6413	0.7877	0.8875	0.1057	0.8808	0.0848	0.9521	0.9496
	Tansig	− 0.0281	0.4924	0.4845	0.6960	0.0809	0.6987	0.0672	0.9705	0.9694
	Tribas	− 0.0245	0.8951	1.4733	1.2138	0.1688	1.2192	0.1173	0.9104	0.8894
Training function		Conjugate Gradient Backpropagation (Traincgb)								
Transfer function	Logsig	− 0.0385	0.5658	0.6924	0.8321	0.0951	0.8351	0.0804	0.9579	0.9566
	Purelin	− 0.0052	0.5442	0.6152	0.7843	0.0887	0.7880	0.0758	0.9626	0.9611

Table 16 (continued)

Training function		BFGS Quasi-Newton (Trainbfg)								
Validation metrics		ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj*-R^2
	Radbas	0.0346	0.6613	0.8680	0.9317	0.1131	0.9354	0.0900	0.9472	0.9465
	Softmax	− 0.0014	0.5421	0.6174	0.7857	0.0897	0.7894	0.0760	0.9625	0.9612
	Tansig	0.0267	0.5519	0.5439	0.7375	0.0890	0.7404	0.0712	0.9669	0.9663
	Tribas	− 0.0818	0.9341	1.4517	1.2049	0.1722	1.2077	0.1162	0.9117	0.9013
Training function		Levenberg-Marquardt (Trainlm)								
Transfer function	Logsig	0.0268	0.3123	0.2868	0.5355	0.0656	0.5373	0.0517	0.9826	0.9824
	Purelin	0.0300	1.4290	3.6037	1.8983	0.2436	1.9070	0.1835	0.7809	0.7253
	Radbas	− 0.0239	0.3437	0.3385	0.5818	0.0563	0.5840	0.0562	0.9794	0.9787
	Softmax	− 0.0002	0.4846	0.5762	0.7591	0.0774	0.7626	0.0734	0.9650	0.9632
	Tansig	0.0722	0.4067	0.4357	0.6601	0.0560	0.6592	0.0634	0.9735	0.9743
	Tribas	− 0.0747	0.6581	0.7512	0.8667	0.1088	0.8675	0.0835	0.9543	0.9524
Training function		Resilient Backpropagation (Trainrp)								
Transfer function	Logsig	0.0301	0.5847	0.6424	0.8015	0.0988	0.8047	0.0774	0.9609	0.9602
	Purelin	− 0.0191	1.4376	3.6057	1.8989	0.2429	1.9076	0.1836	0.7807	0.7199
	Radbas	− 0.0652	0.4766	0.4937	0.7026	0.0835	0.7028	0.0676	0.9700	0.9697
	Softmax	− 0.0542	0.5156	0.5816	0.7626	0.0835	0.7642	0.0735	0.9646	0.9645
	Tansig	0.0140	0.5599	0.7281	0.8533	0.0929	0.8571	0.0825	0.9557	0.9535
	Tribas	− 0.0024	0.6597	0.9227	0.9605	0.1125	0.9650	0.0929	0.9439	0.9406
Training function		Scaled Conjugate Gradient (Trainscg)								
Transfer function	Logsig	− 0.0011	0.3984	0.3838	0.6195	0.0659	0.6224	0.0599	0.9767	0.9763
	Purelin	− 0.0075	1.4455	3.6144	1.9012	0.2465	1.9100	0.1838	0.7802	0.7125
	Radbas	− 0.0331	0.4739	0.4221	0.6497	0.0790	0.6519	0.0627	0.9743	0.9746
	Softmax	− 0.0078	0.7305	1.1747	1.0838	0.1021	1.0889	0.1048	0.9286	0.9240
	Tansig	− 0.0387	0.5011	0.6315	0.7947	0.0813	0.7974	0.0767	0.9616	0.9605
	Tribas	− 0.0518	0.9018	1.4827	1.2177	0.1762	1.2222	0.1176	0.9098	0.8834

Table 17 Validation metrics of all ANN models for predicting formability (one-output structure)

