Journal of Intelligent Manufacturing (2023) 34:331–367
https://doi.org/10.1007/s10845-022-02026-8

# Investigation and machine learning-based prediction of parametric effects of single point incremental forming on pillow effect and wall profile of AlMn1Mg1 aluminum alloy sheets

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

Received: 23 March 2022 / Accepted: 14 September 2022 / Published online: 18 October 2022

## Abstract

Today the topic of incremental sheet forming (ISF) is one of the most active areas of sheet metal forming research. ISF can be an essential alternative to conventional sheet forming for prototypes or non-mass products. Single point incremental forming (SPIF) is one of the most innovative and widely used fields in ISF with the potential to form sheet products. The formed components by SPIF lack geometric accuracy, which is one of the obstacles that prevents SPIF from being adopted as a sheet forming process in the industry. Pillow effect and wall displacement are influential contributors to manufacturing defects. Thus, optimal process parameters should be selected to produce a SPIF component with sufficient quality and without defects. In this context, this study presents an insight into the effects of the different materials and shapes of forming tools, tool head diameters, tool corner radiuses, and tool surface roughness (Ra and Rz). The studied factors include the pillow effect and wall diameter of SPIF components of AlMn1Mg1 aluminum alloy blank sheets. In order to produce a well-established study of process parameters, in the scope of this paper different modeling tools were used to predict the outcomes of the process. For that purpose, actual data collected from 108 experimentally formed parts under different process conditions of SPIF were used. Neuron by Neuron (NBN), Gradient Boosting Regression (GBR), CatBoost, and two different structures of Multilayer Perceptron were used and analyzed for studying the effect of parameters on the factors under scrutiny. Different validation metrics were adopted to determine the quality of each model and to predict the impact of the pillow effect and wall diameter. For the calculation of the pillow effect and wall diameter, two equations were developed based on the research parameters. As opposed to the experimental approach, analytical equations help researchers to estimate results values relatively speedily and in a feasible way. Different partitioning weight methods have been used to determine the relative importance (RI) and individual feature importance of SPIF parameters for the expected pillow effect and wall diameter. A close relationship has been identified to exist between the actual and predicted results. For the first time in the field of incremental forming study, through the construction of Catboost models, SHapley Additive exPlanations (SHAP) was used to ascertain the impact of individual parameters on pillow effect and wall diameter predictions. CatBoost was able to predict the wall diameter with $R^2$ values between the range of 0.9714 and 0.8947 in the case of the training and testing dataset, and between the range of 0.6062 and 0.6406 when predicting pillow effect. It was discovered that, depending on different validation metrics, the Levenberg–Marquardt training algorithm performed the most effectively in predicting the wall diameter and pillow effect with $R^2$ values in the range of 0.9645 and 0.9082 for wall diameter and in the range of 0.7506 and 0.7129 in the case of the pillow effect. NBN has no results worthy of mentioning, and GBR yields good prediction only of the wall diameter.

**Keywords** SPIF · Incremental sheet forming · Pillow effect · CatBoost · SHapley additive exPlanations · SHAP

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

Extended author information available on the last page of the article

## Introduction

Sheet metal forming technology forms parts through a series of localized incremental plastic deformations called Incremental Sheet Forming (ISF) (Jackson & Allwood, 2009; Kumar & Kumar, 2015). ISF is a moderate, innovative sheet-forming technology without employing classic punch and dies (Najm et al., 2030). The components are formed by way of a predefined strategy called path, which guides a simple tool that performs incremental movements over the clamped sheet to form the desired final shape (Azaouzi & Lebaal, 2012; M. Amala Justus Selvam, R. Velu, & T. Dheerankumar, 1050; Szpunar et al., 2021). The first dieless incremental forming process can be traced back to Leszak (Leszak, 1967). The main two types of ISF are Single Point Incremental Forming (SPIF) and Two Point Incremental Forming (TPIF). The first patent mentioned before can be considered a SPIF, and TPIF was first presented by Matsubara (Matsubara, 1994). SPIF is more appropriate than conventional sheet-forming methods to manufacture prototypes, small production batches, and customized components (Mezher et al., 2018; Paniti et al., 2020; Trzepieciński et al., 2022a). SPIF is flexible and with its help it is very simple to produce complex geometries by following a programmed path strategy using a CNC machine with at least three controlled axes (Trzepieciński et al., 2022b). A very up-to-date paper briefly overviewed recent developments in SPIF of lightweight materials (Trzepieciński et al., 2021a). SPIF is one of the promising technologies in the future of sheet forming for aerospace applications (Trzepieciński et al., 2021b). However, the quality, surface finish, and geometric accuracy of products formed by SPIF are affected by different process parameters. At the same time, improper selection of process parameters can cause deviation and inaccuracy with respect to desired shapes. Still, ISF in general, and SPIF in particular, have disadvantages in terms of geometric accuracy and pillow effect. Due to the elastic–plastic deformation throughout the forming process, which is apt to instability, the formed sheet is prone to be impacted by a springback (Khan et al., 2015; Zhang et al., 2016) and pillow effect (Bai et al., 2019; Najm & Paniti, 2020). Matching the wall profile of the product to the CAD model based on the path strategy is one of the challenges of this process. One of the main factors that affects the accuracy of the components in terms of geometric accuracy is springback (Mezher et al., 2021a, 2021b). Wall displacement due to springback is the difference between the actual wall angle or diameter and the angle or diameter of the CAD model. The springback of component walls is induced mainly by two factors: local springback and global springback. Due to the sheet's position following the use of the advanced forming tool, the local forming that occurred while the sheet returns to its initial position after the tool has passed causes local springback. On the other hand, global springback occurs as a result of residual stresses, which stem from the unconstrained material in the formed sheet after releasing the sheet from clamping (Edwards et al., 2017). In addition to the two types of springback mentioned above, Gates et al. (2016) noted that another type of springback ensues upon the displacement of the forming tool, which they named continuous local springback. Kiridena et al. (2016) invented innovative tools to increase the geometric accuracy of products formed incrementally. They described that dimensional accuracy is significantly impacted by the lengthening of the tool shank and by the decrease of the tool diameter, while the underside flat part of the donut-shaped tool increases the formed part's accuracy. The papers (Najm & Paniti, 2018, 2020) stated that, as compared to the hemispherical tool of components formed by SPIF, flat tools led to better formability, increased homogeneity in thickness distribution, and a minimized pillow effect.

The pillow effect—also called bulge—is a concave surface developing on the unformed bottom area in the center of the parts (Nasulea & Oancea, 2021); pillow effect is the influential forming error that negatively impacts the geometric accuracy of SPIF components and limits formed part's formability (Wei et al., 2020). The pillow effect occurs due to the springback of the wall between the obtained geometry and the CAD model, in which the forming tool generates tension on the unformed surface from the corner, and the middle part remains free to become puffy. The pillow appearing in the form of a bulge on the remaining unformed surface must stay flat during and after the forming process. Researchers have attempted to find a proper method to prevent the formation of pillow effect in products formed using SPIF. Ambrogio et al. (2007) suggest an ANOVA analysis based equation which allows for identifying quadratic model equations for the purpose of predicting geometrical errors between an ideal surface and real surfaces in the case of truncated pyramid shape aluminum alloy AA1050–O sheets. The researchers state that pillow effect is strongly influenced by tool diameter and product height. Micari et al. (2007) indicated that two types of errors in the typologies result in inaccurately formed parts of ISF: the two typologies' errors are springback and pillow effect. Pillow effect increases forming force, which results in the inaccuracy of parts. Thus, work hardening during multi-point forming decreases the pillow effect in the case of multi-point forming (Zhang et al., 2017). Al-Ghamdi and Hussain (2015) have studied the mechanical properties of formed parts on pillow effect. They state that the tensile fracture affecting and controlling formability has an insignificant impact on bulge. However, a decrease in hardening exponent, which controls stretchability, decreases bulge, and this can be considered a significant property that affects the pillow effect. Furthermore, higher forming depth leads to bigger billowing but not in a linear way: certain specified depths relieve the pillow effect because of the property of the hardening exponent. Afzal (2021) sees this differently: he claims that the pillow effect sets in because of two different states in the formed sheet: the unformed base is in an elastic state, while the formed wall is in a plastic state. Isidore et al. (Isidore, 2014; Isidore et al., 2016) found that parts formed with the help of a hemispherical tool caused more oversized pillows because strains and compressive stresses were generated due to the compression of the material. Also, it was observed that smaller pillows resulted when a flat tool was used because tensile stress and strains impacted in transverse directions. Essa and Hartley (2011) discovered various ways to improve geometric accuracy by executing FE in SPIF. They reduced the bending of components via flanking using a support plate, and thus minimized springback by stationing a supporting tool and by eliminating the pillow effect through modifying the last stage of the tool path.

Due to their excellent formability and resistance to corrosion, general-purpose 3xxx's alloys are used for architectural applications and for the manufacturing of various products. 3xxx's alloys are non-heat-treatable but exhibit about 20% more strength than 1xxx series alloys. Manganese is the principal alloying element of 3xxx alloys, which is added either of its own or with magnesium. However, magnesium is considerably more effective than manganese as a hardener: about 0.8% Mg is equal to 1.25% Mn (Davis, 2001). In the experiments of single point incremental forming conducted in the scope of this study, an AlMn1Mg1 aluminum alloy blank sheet with an initial thickness of 0.22 mm was used. This alloy belongs to the 3xxx series based on its sequence of elements. Examples of common AlMn1Mg1 aluminum alloy applications include beer & beverage cans (Hirsch, 2006), where good formability is achieved by (Mg) with strengthening effects by (Mn). AlMn1Mg1 aluminum alloy is also used in automotive radiator heat exchangers and as tubing in commercial power plant heat exchangers (Kaufman, 2000), as well as for the following specific purposes: sheet metal work, storage tanks, agricultural applications, building products, containers, electronics, furniture, kitchen equipment, recreation vehicles, trucks and trailers (Davis, 2001).

Recently, various techniques of artificial intelligence have been used in many industries including the metal forming industry. In the last decade, machine learning has been applied using various Artificial Neural networks (ANN) techniques in a number of applications and industries. Furthermore, machine learning has dominated manufacturing with a view to designing the most practical, sufficient and adequate predictive models (Hussaini et al., 2014; Kondayya & Gopala Krishna, 2013; Lela et al., 2009; Li, 2013; Marouani & Aguir, 2012). A recent state-of-the-art review discovered analytical and numerical models of incremental formation (IF). IF-related issues have been solved using artificial intelligence AI-based computational approaches. This research evaluates IF literature. Artificial neural networks, support vector regression, decision trees, fuzzy logic, evolutionary algorithms, and particle swarm optimization solve IF-related issues. Hybrid approaches integrate some of the previous strategies (Nagargoje et al., 2021). Different intelligences with or without controlled manufacturing have been generated or developed as predictive models in end-milling machining, high-speed machining, and powder metallurgy (Amirjan et al., 2013; Ezugwu et al., 2005; Zain et al., 2010). Artificial neural network architecture has generated tool paths directly from a digital component model for ISF components. Multiple training techniques, network topologies, and training sets were examined in a feedforward network structure with backpropagation. They prove neural networks can generate tool routes for sheet metal free forming (Hartmann et al., 2019). Notably with respect to SPIF, different studies have developed ANN, support Victor Regression (SVR), Gradient Boosting Regressions (GBR) models for the prediction of formability (Najm & Paniti, 2021a), surface roughness (Najm & Paniti, 2021b; Trzepieciński et al., 2021c) and hardness (Najm et al., 2021) of components formed using SPIF under various forming conditions. Low et al. (2022) predict this error distribution from input CAD geometry of SPIF components using Convolutional Neural Networks-Forming Prediction (CNN-FP). For the untrained wall angle, the CNN-FP model had an RMSE (Root mean squared error) of 0.381 mm at 50 mm depth. The CNN-FP performance for the untrained complicated geometry was determined to be 0.391 mm at 30 mm depth. However, there was considerable degradation at 50 mm depth of the complicated geometry when the model's prediction had an RMSE of 0.903 mm.

In light of the literature, the above-detailed concerns in connection with SPIF as well as the lacking standardization of SPIF process parameters and the scarcity of referent mathematical models have motivated the authors of this paper to deal with the investigation and prediction of the pillow effect and wall diameter of truncated frustums processed by SPIF. To the authors' knowledge, such an experimental process has not been tested or described in the literature so far. The researchers consider SPIF process parameters on geometric accuracy (pillow effect and wall diameter) as one of the main significant drawbacks of the SPIF process. Geometric accuracy in the form of components' wall and pillow effect can be influenced by various factors including tool materials, tool shape and size and the surface finish of the tooltip. Therefore, in the present paper, geometric accuracy with respect to components' wall accuracy and pillow effect have been investigated experimentally and have been studied in the scope of the above-mentioned parameters. Furthermore, as an aim and novelty in the scope of this paper, various models have been built along different combinations of parameters for both pillow effect and wall diameter datasets, and accordingly prediction equations based on weights and biases have been derived. The combined partitioning weight of the NN was adopted to estimate the relative importance (RI) of SPIF parameters on model output. In addition, and for the first time in ISF, process parameters have been interpreted via SHapley Additive exPlanations (SHAP), which were utilized to establish parameters' relevance on the pillow effect and wall diameter.

## Material properties

In this section material properties of the investigated sheet and properties of the applied tools are presented. In the scope of this study, an initial 0.22 mm thick AlMn1Mg1 aluminum alloy square-shaped blank sheets of 150 mm × 150 mm were used, the tensile of the specimens cut from the blank sheet was tested according to the EN ISO 6892-1:2010 standard and the tensile was conducted using an INSTRON 5582 universal testing machine at room temperature. Based on the rolling direction, the specimens were cut from three directions: 0°, 45°, and 90°, and three samples were produced for each direction. The relative standard deviation did not exceed 3% based on the test procedure. Furthermore, the planar anisotropy values (r10) were established using an Advanced Video Extensometer (AVE). The average values of the mechanical properties are listed in Table 1. The chemical composition of the sheet material was analyzed with the help of a WAS FOUNDRY-MASTER Optical Emission Spectroscopy (OES), and the pertaining data are found in Table 2.

For the formation of the sheet in the experiments of SPIF in the scope of this research design, various forming tools were used as far as tool shapes, tool materials, the tip radius (R), and the corner radius (r) of the tool are concerned. Two different tool designs have been selected: hemispherical with variety in tip radius (R) (see Fig. 1a) and flat tools with changes in corner radius (r) (see Fig. 1b). The (R) values are 1 mm, 2 mm, and 3 mm, while the (r) values are 0.1 mm, 0.3 mm, and 0.5 mm. Each set of tools was created using six different materials: Table 3 details the metal tools and their properties, and Table 4 lists the tools created using polymer provided by STRATASYS. Hardness was experimentally tested with the help of a Wolpert Diatronic 2RC S hardness tester according to ISO 6506-1:2014. Consequently, the corresponding hardness value for each tool was adopted for matching with the ISO code of the tested materials in question. Thus, the mechanical properties were determined based on the ISO code of the forming tool's material. For the determination of the ISO code, a FOUNDRY-MASTER Pro2 Optical Emission Spectrometer was used to measure each forming tool material.

**Table 1** Mechanical properties of AlMn1Mg1 aluminum alloy sheet

| Direction | 0° | 45° | 90° | SD (σ) |
|---|---|---|---|---|
| Yield strength (MPa) 0.2 | 88.30 | 90 | 86.30 | 1.5122 |
| Ultimate tensile strength (MPa) | 183 | 155.5 | 170.3 | 11.2377 |
| Elongation (%) | 16.44 | 9.27 | 12.48 | 2.9325 |
| Elongation A50 (%) | 16.88 | 10.45 | 12.95 | 2.6466 |
| n5 | 0.297 | 0.266 | 0.268 | 0.0142 |
| r10 | 0.554 | 0.580 | 0.594 | 0.0166 |

**Table 2** Chemical composition of AlMn1Mg1 aluminum alloy sheet (in wt%)

| 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 metal tools

| 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 |

## Experiments

A frustum geometry shape with the dimensions shown in Fig. 2 was created experimentally during the forming process. The failure criterion was defined as the end of the forming process, and an example of a failed part is shown in Fig. 3. Two different tool shapes (spherical and flat), with three different tooltip sizes (three different radii (R) for the hemispherical and three different corner radii (r)) for the flat tool, and six different materials (see Fig. 4)—(Steel (C45), Brass (CuZn39Pb3), Bronze (CuSn8), Copper (E-Cu57), Aluminum (AlMgSi 0.5) and polymer VeroWhitePlus (RGD835))—altogether produce 36 various forming process conditions. For reasons of reliability and correct measurement assurance, three parts were formed for each process condition for the purpose of ensuring the accuracy of the obtained results; the total number of formed components thus totaled 108. The Design of Experiments (DOE) was not applied for minimizing the number of experiments because the resulting data were collected and used as an actual dataset (input and output) for the predictive models. A SIEMENS