Training function		BFGS Quasi-Newton (Trainbfg)								
Validation metrics		ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj*-R^2
Transfer function	Logsig	0.0061	0.4019	0.4625	0.6801	0.0262	0.6832	0.0657	0.8154	0.7719
	Purelin	0.0127	0.7181	1.0912	1.0446	0.0466	1.0494	0.1010	0.5645	0.2841
	Radbas	− 0.0177	0.4577	0.6258	0.7911	0.0295	0.7946	0.0765	0.7502	0.6594
	Softmax	0.0179	0.4938	0.7523	0.8673	0.0324	0.8712	0.0838	0.6998	0.5680
	Tansig	− 0.0065	0.3717	0.4397	0.6631	0.0237	0.6662	0.0641	0.8245	0.7842
	Tribas	0.0571	0.7680	1.3460	1.1602	0.0491	1.1642	0.1120	0.4628	0.0145
Training function		Bayesian Regularization Backpropagation (Ttrainbr)								
Transfer function	Logsig	− 0.0442	0.4200	0.5709	0.7556	0.0275	0.7578	0.0729	0.7721	0.6926
	Purelin	− 0.5426	0.9339	1.4040	1.1849	0.0576	1.0583	0.1018	0.4397	− 0.2457
	Radbas	− 0.0163	0.3948	0.5521	0.7430	0.0260	0.7463	0.0718	0.7796	0.7133
	Softmax	− 0.0009	0.4160	0.5938	0.7706	0.0275	0.7742	0.0745	0.7630	0.6865
	Tansig	− 0.0251	0.4152	0.5362	0.7323	0.0272	0.7352	0.0707	0.7860	0.7162
	Tribas	− 0.0679	0.4810	0.6836	0.8268	0.0313	0.8278	0.0797	0.7272	0.5467

Table 17 (continued)

Training function		BFGS Quasi-Newton (Trainbfg)								
Validation metrics		ME	MAE	MSE	RMSE	MRE	SD	SEM	R^2	*adj-*R^2
Training function		Conjugate Gradient Backpropagation (Traincgb)								
Transfer function	Logsig	0.0097	0.3663	0.3775	0.6144	0.0236	0.6172	0.0594	0.8493	0.8225
	Purelin	0.0158	0.4002	0.4883	0.6988	0.0263	0.7018	0.0675	0.8051	0.7623
	Radbas	0.0086	0.3714	0.4574	0.6763	0.0245	0.6794	0.0654	0.8175	0.7714
	Softmax	0.0070	0.3360	0.4046	0.6361	0.0221	0.6390	0.0615	0.8385	0.8031
	Tansig	− 0.0343	0.3764	0.4385	0.6622	0.0248	0.6644	0.0639	0.8250	0.7952
	Tribas	0.0463	0.5499	0.9323	0.9656	0.0361	0.9690	0.0932	0.6279	0.3695
Training function		Levenberg-Marquardt (Trainlm)								
Transfer function	Logsig	− 0.0098	0.2403	0.2964	0.5444	0.0158	0.5469	0.0526	0.8817	0.8643
	Purelin	− 0.0117	0.7175	1.0906	1.0443	0.0463	1.0491	0.1010	0.5647	0.2437
	Radbas	0.0226	0.3239	0.2794	0.5286	0.0207	0.5306	0.0511	0.8885	0.8755
	Softmax	0.1025	0.3555	0.5028	0.7091	0.0238	0.7049	0.0678	0.7993	0.7648
	Tansig	− 0.0222	0.3906	0.3783	0.6151	0.0251	0.6175	0.0594	0.8490	0.8327
	Tribas	0.0298	0.5271	0.5559	0.7456	0.0336	0.7485	0.0720	0.7781	0.7211
Training function		Resilient Backpropagation (Trainrp)								
Transfer function	Logsig	0.0109	0.3830	0.4395	0.6630	0.0250	0.6660	0.0641	0.8246	0.7984
	Purelin	− 0.0378	0.7241	1.0935	1.0457	0.0467	1.0499	0.1010	0.5636	0.1406
	Radbas	0.0222	0.3813	0.5156	0.7180	0.0252	0.7210	0.0694	0.7942	0.7479
	Softmax	0.0034	0.3283	0.4083	0.6390	0.0218	0.6419	0.0618	0.8371	0.8102
	Tansig	− 0.0067	0.3705	0.4349	0.6595	0.0239	0.6625	0.0637	0.8264	0.8000
	Tribas	− 0.0449	0.4787	0.6451	0.8032	0.0308	0.8057	0.0775	0.7425	0.6750
Training function		Scaled Conjugate Gradient (Trainscg)								
Transfer function	Logsig	− 0.0133	0.3308	0.4186	0.6470	0.0219	0.6499	0.0625	0.8329	0.8000
	Purelin	0.0202	0.7214	1.0894	1.0437	0.0468	1.0484	0.1009	0.5652	0.2057
	Radbas	0.0149	0.4168	0.6074	0.7794	0.0273	0.7829	0.0753	0.7576	0.7027
	Softmax	− 0.0217	0.4534	0.5749	0.7583	0.0299	0.7615	0.0733	0.7705	0.7066
	Tansig	0.0114	0.4041	0.5108	0.7147	0.0265	0.7179	0.0691	0.7961	0.7425
	Tribas	− 0.0062	0.5507	0.7301	0.8545	0.0355	0.8584	0.0826	0.7086	0.5809