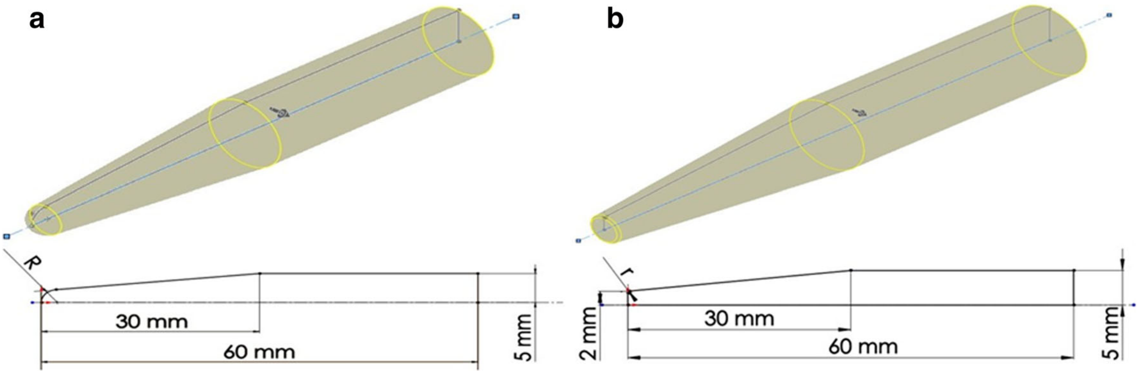


**Fig. 1** Forming tools: **a** hemispherical tool, **b** flat tool

**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 | 2500 | 58 | 25 | 85 |

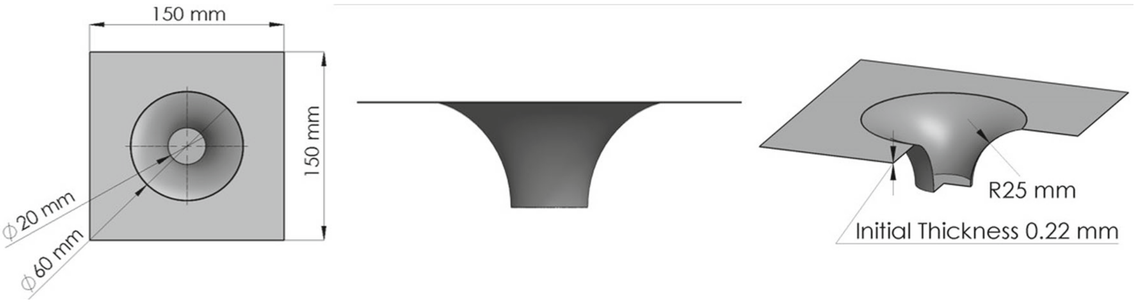


**Fig. 2** CAD geometry and dimensions of experimental product

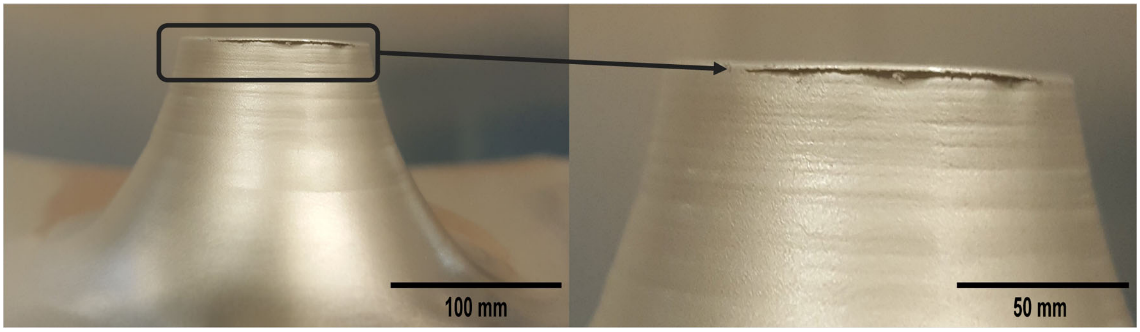
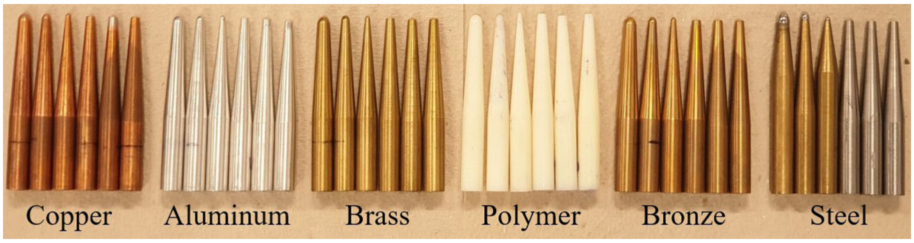


**Fig. 3** Failed specimen

**Fig. 4** Tool materials



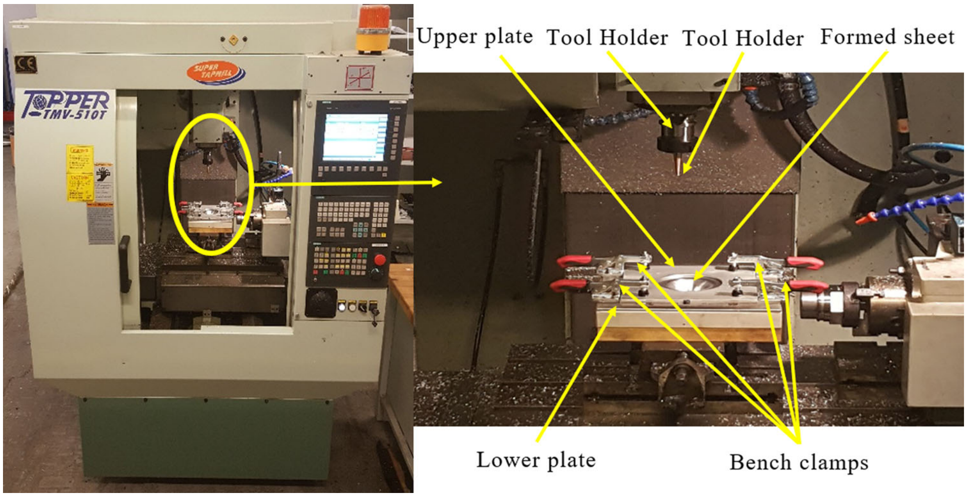


**Fig. 5** Topper CNC machine with rapid clamping rig on the CNC milling table

**Table 5** Fixed process parameters

| Path strategy | Feed rate (mm/min) | Spindle speed (rpm) | Step down (mm) | Lubricant |
|---|---|---|---|---|
| Helical | 1500 | 2000 | 0.05 | Machine oil |

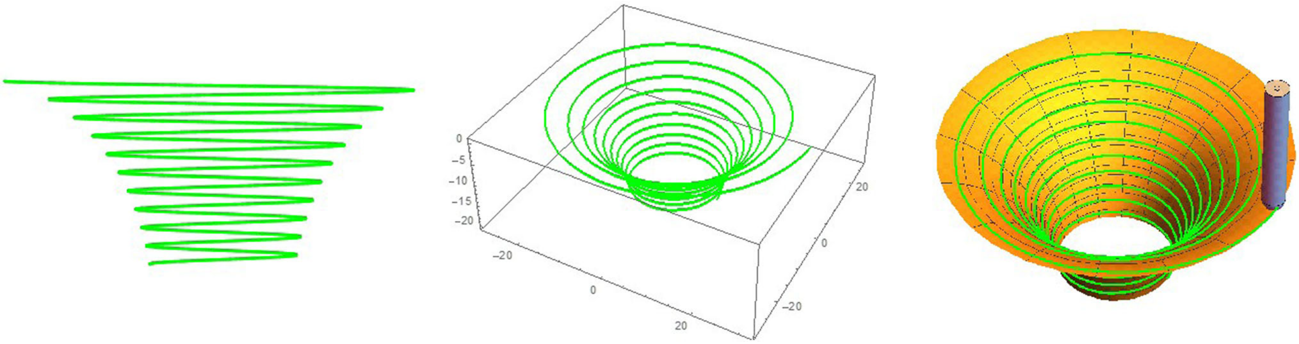


**Fig. 6** CAD geometry of experimental product with view of an inward spiral path

Topper TMV-510T 4-axis CNC milling machine combined with sinumerik 840-D controller was used in the forming process (see Fig. 5 for the machine and clamping design). The fixed forming process parameters are shown in Table 5. A step size with a value of 0.05 mm in direction Z was used as a constant value for the whole forming path. The application of a smaller step size results in an accurate geometry and better surface finish of final parts formed by SPIF (Lu et al., 2014; Mulay et al., 2018). A strategy of spiral path moves inward to the sheet center, with the center adopted to form the frustum parts incrementally, as shown in Fig. 6. This strategy was developed by Skjoedt et al. (2007) to reach the maximum axial loads in step size and to achieve a better surface of the impacted inner surface with the help of the forming tool. Also, this strategy aids the more successful and precise formation of components (Kumar & Gulati, 2020).

In each part of forming, the surface roughness of the forming tool was measured before and after starting the forming process. The forming tool's surface roughness measured before the forming process was adopted as an input value of the upcoming formed part. For this value and the upcoming forming process, the surface roughness of the tool was adopted as input and so on. The above-mentioned scenario was applied to all forming tools used in the scope of this study. Nevertheless, a new polymer tool was employed in each forming process because of the wear on the forming tool developed by the polymer. The surface roughness of each polymer tool was measured before the start of the process.

The profiles of all formed parts and their unformed bases (Wall diameter and Pillow effect) were measured using a Mitutoyo Coordinate Machine CV-1000 (MCM), see Fig. 7. The average deviation of the three parts experimentally formed under the same SPIF condition was compared to the CAD model in terms of wall diameter and unformed base, and this was adopted as values of the geometrical accuracy and pillow effect studied in this research.

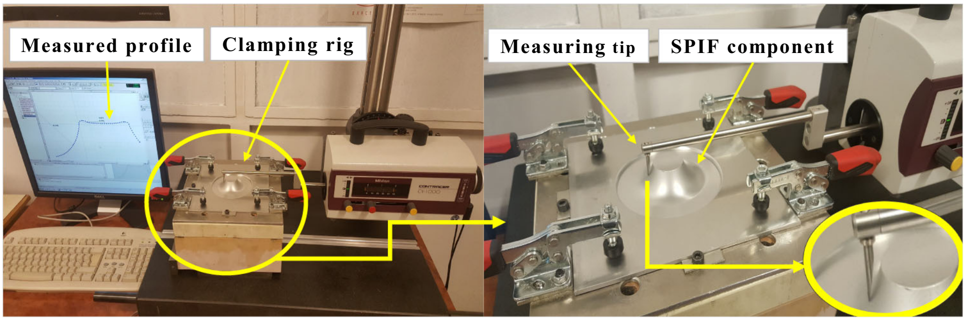


**Fig. 7** Mitutoyo coordinate machine

The MCM instrument was operated in the measurement processes along 49 mm at a speed of 0.5 mm per second together with a pitch size of 0.005 mm. MCM automatically creates a component profile with the possibility of measuring each point, as shown in Fig. 8. This figure shows examples only for 20 points. For each part, more than 9000 points were measured automatically with the help of axis X and Z coordinates. The measured data points were used to build an actual profile so that the CAD model, the measurement of the wall diameter and the pillow effect could be compared with this resulting profile. The pillow effect is the difference between the highest peak and the lowest valley values measured. In an ideal case of forming, the lower bottom unformed surface should be flat between the peak and the lowest valley value.

## Artificial neural networks

The notion of the neural network is often traced back to Warren McCulloch and Walter Pitts's 1940 study. Their basic idea was that neural networks could compute any logical functions or mathematical formulations. Furthermore, the invention of the Perceptron Network at the end of 1950 can be considered as the first practical application of ANN (Hagan et al., 2014). Recently, neural networks have become the main interest for thousands of researchers in different fields of science. As a whole, it can be established that there is no scientific field that does not have any links with neural networks. Scientists in different areas including healthcare, aerospace, defense, arts, filmmaking, music and various industry technologies adopted ANN.