Table 18 Weights and biases of the best ANN model for predicting accuracy (Trainscg with Radbas)—one-output structure

b_1	b_2	*IW*										*LW*
− 5.1413	3.9206	− 1.1773	− 0.7032	− 1.7944	− 1.8483	1.2188	2.0108	− 0.6296	1.9369	0.7717	0.3779	6.4552
1.2616		− 1.7346	− 1.1106	0.1240	0.7292	0.0289	− 0.3600	0.6209	− 2.3675	− 0.6774	− 1.6394	− 2.1276
0.6642		− 0.4661	1.3360	− 0.1381	− 1.3452	− 2.0399	− 0.3382	− 1.2443	3.8528	0.3032	1.1052	2.7984
2.7010		− 0.0229	0.2218	− 1.2916	2.9920	− 4.6045	3.0664	1.2266	− 2.3559	2.6646	4.1350	7.8625
1.5268		1.9534	− 2.5503	1.0572	2.3674	− 1.2938	2.2008	− 0.7986	− 2.0592	0.8307	0.2103	4.7908
− 2.5527		− 2.3043	− 1.3088	− 1.9797	2.4409	0.0598	0.7101	− 0.6936	0.8617	− 1.1759	0.2284	4.3514
0.9677		0.0759	1.0141	1.6768	− 0.7190	1.8965	− 1.7534	− 0.7307	2.1473	− 1.5098	0.2528	− 6.4366
0.9160		− 0.2273	− 0.7655	− 0.1006	− 0.3139	1.1390	2.3990	− 0.9744	− 1.6789	− 0.5358	− 2.2392	− 4.2600
2.3191		0.3222	− 0.2106	− 0.1155	1.1132	2.3911	− 0.5010	− 1.7080	0.0188	0.4250	− 0.6822	2.3087
− 4.4413		0.1749	− 0.6833	− 0.8397	1.7357	− 1.2086	− 0.6854	− 0.8242	1.5373	− 1.4357	0.2100	4.7381

Table 19 Weights and biases of the best ANN model for predicting formability (Trainbfg with Logsig)—one-output structure

b_1	b_2	*IW*										*LW*
− 6.2268	− 15.9818	− 3.4709	13.2876	− 7.2619	− 16.9449	− 13.0659	15.1323	− 2.0860	22.3754	14.3554	− 1.4070	1.7159
23.0398		− 12.3566	13.6008	3.3758	0.6860	5.9896	7.1386	6.7221	− 11.7328	4.2203	− 26.1003	1.0223
4.0820		− 8.0432	3.0246	0.5947	5.3230	− 7.8297	6.9763	− 1.2762	− 15.8632	− 1.6303	− 0.2870	− 1.0233
6.1367		1.7346	6.3500	8.5891	2.2943	− 18.7526	3.2920	15.8267	− 5.0288	− 1.2114	− 0.8662	3.9185
− 5.2423		− 0.8079	− 1.3571	3.6600	− 1.0886	− 2.3603	1.3466	− 10.1771	− 0.6398	− 0.0152	− 0.0121	− 17.6218
5.0917		− 3.0961	− 4.1506	− 6.1267	4.1379	4.7903	12.5706	− 8.8120	− 16.4002	3.2859	− 8.6912	16.1231
− 2.2796		8.2305	− 9.0868	− 0.7259	4.8521	− 4.1178	4.7707	− 6.4077	13.9174	21.7924	9.8806	3.1300
− 30.7090		− 1.0452	23.8309	− 28.0332	− 11.3384	− 3.0887	− 6.9537	− 9.0903	− 27.3553	14.7710	37.7944	25.4021
11.6168		17.0348	− 20.7669	11.0464	4.7830	− 3.2273	2.6294	16.1645	21.9736	19.5971	− 24.7539	25.2140
19.9338		9.7667	− 8.6318	− 2.0005	9.1440	18.1230	− 9.7823	6.2870	− 9.6843	− 6.1344	0.0711	1.1739

References

1. Emmens WC, Sebastiani G, van den Boogaard AH (2010) The technology of incremental sheet forming—a brief review of the history. J Mater Process Technol 210(8):981–997. https://doi.org/10.1016/j.jmatprotec.2010.02.014
2. Reddy NV, Lingam R (2018) Double sided incremental forming: capabilities and challenges. J Phys Conf Ser 1063:012170. https://doi.org/10.1088/1742-6596/1063/1/012170
3. Behera AK, de Sousa RA, Ingarao G, Oleksik V (2017) Single point incremental forming: An assessment of the progress and technology trends from 2005 to 2015. J Manuf Process 27:37–62. https://doi.org/10.1016/j.jmapro.2017.03.014
4. Shrivastava P, Tandon P (2019) Microstructure and texture based analysis of forming behavior and deformation mechanism of AA1050 sheet during Single Point Incremental Forming. J Mater Process Technol 266:292–310. https://doi.org/10.1016/j.jmatprotec.2018.11.012
5. Li Y, Chen X, Liu Z, Sun J, Li F, Li J, Zhao G (2017) A review on the recent development of incremental sheet-forming process. Int J Adv Manuf Technol 92(5–8):2439–2462. https://doi.org/10.1007/s00170-017-0251-z
6. Duflou JR, Habraken AM, Cao J, Malhotra R, Bambach M, Adams D, Vanhove H, Mohammadi A, Jeswiet J (2018) Single point incremental forming: state-of-the-art and prospects. Int J Mater Form 11(6):743–773. https://doi.org/10.1007/s12289-017-1387-y
7. Maqbool F, Bambach M (2019) Experimental and numerical investigation of the influence of process parameters in incremental sheet metal forming on residual stresses. J Manuf Mater Process 3(2):31. https://doi.org/10.3390/jmmp3020031
8. Cawley B, Adams D, Jeswiet J (2013) Examining tool shapes in single point incremental forming. Trans North Am Manuf Res Inst SME 41(April):114–120
9. Najm SM, Paniti I (2018) Experimental investigation on the single point incremental forming of AlMn1Mg1 foils using flat end tools. IOP Conf Ser Mater Sci Eng 448:012032. https://doi.org/10.1088/1757-899X/448/1/012032
10. Ziran X, Gao L, Hussain G, Cui Z (2010) The performance of flat end and hemispherical end tools in single-point incremental forming. Int J Adv Manuf Technol 46(9–12):1113–1118. https://doi.org/10.1007/s00170-009-2179-4
11. Wu SH, Reis A, Andrade Pires FM, Santos AD, Barata da Rocha A (2012) Study of Tool Trajectory in Incremental Forming. Adv Mater Res 472–475:1586–1591. https://doi.org/10.4028/www.scientific.net/AMR.472-475.1586
12. 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