### Neuron by Neuron (NBN)

Using Visual Studio 6.0 and C + + language, Hao Yu and Wilamowski (Yu & Wilamowski, 2009a, 2009b) developed a Neuron by Neuron (NBN) trainer. NBN can work with Fully Connected Neurons (FCN) and needs a lower number of neurons than Multilayer Perceptron (MLP). Figure 9 shows a scheme of five inputs fully connected with three neurons as well as one output and its topology. The developed tool supports three different types of neurons. The neurons are bipolar ("mbip"), unipolar (mu), and linear ("mlin"). Both the mbip and mu have outputs not exceeding 1, with negative and positive values for mbip, and only positive values for mu. The equations of the three types of neurons are presented in Eqs. 1, 2, and 3 (Yu & Wilamowski, 2009a, 2009b), respectively. In this study, running time was 100, and iteration was 500 with a maximum error of 0.001. The training tool provides a direct sum squared error (SSE) with the plotting area on the interface, which can indicate the accuracy of the prediction. All other parameters were set as default because they gave the best results. All the neuron types were tried with different numbers of connection neurons. It is worth mentioning that categorical encoding was used because the collected data from the 108 samples contained numerical and categorical datasets (see Categorical Encoding).

$$\text{Bipolar (mbip) } f_b(net) = tanh(gain \times net) + der \times net \tag{1}$$

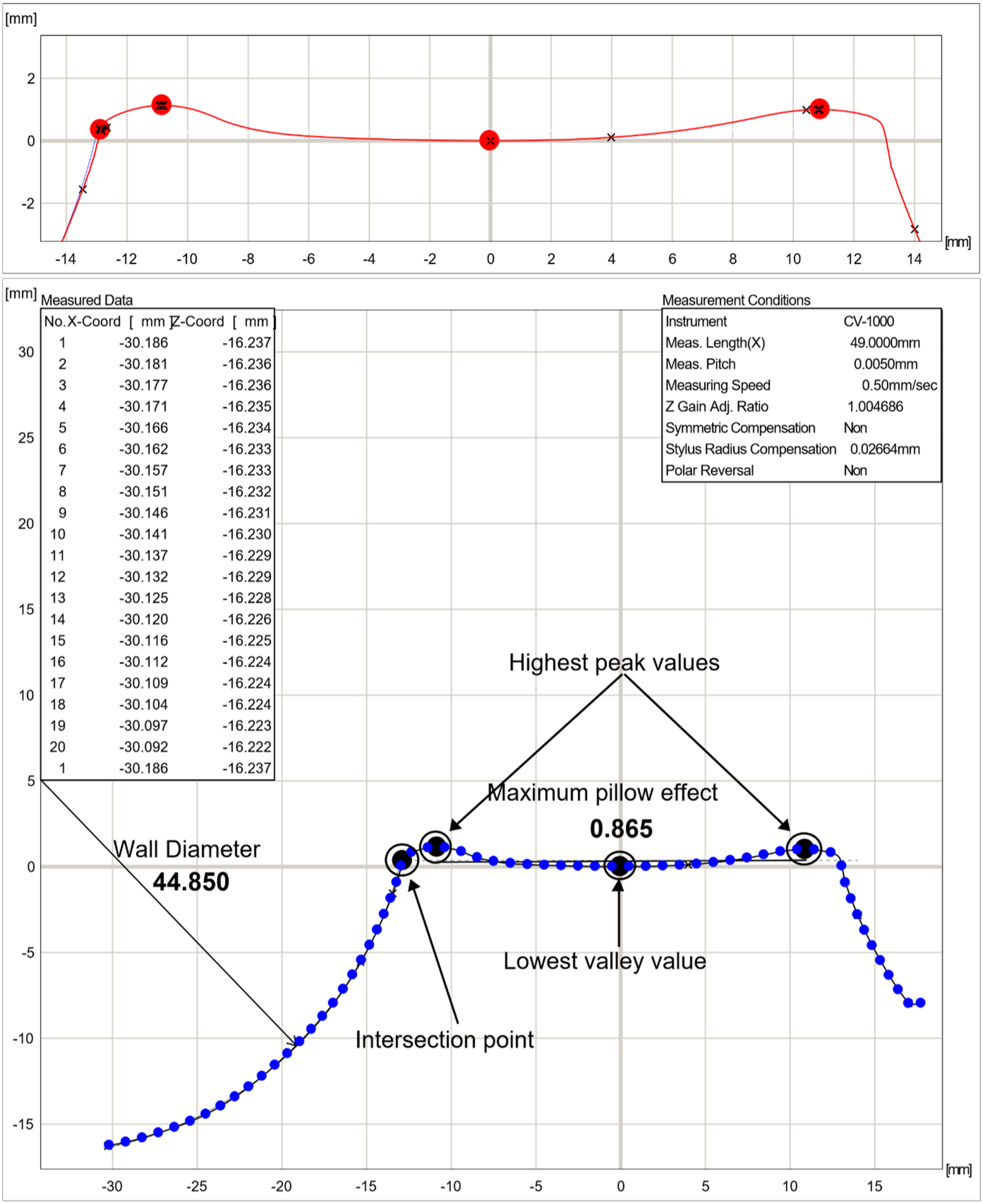
$$\text{Unipolar (mu) } f_u(net) = \frac{1}{1 + e^{-gain \times net}} + der \times net \tag{2}$$

$$\text{Linear (mlin) } f_l(net) = gain \times net \tag{3}$$

where "gain" and "der" are parameters of activation functions.

### Gradient boosting regression (GBR)

Gradient Boosting is one of the most common and powerful tree algorithms (Pedregosa, 2011). Gradient boosting is an ML method used as a classifier and regressor. This is a prediction model of collective learning, where each level attempts to correct the errors of a previous level in a formation called decision trees. Gradient Boosting model (Chen et al., 2021) with least-squares loss and 1000 regression trees with a depth of 6 and a minimum samples split of 2 was run to predict the pillow effect and wall diameter as shown in Fig. 10. Least-squares loss tries to locate predictive points and fits them on



Measured Data

| No. | X-Coord [ mm ] | Z-Coord [ mm ] |
|---|---|---|
| 1 | -30.186 | -16.237 |
| 2 | -30.181 | -16.236 |
| 3 | -30.177 | -16.236 |
| 4 | -30.171 | -16.235 |
| 5 | -30.166 | -16.234 |
| 6 | -30.162 | -16.233 |
| 7 | -30.157 | -16.233 |
| 8 | -30.151 | -16.232 |
| 9 | -30.146 | -16.231 |
| 10 | -30.141 | -16.230 |
| 11 | -30.137 | -16.229 |
| 12 | -30.132 | -16.229 |
| 13 | -30.125 | -16.228 |
| 14 | -30.120 | -16.226 |
| 15 | -30.116 | -16.225 |
| 16 | -30.112 | -16.224 |
| 17 | -30.109 | -16.224 |
| 18 | -30.104 | -16.224 |
| 19 | -30.097 | -16.223 |
| 20 | -30.092 | -16.222 |
| 1 | -30.186 | -16.237 |

Measurement Conditions

| | |
|---|---|
| Instrument | CV-1000 |
| Meas. Length(X) | 49.0000mm |
| Meas. Pitch | 0.0050mm |
| Measuring Speed | 0.50mm/sec |
| Z Gain Adj. Ratio | 1.004686 |
| Symmetric Compensation | Non |
| Stylus Radius Compensation | 0.02664mm |
| Polar Reversal | Non |

**Fig. 8** Component profile; pillow effect and wall diameter as measured by a Mitutoyo coordinate machine

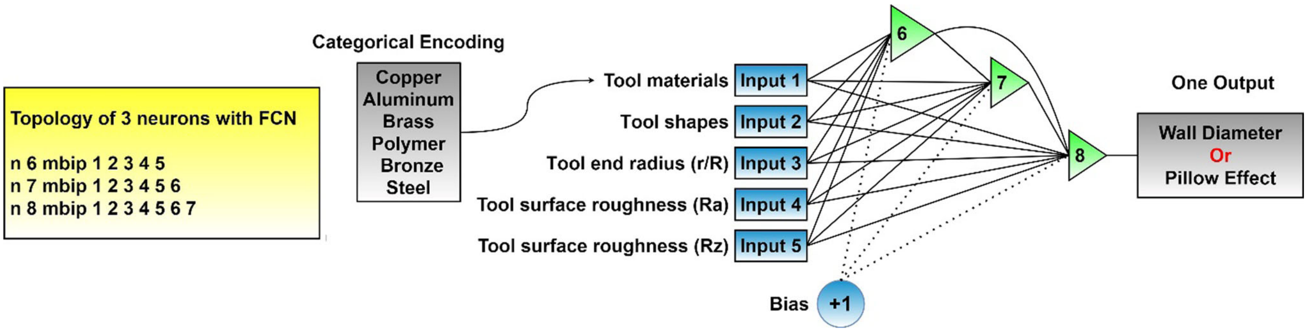


**Fig. 9** Scheme of neuron by neuron (NBN) of five inputs fully connected with three neurons and one output

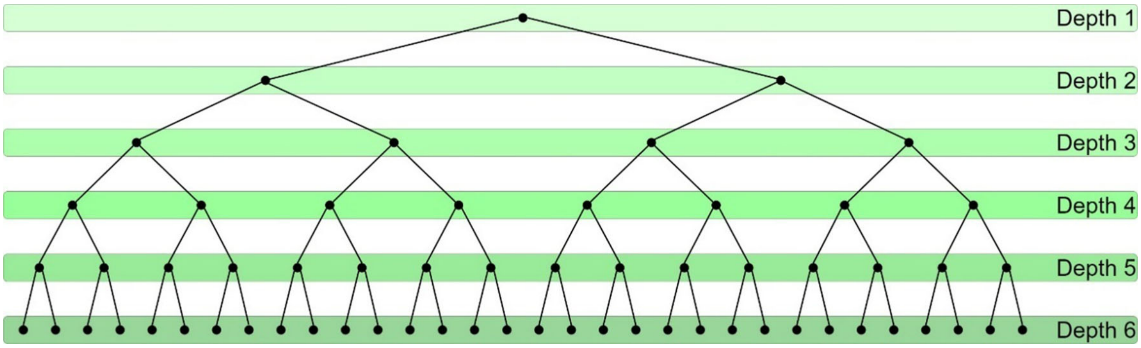


**Fig. 10** Regression trees of depth 6 and minimum samples split 2

the already fitted line in an attempt to minimize error. The regression tree depth shows how many splits a tree allows to be created before a prediction is made. The boosting method is about consecutively learning learners' nodes on the basis of previous ones by way of fitting the data set and analyzing the ensuing errors. In other words, this means that the boosting method works as a cycle consisting of learning nodes, fitting results to the data set, analyzing errors between the actual and predicted values, and re-starting learning the nodes that have been learned before. Moreover, this cycle is repeated until the previously set iteration number is reached.