13. Amirjan M, Khorsand H, Siadati MH, Eslami Farsani R (2013) Artificial neural network prediction of Cu–Al2O3 composite properties prepared by powder metallurgy method. J Mater Res Technol 2(4):351–355. https://doi.org/10.1016/j.jmrt.2013.08.001
14. Ezugwu EO, Fadare DA, Bonney J, Da Silva RB, Sales WF (2005) Modelling the correlation between cutting and process parameters in high-speed machining of Inconel 718 alloy using an artificial neural network. Int J Mach Tools Manuf 45(12–13):1375–1385. https://doi.org/10.1016/j.ijmachtools.2005.02.004
15. Li E (2013) Reduction of springback by intelligent sampling-based LSSVR metamodel-based optimization. Int J Mater Form 6(1): 103–114. https://doi.org/10.1007/s12289-011-1076-1
16. Marouani H, Aguir H (2012) Identification of material parameters of the Gurson–Tvergaard–Needleman damage law by combined experimental, numerical sheet metal blanking techniques and artificial neural networks approach. Int J Mater Form 5(2):147–155. https://doi.org/10.1007/s12289-011-1035-x
17. Lela B, Bajić D, Jozić S (2009) Regression analysis, support vector machines, and Bayesian neural network approaches to modeling surface roughness in face milling. Int J Adv Manuf Technol 42(11–12):1082–1088. https://doi.org/10.1007/s00170-008-1678-z
18. Hussaini SM, Singh SK, Gupta AK (2014) Experimental and numerical investigation of formability for austenitic stainless steel 316 at elevated temperatures. J Mater Res Technol 3(1):17–24. https://doi.org/10.1016/j.jmrt.2013.10.010
19. Kondayya D, Gopala Krishna A (2013) An integrated evolutionary approach for modelling and optimization of laser beam cutting process. Int J Adv Manuf Technol 65(1–4):259–274. https://doi.org/10.1007/s00170-012-4165-5
20. Ning J, Nguyen V, Huang Y, Hartwig KT, Liang SY (2018) Inverse determination of Johnson–Cook model constants of ultra-fine-grained titanium based on chip formation model and iterative gradient search. Int J Adv Manuf Technol 99(5–8):1131–1140. https://doi.org/10.1007/s00170-018-2508-6
21. Ning J, Liang SY (2019) Inverse identification of Johnson-Cook material constants based on modified chip formation model and iterative gradient search using temperature and force measurements. Int J Adv Manuf Technol 102(9–12):2865–2876. https://doi.org/10.1007/s00170-019-03286-0
22. Gok A (2015) A new approach to minimization of the surface roughness and cutting force via fuzzy TOPSIS, multi-objective grey design and RSA. Measurement 70:100–109. https://doi.org/10.1016/j.measurement.2015.03.037
23. Zuo K-T, Chen L-P, Zhang Y-Q, Yang J (2006) Manufacturing- and machining-based topology optimization. Int J Adv Manuf Technol 27(5–6):531–536. https://doi.org/10.1007/s00170-004-2210-8
24. Kurra S, Hifzur Rahman N, Regalla SP, Gupta AK (2015) Modeling and optimization of surface roughness in single point incremental forming process. J Mater Res Technol 4(3):304–313. https://doi.org/10.1016/j.jmrt.2015.01.003
25. Nasrollahi V, Arezoo B (2012) Prediction of springback in sheet metal components with holes on the bending area, using experiments, finite element and neural networks. Mater Des 36:331–336. https://doi.org/10.1016/j.matdes.2011.11.039
26. Mekras N (2017) Using artificial neural networks to model aluminium based sheet forming processes and tools details. J Phys Conf Ser 896:012090. https://doi.org/10.1088/1742-6596/896/1/012090
27. Kashid S, Kumar S (2013) Applications of artificial neural network to sheet metal work—a review. Am J Intell Syst 2(7):168–176. https://doi.org/10.5923/j.ajis.20120207.03
28. Maji K, Kumar G (2020) Inverse analysis and multi-objective optimization of single-point incremental forming of AA5083 aluminum alloy sheet. Soft Comput 24(6):4505–4521. https://doi.org/10.1007/s00500-019-04211-z
29. Oraon M, Sharma V (2018) Prediction of surface roughness in single point incremental forming of AA3003-O alloy using artificial neural network. Int J Mater Eng Innov 9(1):1. https://doi.org/10.1504/IJMATEI.2018.092181
30. Mulay A, Ben BS, Ismail S, Kocanda A (2019) Prediction of average surface roughness and formability in single point incremental forming using artificial neural network. Arch Civ Mech Eng 19(4): 1135–1149. https://doi.org/10.1016/j.acme.2019.06.004
31. Oraon M, Sharma V, and Mandal S (2020) “Performance measurement in incremental deformation of Brass Cu67Zn33 through soft computing tool”. 83–89
32. Radu C, Cristea I, Herghelegiu E, Tabacu S (2013) Improving the accuracy of parts manufactured by single point incremental forming. Appl Mech Mater 332:443–448. https://doi.org/10.4028/www.scientific.net/AMM.332.443
33. Behera AK, Afonso D, Murphy A, Jin Y, and de Sousa RA (2018) “Accuracy analysis of incrementally formed tunnel shaped parts”. 40–49
34. McAnulty T, Jeswiet J, Doolan M (2017) Formability in single point incremental forming: a comparative analysis of the state of the art. CIRP J Manuf Sci Technol 16:43–54. https://doi.org/10.1016/j.cirpj.2016.07.003
35. Bayram H, Köksal NS (2017) Investigation of the geometrical accuracy and thickness distribution using 3D laser scanning of AA2024-T3 sheets formed by SPIF. Mater Tehnol 51(1):111–116. https://doi.org/10.17222/mit.2015.296
36. Nama SA, Namer NSM, Najm SM (2014) The effect of using grease on the surface roughness of aluminum 1100 sheet during the single point incremental forming process. J Trends Mach Des 1(1):53–56 [Online]. Available: www.stmjournals.com
37. Rattanachan K, Chungchoo C (2009) Formability in single point incremental forming of dome geometry. Asian Int J Sci Technol Prod Manuf Eng 2(4):57–63
38. Nimbalkar DH, Nandedkar VM (2013) Review of Incremental Forming of Sheet Metal Components. Int J Eng Res Appl 3(5): 39–51
39. Gatea S, Lu B, Chen J, Ou H, McCartney G (2019) Investigation of the effect of forming parameters in incremental sheet forming using a micromechanics based damage model. Int J Mater Form 12(4): 553–574. https://doi.org/10.1007/s12289-018-1434-3
40. Kumar A, Gulati V, Kumar P, Singh V, Kumar B, Singh H (2019) Parametric effects on formability of AA2024-O aluminum alloy sheets in single point incremental forming. J Mater Res Technol 8(1):1461–1469. https://doi.org/10.1016/j.jmrt.2018.11.001
41. Shamsari M, Mirnia MJ, Elyasi M, Baseri H (2018) Formability improvement in single point incremental forming of truncated cone using a two-stage hybrid deformation strategy. Int J Adv Manuf Technol 94(5–8):2357–2368. https://doi.org/10.1007/s00170-017-1031-5
42. Lu B, Fang Y, Xu DK, Chen J, Ai S, Long H, Ou H, Cao J (2015) Investigation of material deformation mechanism in double side incremental sheet forming. Int J Mach Tools Manuf 93:37–48. https://doi.org/10.1016/j.ijmachtools.2015.03.007
43. Li Y, Liu Z, W. J. T. (Bill. Daniel, and P. A. Meehan (2014) Simulation and experimental observations of effect of different contact interfaces on the incremental sheet forming process. Mater Manuf Process 29(2):121–128. https://doi.org/10.1080/10426914.2013.822977
44. Lora F, Schaeffer L (2014) Incremental forming process strategy variation analysis through applied strains. Braz J Sci Technol 1(1): 5. https://doi.org/10.1186/2196-288X-1-5
45. Golabi S, Khazaali H (2014) Determining frustum depth of 304 stainless steel plates with various diameters and thicknesses by incremental forming. J Mech Sci Technol 28(8):3273–3278. https://doi.org/10.1007/s12206-014-0738-6