## CatBoost

CatBoost is a high-performance open-source library for gradient boosting of decision trees (Prokhorenkova et al., 2018). A machine learning algorithm called CatBoost was developed by Yandex researchers and engineers. CatBoost has many features. One of the main features allows using categorical data (non-numeric factors), for which pre-processing of data is not needed. In addition, turning such data into numbers by encoding is not necessary, either (Dorogush et al., 2018). Furthermore, CatBoost aptly predicts with default parameters, so parameter tuning is unnecessary (Ibragimov & Gusev, 2019). By default, CatBoost builds 1000 trees, i.e., full symmetric binary trees with six in-depth and two leaves. The learning rate is determined automatically according to the properties of the trained dataset and the number of iterations. The automatically selected learning rate should be close to the optimal one. The number of iterations can be lowered for faster training, but in this case learning rate must be increased.

## Multilayer Perceptron MLP

Network elements' organization or structure, interconnections, inputs, and outputs constitute net topology. The topology of an ANN can be defined according to the number of input and output layers, with the transfer functions between these layers and the number of neurons in each layer (Nabipour & Keshavarz, 2017). ANN structure consists of input and output layers with a minimum of one hidden layer. Each layer of the net contains a number of neurons. Neurons in the input are equal to the number of input variables, and output neurons are equal to the number of outputs associated with each input. Based on the transfer function or so-called activation function (Beale et al., 2013), these neurons in the layers allow for the transfer of weight between the layers backward and forward. The current study adopted the multilayer perceptron (MLP) structure for its ANN model using a backpropagation learning algorithm. The idea of the MLP was initiated by Werbos in 1974, and Rumelhart, McClel-

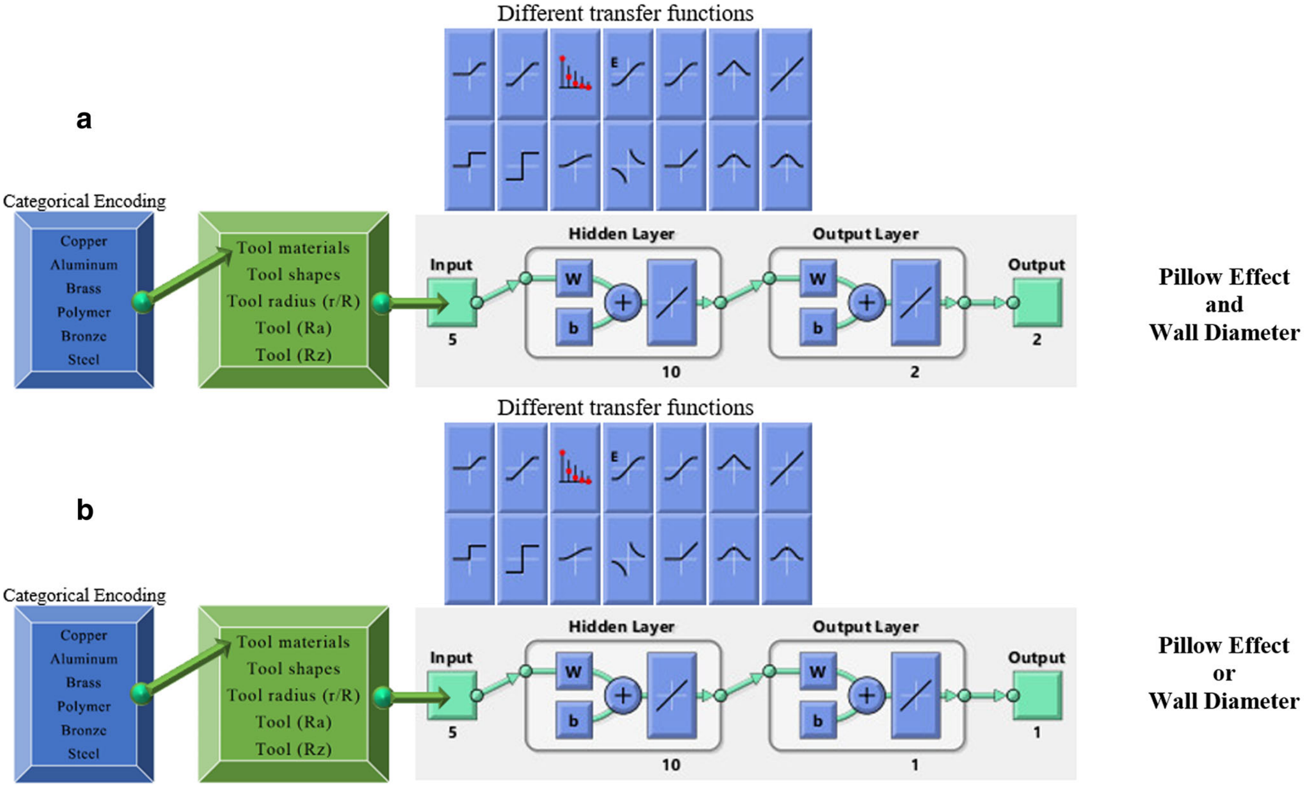


**Fig. 11** Different multilayer perceptron (MLP) structures: **a** two outputs; **b** one output

land and Hinton in 1986 (Riedmiller). Equation 1 defines the multilayer perceptron as follows:

$$y = f(net) = f\left(\sum_{i=1}^{n} w_i x + b\right) \tag{4}$$

where $y$ is the output and $x$ is input, $w_i$ are the weights, and $b$ is the bias (Principe et al., 1997).

Using MATLAB R2020a (Beale et al., 2019), two different MLP structures were created to predict wall diameter and pillow effect. The inputs and targets dataset were obtained from actual measured data from the formed parts by SPIF. The main difference between the two trained structures is the number of prediction outputs: one of the networks deals with one target, and the second deals with two targets. The outputs are wall diameter or pillow effect, or wall diameter together with the pillow effect. However, the inputs have 5 neurons: tool materials, tool shapes, tool radius, and tool surface roughness values Ra and Rz. In the scope of this research, each net structure exhibited one hidden layer with ten neurons connected to the input and output layers, as shown in the illustrated scheme in Fig. 11a and b. The other main parameters selected for the purpose of training in this study were as follows: learning rate 0.01, performance goal 0.001, and 1000 as the number of epochs. It is noteworthy that different training and transfer functions were tried and trained for finding the best model and structure (see Training function and Transfer function).

The training flowchart of the developed model and the checking process using the test data are presented in Fig. 12. Two main conditions make decisions during the running process. The first loop in light blue color saves the model and all variables with low limits of condition. The second loop in light red is activated after the first condition occurs and stops. The second loop finds the training and variables, compares these with the variables saved from the previous training, and continues to do so until 1000 iterations. Shared step loops are displayed as light green arrows.

### Training function

In neural networks, dataset training utilizes the optimization technique for tuning and finding a set of network weights for building a good prediction map. There are various optimization algorithms also called training functions. The training function is an algorithm for training the network to identify a specific input and for mapping such input to an output. The training function depends on many characteristics, including the trained dataset, weights and biases, as well as the performance goal. One challenge of building good, fast and really accurate predictions lies in selecting a fitted training function for the network. With this in mind, in the scope of this paper,

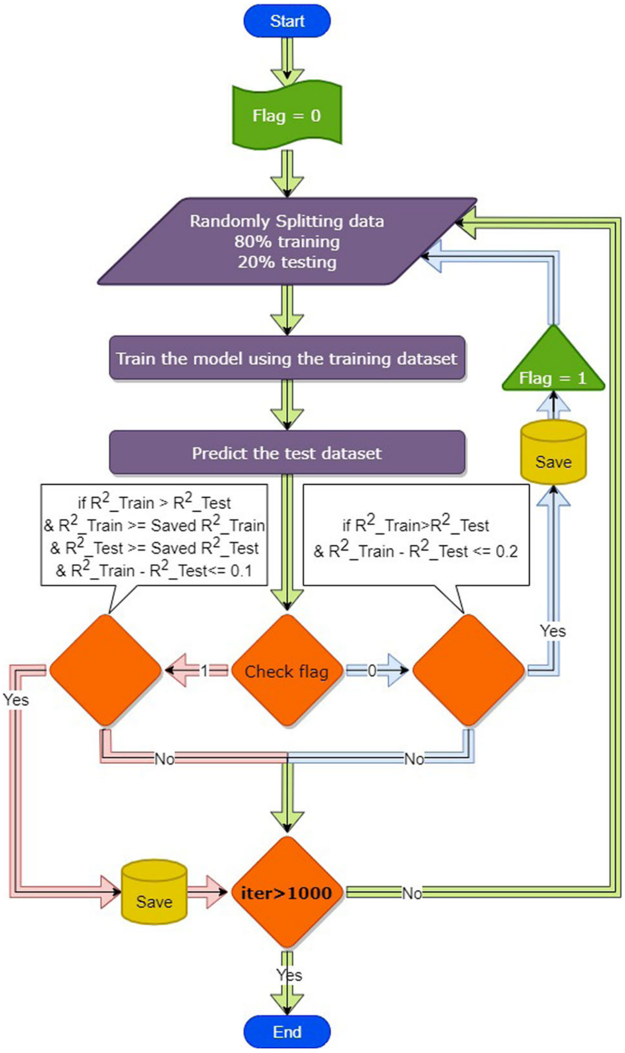


**Fig. 12** Flowchart of developed multilayer perceptron (MLP) model

ten different types of training function “learning algorithms” were executed in both MLP nets for the purpose of mapping outputs associated with inputs. Levenberg–Marquardt (Trainlm) is one of these implemented training functions and is considered the fastest compared to the others. Similarly, the BFGS Quasi-Newton algorithm is quite fast (Beale et al., 2019). The corresponding training functions are listed in Table 6.