46. Fiorentino A, Ceretti E, Attanasio A, Mazzoni L, Giardini C (2009) Analysis of forces, accuracy and formability in positive die sheet incremental forming. Int J Mater Form 2(S1):805–808. https://doi.org/10.1007/s12289-009-0467-z
47. Lu HB, Le Li Y, Liu ZB, Liu S, Meehan PA (2014) Study on step depth for part accuracy improvement in incremental sheet forming process. Adv Mater Res 939:274–280. https://doi.org/10.4028/www.scientific.net/AMR.939.274
48. Mulay A, Ben BS, Ismail S, Kocanda A, Jasiński C (2018) Performance evaluation of high-speed incremental sheet forming technology for AA5754 H22 aluminum and DC04 steel sheets. Arch Civ Mech Eng 18(4):1275–1287. https://doi.org/10.1016/j.acme.2018.03.004
49. Hagan MT, Demuth HB, Beale MH, and De Jesús O (2014) Neural Network Design. Martin Hagan
50. Nabipour M, Keshavarz P (2017) Modeling surface tension of pure refrigerants using feed-forward back-propagation neural networks. Int J Refrig 75:217–227. https://doi.org/10.1016/j.ijrefrig.2016.12.011
51. Beale MH, Hagan MT, and Demuth HB (2013) "Neural Network Toolbox TM User ' s Guide R2013b," Mathworks Inc
52. Riedmiller PM "Machine Learning : Multi Layer Perceptrons," Albert-Ludwigs-University Freibg. AG Maschinelles Lernen, [Online]. Available: http://ml.informatik.uni-freiburg.de/_media/documents/teaching/ss09/ml/mlps.pdf
53. Principe J, Euliano NR, Lefebvre WC (1997) Neural and adaptive systems: fundamentals through simulation: multilayer perceptrons. Neural Adapt Syst Fundam Through Simulation©:1–108. https://doi.org/10.1002/ejoc.201200111
54. Beale MH, Hagan M, and Demuth H (2019) "Deep learning toolbox getting started guide," Deep Learn. Toolbox. https://doi.org/10.1016/j.neunet.2005.10.002
55. A. C. M. & S (1997) Guido, Introduction to with Python Learning Machine
56. Shahin M, Maier HR, and Jaksa MB (2000) "Evolutionary data division methods for developing artificial neural network models in geotechnical engineering Evolutionary data division methods for developing artificial neural network models in geotechnical engineering by M A Shahin M B Jaksa Departmen," no. November
57. Zhang G, Eddy Patuwo B, Hu MY (1998) Forecasting with artificial neural networks:: The state of the art. Int J Forecast 14(1):35–62 [Online]. Available: https://econpapers.repec.org/RePEc:eee:intfor:v:14:y:1998:i:1:p:35-62
58. Bishop CM (1995) Neural Networks for Pattern Recognition. Oxford University Press, Inc., Oxford
59. Beale MH, Hagan MT, and Demuth HB (2020) "Deep Learning Toolbox TM User ' s Guide How to Contact MathWorks"
60. Misra S and He J (2020) "Stacked neural network architecture to model the multifrequency conductivity/permittivity responses of subsurface shale formations," in Machine Learning for Subsurface Characterization, Elsevier, pp. 103–127
61. Garson GD (1991) Interpreting Neural-Network Connection Weights. AI Expert 6(4):46–51
62. Zarei MJ, Ansari HR, Keshavarz P, Zerafat MM (2020) Prediction of pool boiling heat transfer coefficient for various nano-refrigerants utilizing artificial neural networks. J Therm Anal Calorim 139(6):3757–3768. https://doi.org/10.1007/s10973-019-08746-z
63. Ding H, Luo W, Yu Y, Chen B (2019) Construction of a robust cofactor self-sufficient bienzyme biocatalytic system for dye decolorization and its mathematical modeling. Int J Mol Sci 20(23):6104. https://doi.org/10.3390/ijms20236104
64. Zhou B et al (2015) Relative importance analysis of a refined multi-parameter phosphorus index employed in a strongly agriculturally influenced watershed. Water Air Soil Pollut 226(3):25. https://doi.org/10.1007/s11270-014-2218-0
65. Shabanzadeh P, Yusof R, Shameli K (2015) Artificial neural network for modeling the size of silver nanoparticles' prepared in montmorillonite/starch bionanocomposites. J Ind Eng Chem 24:42–50. https://doi.org/10.1016/j.jiec.2014.09.007
66. Vatankhah E, Semnani D, Prabhakaran MP, Tadayon M, Razavi S, Ramakrishna S (2014) Artificial neural network for modeling the elastic modulus of electrospun polycaprolactone/gelatin scaffolds. Acta Biomater 10(2):709–721. https://doi.org/10.1016/j.actbio.2013.09.015
67. Rezakazemi M, Razavi S, Mohammadi T, Nazari AG (2011) Simulation and determination of optimum conditions of pervaporative dehydration of isopropanol process using synthesized PVA–APTEOS/TEOS nanocomposite membranes by means of expert systems. J Membr Sci 379(1–2):224–232. https://doi.org/10.1016/j.memsci.2011.05.070
68. Goh ATC (1995) Back-propagation neural networks for modeling complex systems. Artif Intell Eng 9(3):143–151. https://doi.org/10.1016/0954-1810(94)00011-S
69. Hagan E, Jeswiet J (2004) Analysis of surface roughness for parts formed by computer numerical controlled incremental forming. Proc Inst Mech Eng B J Eng Manuf 218(10):1307–1312. https://doi.org/10.1243/0954405042323559

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