**Table 6** Details of training functions used in multilayer perceptron (MLP)

| Acronym | Algorithm | Description |
|---|---|---|
| LM | Trainlm | Levenberg–Marquardt |
| BFG | Trainbfg | BFGS Quasi-Newton |
| RP | Trainrp | Resilient backpropagation |
| SCG | Trainscg | Scaled conjugate gradient |
| CGB | Traincgb | Conjugate gradient with Powell/Beale restarts |
| CGF | Traincgf | Fletcher–Powell conjugate gradient |
| CGP | Traincgp | Polak–Ribiére conjugate gradient |
| OSS | Trainoss | One step secant |
| GDX | Traingdx | Variable learning rate backpropagation |
| BRB | Trainbr | Bayesian regularization backpropagation |

## Transfer function

In machine learning, the sums of each node are weighted, and the sum is passed through a function known as activation function or transfer function. The summed weights undergo a transfer function, and the transfer function computes the output of each layer through adopting summed weights entering the given layer. Setting proper transfer functions is a challenging task and is based on many factors but mainly on network structure. Usually, in multilayer perceptron (MLP) Log-sigmoid (Logsig) is used, and there are alternative functions like Hyperbolic tangent sigmoid (Tansig), which is generally used for pattern recognition (Beale et al., 2019). In this study, besides the aforementioned two functions, thirteen different transfer functions were performed to improve prediction accuracy. Eventually, the linear (Purelin) transfer function was chosen for the output layer in all cases. Table 7 lists all the transfer function algorithms with related Eqs. 5–18 used for this study (Demuth, 2000).

## Dataset distribution

The actual data of SPIF components were used as input data for all the structures and models built and trained in the scope of this study to predict pillow effect and wall diameter as network outputs. Predicting results when forming new parts without executing any new process of forming is both economical and more practical. However, existing data must be separated into various subsets: i.e., training, validation, and testing datasets. In fact, prediction accuracy and training performance are significantly influenced by dividing the dataset into training and testing subsets (Zhang et al., 1998). Inappropriate subsets negatively affect benchmark performance. On the other hand, Shahin (Shahin et al., 2000) claimed there is no apparent association between the splitting ratio of the dataset, but Zhang et al. (Zhang et al., 1998) explain that the splitting ratio is one of the main problems. Yet, no general setting is available as a solution. Based on their surveys, most researchers split the datasets into lines with a different ratio of subsets. The most broadly adopted ratios are 90% vs. 10%, 80% vs. 20%, or 70% vs. 30% for training and testing. As part of the training run in the scope of this paper, optimal prediction was obtained by adopting a dividing ratio of 80% vs. 20% of the actual data (108 samples) concerning training and testing datasets, respectively. Regarding the

**Table 7** Details of transfer functions used in multilayer perceptron (MLP)

| | | | |
|---|---|---|---|
| E | Elliot sigmoid (Elliotsig) | $f(x) = elliotsig(x) = (x)/1+\|x\|$ | (5) |
| | Positive hard limit (Hardlim) | $f(x) = hardlim(x) = 1, \text{ if } x >= 0;$<br>$= 0, \text{ otherwise.}$ | (6) |
| | Symmetric hard limit (Hardlims) | $(x) = hardlim(x) = 1, \text{ if } x >= 0;$<br>$= -1, \text{ otherwise.}$ | (7) |
| | Logarithmic sigmoid (Logsig) | $f(x) = logsig(x) = 1 / (1 + exp(-x))$ | (8) |
| | Inverse (Netinv) | $f(x) = netinv\ (x) = 1/x$ | (9) |
| | Positive linear (Poslin) | $f(x) = poslin(x) = x, \text{ if } x >= 0;$<br>$= 0, \text{ if } x <= 0.$ | (10) |
| | Linear (Purelin) | $f(x) = purelin(x) = x$ | (11) |
| | Radial basis (Radbas) | $f(x) = radbas(x) = exp\ (-x\ \hat{}2)$ | (12) |
| | Radial basis normalized (Radbasn) | $f(x) = radbasn(x) = exp\ (-x\ \hat{}2) / sum(exp(-x\hat{}2))$ | (13) |
| | Positive saturating linear (Satlin) | $f(x) = satlin(x) \quad = 0, \text{ if } x <= 0;$<br>$= x, \text{ if } 0 <= x <= 1;$<br>$= 1, \text{ if } 1 <= x.$ | (14) |
| | Symmetric saturating linear (Satlins) | $f(x) = satlins(x) = -1, \text{ if } x <= -1;$<br>$= \ x, \text{ if } -1 <= x <= 1;$<br>$= 1, \text{ if } 1 <= x.$ | (15) |
| | Soft max (Softmax) | $f(x) = softmax(x) = exp(x)/sum(exp(x))$ | (16) |
| | Symmetric sigmoid (Tansig) | $f(x) = tansig(x) = 2/ (1+exp\ (-2*x))-1$ | (17) |
| | Triangular basis (Tribas) | $f(x) = tribas(x) = 1 - abs(x), \text{ if } -1 <= x <= 1;$<br>$= 0, \text{ otherwise.}$ | (18) |

where $x$ is the weighted sum of $w\_i$, $b$, and $y$ of Eq. 4

actual dataset, 108 rows were taken from the SPIF components formed experimentally, and the recorded rows were used as training and testing datasets.

## Categorical encoding

There are three typical methods for altering categorical variables to numerical values. One of them represents categorical data sets: this is the One-Hot Encoding variables (Guido, 1997). The other two are Ordinal Encoding and Dummy Encoding. The categorical variables will sparse-binarize and can be integrated into training in different machine learning models. The idea of one-hot-encoding is to replace categorical data with one or more new features. Each parameter takes a new numerical feature, and one of these features is always active through their substitution by 0 and 1 by way of creating

**Table 8** Details of different encoding methods

| Tool Material | One-Hot Encoding | | | | | | | Dummy Encoding | | | | | | Ordinal Encoding |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Copper | 1 | 0 | 0 | 0 | 0 | 0 | | 1 | 0 | 0 | 0 | 0 | | 0 |
| Aluminum | 0 | 1 | 0 | 0 | 0 | 0 | | 0 | 1 | 0 | 0 | 0 | | 1 |
| Brass | 0 | 0 | 1 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | 0 | | 2 |
| Polymer | 0 | 0 | 0 | 1 | 0 | 0 | | 0 | 0 | 0 | 1 | 0 | | 3 |
| Bronze | 0 | 0 | 0 | 0 | 1 | 0 | | 0 | 0 | 0 | 0 | 1 | | 4 |
| Steel | 0 | 0 | 0 | 0 | 0 | 1 | | 0 | 0 | 0 | 0 | 0 | | 5 |

one binary for each category. In ordinal encoding, each category is given an Integer value and sequences up to the number of the actual features. The dummy encoding is a slightly improved version of one-hot-encoding. In one-hot-encoding the numerical values are equal to the number of categories and the dummy is equal to the number of categories minus 1. Table 8 shows different ways of encoding the categorical data as part of the current study. In the scope of this research, two categorical sets were encoded. Tool materials and tool shapes were binarized as sparse matrices. However, ordinal encoding was adopted for tool materials because the other methods were in conflict either with feature importance calculation (see Contribution analysis of input variables) or one-hot-encoding concerning the tool shape; in the scope of latter the flat tool is replaced by 0 and the hemispherical tool is represented by 1.

## Overfitting

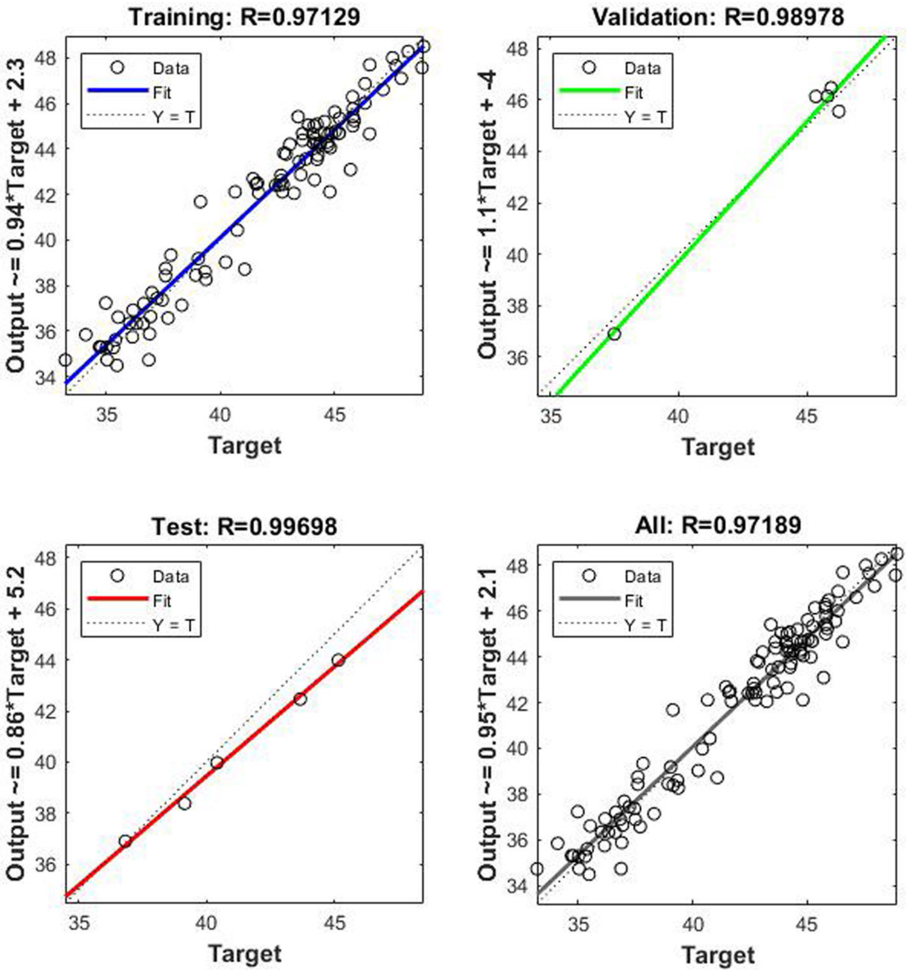
A neural network is a practical tool used in various applications, but it has several drawbacks. One of these drawbacks is underfitting or overfitting. Underfitting happens when a model is too simple for training the selected dataset, and overfitting is when the network gives a larger error for the new data set than concerning the set trained before. In other words, the trained net can memorize the learned data set but is not trained to generalize those new data sets that are not fitted. Boosting algorithms are generally supplied with regularization methods to avert overfitting (Ibragimov & Gusev, 2019) because the number of samples in an ensemble set does not always improve the accuracy but can reduce generalization ability (Mease & Wyner, 2008). There are various ways to improve and handle network generalization, such as: using a large network to provide a good fit, training several nets to guarantee that good generalization is found, averaging the outputs of trained multiple neural networks, separating data randomly, and tuning the complexity of a net through regularization (Bishop, 1995).

In the MLP network created for the purpose of prediction in this study, generalization has been improved by the so-called early stopping method. The early stopping technique is a default method automatically provided for all supervised network creation functions, including backpropagation networks. This method splits the training dataset into three subsets: training, validation, and testing (see Fig. 13). During network training, training subset data will be utilized for calculating the gradient and for updating weights and biases to fit the models. Also, the validation subset estimates prediction error during the training process; and finally, the test subset will be utilized to test the learned network as well as to assess generalization errors and plot them instantly as the training is running. However, if the overfitting of data starts over during the training of the net, the errors will be larger in the validation subset. In addition, if the validation error increases above the minimum at a significantly different iteration number of iterations and, at the same time, becomes larger than the error of the test subset, the training stops, and network weights and biases return to the smallest validation error (Beale et al., 2020).

## Investigation of accuracy

There are numerous validation metrics, but using the appropriate validation metric is essential for evaluating a predictive model, and physical observations are also necessary for improving model performance. This study compared and validated different structures and various training and transferring algorithms for assessing and measuring the agreement between actual and predicted values. Picking the suitable validation metric is crucial and challenging for assessing results and for minimizing errors. In the scope of assessing results to test performance, all structures and models trained and tested in this study were compared based on their evaluation by appropriate metrics. Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determination ($R^2$), together with the primary metrics used to extract these above-mentioned metrics, were adopted. An $R^2$ value close to 1 indicates good performance, and RMSE and also MAE near 0 means lower error. In this scenario reliable performance can be ensured; and vice versa: a large deviance between RMSE and MAE values points to significant variations in the distribution of error. However, the limitations of $R^2$ are listed in Misra and He (2020); furthermore, RMSE and MAE have an accurate evaluation compared to other validation metrics because if MAE has more stability, RMSE is

**Fig. 13** Regression process; training, validation, and testing



more sensitive to error. For our purposes and for prediction values validation, Standard Error Mean (SEM) was used, which is the original Standard Deviation (SD) of the sample size divided by the square root of the sample size. The other metrics like Error (E), Mean Error (ME), and Mean Square Error (MSE) were involved in originating validation equations. The Total Sum of Square ($SS_{tot}$) and the Sum of the Square of Residuals ($SS_{res}$) were adopted for the derivation of $R^2$ and adj. $R^2$. Analytically, pertaining validation equations are as follows:

$$E = \left(y_i^{target} - y_i^{predict}\right) \tag{19}$$

$$ME = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{target} - y_i^{predict}\right) \text{ or } ME = \frac{1}{n}\sum_{i=1}^{n}(E) \tag{20}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left(\left|y_i^{target} - y_i^{predict}\right|\right) \text{ or } MAE = \frac{1}{n}\sum_{i=1}^{n}(|E|) \tag{20}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{target} - y_i^{predict}\right)^2 \text{ or } MSE = \frac{1}{n}\sum_{i=1}^{n}(E)^2 \tag{22}$$

$$RMSE = \sqrt{MSE} \tag{23}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left(\left|\frac{y_i^{target} - y_i^{predict}}{y_i^{target}}\right|\right) \text{ or}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left(\left|\frac{E}{y_i^{target}}\right|\right) \tag{24}$$

$$\overline{E} = \left(y_i^{predict} - y_i^{target}\right) \tag{25}$$

$$\overline{ME} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{predict} - y_i^{target}\right) \tag{26}$$

$$SD = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\overline{E} - \overline{ME}\right)^2} \tag{27}$$

$$SEM = \frac{SD}{\sqrt{n}} \tag{28}$$

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{target}\right) \tag{29}$$

$$SS_{tot} = \sum_{i=1}^{n}\left(y_i^{target} - \overline{y}\right)^2 \tag{30}$$

$$SS_{res} = \sum_{i=1}^{n}\left(y_i^{target} - y_i^{predict}\right)^2 \text{ or } SS_{res} = \sum_{i=1}^{n}(E)^2 \tag{31}$$

$$R^2 = \frac{SS_{tot} - SS_{res}}{SS_{tot}} \tag{32}$$

Thus

$$R^2 = \frac{\sum_{i=1}^{n}\left(y_i^{target} - \overline{y}\right)^2 - \sum_{i=1}^{n}\left(y_i^{target} - y_i^{predict}\right)^2}{\sum_{i=1}^{n}\left(y_i^{actual} - \overline{y}\right)^2} \tag{33}$$

$$\overline{\overline{y}} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i^{predict}\right) \tag{34}$$

$$adj.R^2 = 1 - \left(\frac{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{target} - y_i^{predict}\right)^2}{\left(\frac{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{predict} - \overline{\overline{y}}\right)^2}{n-1}\right)}\right) \text{ or}$$

$$adj.R^2 = 1 - \left(\frac{MSE}{\left(\frac{\frac{1}{n}\sum_{i=1}^{n}\left(y_i^{predict} - \overline{\overline{y}}\right)^2}{n-1}\right)}\right) \tag{35}$$

## Contribution analysis of input variables

The contribution analysis of input variables on the associated outputs is called feature importance, variable importance, or relative importance. However, this analysis indicates each feature's relative importance in the model driving a prediction by showing changes in the averages of predictions in the event that the feature value changes. The substitute of the input variables with high relative importance (RI) values significantly affects results as compared to variables that have lower RI values (Nabipour & Keshavarz, 2017; Rezakazemi et al., 2011; Vatankhah et al., 2014). Concerning this, there are various techniques to calculate feature importance and such techniques include Garson (Garson, 1991), Most Squares (Ibrahim, 2013), and Connection Weights (Olden & Jackson, 2002). These methods are established based on the connection weights of neurons and are depicted in Eqs. (36)–(38), respectively. In addition, there is also built-in feature importance in Gradient Boost and CatBoost regression. In the literature, different studies adopted feature importance to calculate and evaluate their impacts on variables (Ding, 2019; Rezakazemi et al., 2011; Shabanzadeh et al., 2015; Vatankhah et al., 2014; Zarei et al., 2020; Zhou et al., 2015), but the first time feature importance was adopted to find relative importance in ISF was in Najm & Paniti, 2021b and later in Najm & Paniti, 2021a; Najm et al., 2021). The equations below (36–38) describe the above as follows:

$$RI(\%) = \frac{\left[\sum_{j=1}^{n_h}\left(y_{vj}/\sum_{k=1}^{n_v} y_{kj}\right)hO_j\right]}{\sum_{y=1}^{n_v}\left[\sum_{j=1}^{n_h}\left(y_{vj}/\sum_{k=1}^{n_v} y_{kj}\right)hO_j\right]} \tag{36}$$

$$RI(\%) = \frac{\sum_{j=1}^{n_v}\left(y_{vj}^{i} - y_{vj}^{f}\right)^2}{\sum_{j=1}^{n_v}\sum_{v=1}^{n}\left(y_{vj}^{i} - y_{vj}^{f}\right)^2} \tag{37}$$

$$RI(\%) = \sum_{j=1}^{n_v} y_{vj}\, y_{jo} \tag{38}$$

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, $hO_j$ absolute value of connection weights between the hidden and the output layers, $\sum_{j=1}^{n_v}\left(y_{vj}^{i} - y_{vj}^{f}\right)^2$ sum squared difference between initial connection weights and final connection weights from the input layer to the hidden layer, $\sum_{j=1}^{n_v}\sum_{v=1}^{n}\left(y_{vj}^{i} - y_{vj}^{f}\right)^2$ total of sum squared difference of all inputs. $\sum_{j=1}^{n_v} y_{vj} y_{jo}$ sum of product of final weights of connection from input neuron to hidden neurons with the connection from hidden neurons to output neurons. $j$ total number of hidden neurons. $o$ output neurons.

Another way to present the input variables is the SHAP value, which shows a features' importance by quantifying the contributions of each feature and by calculating the contributions of those features that have contributed to the prediction to the greatest extent. SHAP (SHapley Additive exPlanations) presented by Lundberg and Lee (Lundberg & Lee, 2017) is a unified framework for interpreting predictions, and its interpretation has been inspired by several methods (Bach et al., 2015; Ribeiro et al., 2016; Štrumbelj & Kononenko, 2014). The SHAP principal calculation (Lundberg et al., 2018) is shown in Eq. 39.

$$ShapValues_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(M - |S| - 1)!}{M!} \left[f_x(S \cup \{i\}) - f_x(S)\right] \tag{39}$$

where: $M$ number of input features. $N$ set of all input features. $S$ set of non-zero feature indices (features observed and not unknown). $f_x(S) = E[f(X)|X_s]$is the model's prediction for input $xx$, where $E[f(X)|X_s]$is the expected value of the function conditioned on a subset $S$ of input features.

## Results and discussion

Regarding the prediction of pillow effect and wall diameter by way of applying different models and structures, Table 9 depicts values of different validation metrics used for checking performance.

It is imperative to distinguish between training and test errors. Training errors are calculated using the same data as the ones used for training the model, but a stored full dataset unknown to the model is used for calculating test error. It can be concluded that the $R^2$ value of the training dataset implies variance within the trained samples through the model, whereas the $R^2$ value of the testing dataset indicates the predictive quality of the model. From Table 9, it is clear that there is a significant disparity between the different techniques in favor of the developed model as far as the prediction of the pillow effect is concerned. Using the features to predict a pillow and using the same features to predict the wall diameter are two different problems. The features can be readily used to predict the wall diameter. However, not all the prediction models can learn the connections among the data provided and use such information for the prediction of the pillow. A few possible reasons for this are the following: the problem has a stochastic nature, the data set lacks some critical features, the data are insufficient, the model is too simple for the problem, or the combination of any of the preceding causes. All the above-mentioned issues would cause a disparity in results with respect to the estimation of the model's real predicting capabilities in the case of unseen data. Therefore, upon comparing the $R^2$ of the testing with respect to all models and algorithms, it can be noted that the developed MLP model offered the best performance in the prediction of the pillow effect. The best performance of predicting the pillow effect was achieved by using BFGS Quasi-Newton (BFG)—Trainbfg as a training function and Symmetric Sigmoid (Tansig) as a transfer function. Regarding the prediction of wall diameter, the Gradient Boosting Regression (GBR) has the largest $R^2$ value as a model performance in terms of wall diameter prediction, and the developed MLP model with one output comes in second in terms of $R^2$ (see Fig. 14, which exhibits the techniques of the ANN used for predicting the pillow effect and wall diameter values of SPIF components). Due to the fact that the $R^2$ value of testing using the NBN technique is negative, $R^2$ was converted to zero for illustration purposes in Fig. 14. Since $R^2$ is defined as the percentage of variance explained by the fit, the fit can be worse than the simple application of a horizontal line for this purpose, in which case $R^2$ will be negative. Nevertheless, from a logical point of view, GBR's $R^2$ value is slightly larger than the $R^2$ value of the developed model, and all other validation metrics indicate that the developed model is better than GBR in terms of performance. The best performance of the developed MLP model to predict wall diameter was obtained by using the Levenberg–Marquardt (LM)—Trainlm training function and Softmax transfer function. It is noteworthy that the two outputs model was not able to record good performance concerning the other techniques. For checking the results of all the training and transfer functions, see Table 11, 12, 13, 14 in the Appendix.

Predictive testing data for both pillow effect and wall diameter using different ANN techniques versus the actual data are presented in Figs. 15a–d and 16a–d. The solid line displays an exact theoretical fit of actual and predicted values, with superimposed data over them. The distribution and deviation situated far away from the solid line are based on the model's ability to predict the pillow effect or wall diameter values with the lowest errors.

For discovering an alternative approach of predicting pillow effect and wall diameter in an easy, practical and faster way—instead of building, running, and evaluating a new ANN model each time recurringly—analytical equations for the prediction of pillow effect and wall diameter of parts formed by SPIF were extracted from the best model. This gave rise to a new method: by way of substituting only the process parameters, the obtained equations can be directly used for predicting either pillow effect or the wall diameter. Therefore, Eqs. 42 and 45 were formed, which needed constant weights and biases imported from the ANN network with the best performance. The extracted ANN network weights and biases functioned as one set of input weight (IW) and layer weight (LW). The IW is between the inputs and the hidden layer, and the LW is between the hidden and the output layers. The biases for each layer are ($b_1$) and ($b_2$). In the Appendix, Table 15 for pillow effect and Table 16 for wall diameter provide $b_1$, $b_2$, IW, and LW obtained from the best trained ANN model.

$$f(x) = tansig(x) = \frac{2}{\left(1 + exp^{(-2\times x)}\right) - 1} \tag{17}$$

**Table 9** Validation metrics for checking ANN structure and methods used for predicting pillow effect and wall diameter

| | ME | MAE | MSE | RMSE | MRE | SD | SEM | $R^2$ | Adj. $R^2$ |
|---|---|---|---|---|---|---|---|---|---|
| Neuron by neuron (NBN) | | | | | | | | | |
| Pillow effect | | | | | | | | | |
| Training | 0.0000 | 0.0021 | 0.0000 | 0.0034 | 0.0548 | 0.0034 | 0.0004 | 0.9997 | 0.9997 |
| Testing | – 0.3097 | 0.4077 | 0.8395 | 0.9163 | 4.5606 | 0.8826 | 0.1882 | – 32.4884 | 0.1635 |
| Wall diameter | | | | | | | | | |
| Training | 0.0000 | 0.0134 | 0.0010 | 0.0317 | 0.0003 | 0.0319 | 0.0034 | 0.9999 | 0.9999 |
| Testing | 2.0903 | 3.2505 | 29.6215 | 5.4426 | 0.0770 | 5.1434 | 1.0966 | – 1.3733 | 0.1437 |
| Gradient boosting regression (GBR) | | | | | | | | | |
| Pillow effect | | | | | | | | | |
| Training | 0.0000 | 0.0470 | 0.0061 | 0.0783 | 0.4147 | 0.0788 | 0.0085 | 0.8481 | 0.6142 |
| Testing | 0.0127 | 0.0772 | 0.0119 | 0.1092 | 1.6854 | 0.1110 | 0.0237 | 0.3315 | – 2.3002 |
| Wall diameter | | | | | | | | | |
| Training | 0.0000 | 0.0530 | 0.1005 | 0.3170 | 0.0012 | 0.3188 | 0.0344 | 0.9933 | 0.9934 |
| Testing | – 0.2163 | 1.0874 | 1.7549 | 1.3247 | 0.0265 | 1.3377 | 0.2852 | 0.9185 | 0.9227 |
| CatBoost | | | | | | | | | |
| Pillow effect | | | | | | | | | |
| Training | 0.0099 | 0.0688 | 0.0152 | 0.1235 | 1.2768 | 0.1238 | 0.0133 | 0.6062 | 0.1314 |
| Testing | 0.0018 | 0.0668 | 0.0090 | 0.0949 | 1.2411 | 0.0971 | 0.0207 | 0.6406 | 0.3196 |
| Wall diameter | | | | | | | | | |
| Training | 0.0052 | 0.5230 | 0.4959 | 0.7042 | 0.0129 | 0.7083 | 0.0764 | 0.9714 | 0.9677 |
| Testing | 0.1909 | 0.8511 | 1.3138 | 1.1462 | 0.0203 | 1.1568 | 0.2466 | 0.8947 | 0.8653 |
| Multilayer perceptron (MLP) One output | | | | | | | | | |
| Pillow effect | | | | | | | | | |
| Training | 0.0094 | 0.0477 | 0.0101 | 0.1007 | 0.8276 | 0.1009 | 0.0109 | 0.7506 | 0.6500 |
| Testing | – 0.0054 | 0.0559 | 0.0050 | 0.0704 | 2.0491 | 0.0719 | 0.0153 | 0.7129 | 0.6145 |
| Wall diameter | | | | | | | | | |
| Training | 0.0514 | 0.4167 | 0.5711 | 0.7557 | 0.0102 | 0.7584 | 0.0818 | 0.9645 | 0.9635 |
| Testing | 0.5275 | 1.0515 | 1.5923 | 1.2619 | 0.0257 | 1.1733 | 0.2501 | 0.9082 | 0.9116 |

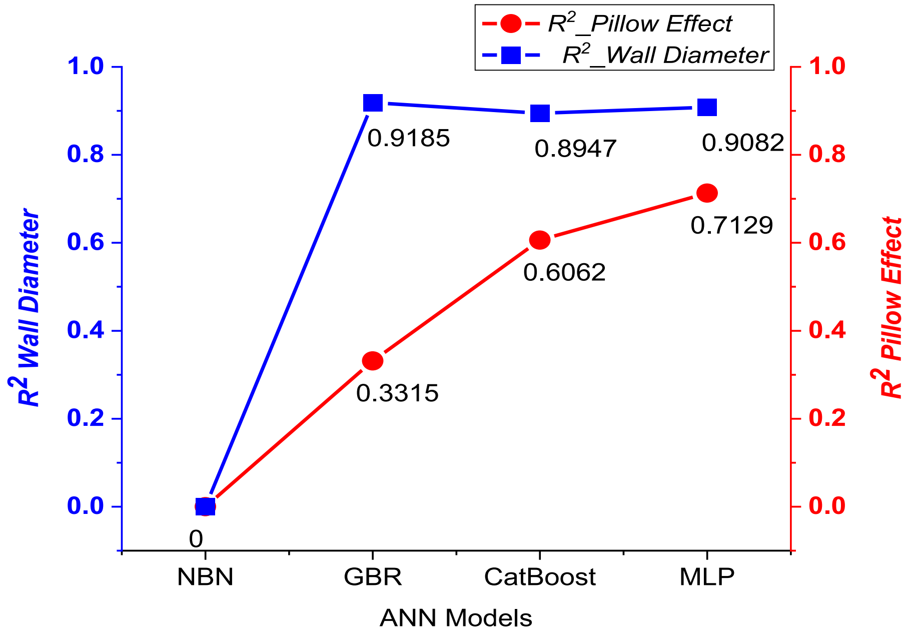


**Fig. 14** $R^2$ values for predicted results (pillow effect and wall diameter) in the case of different ANN models

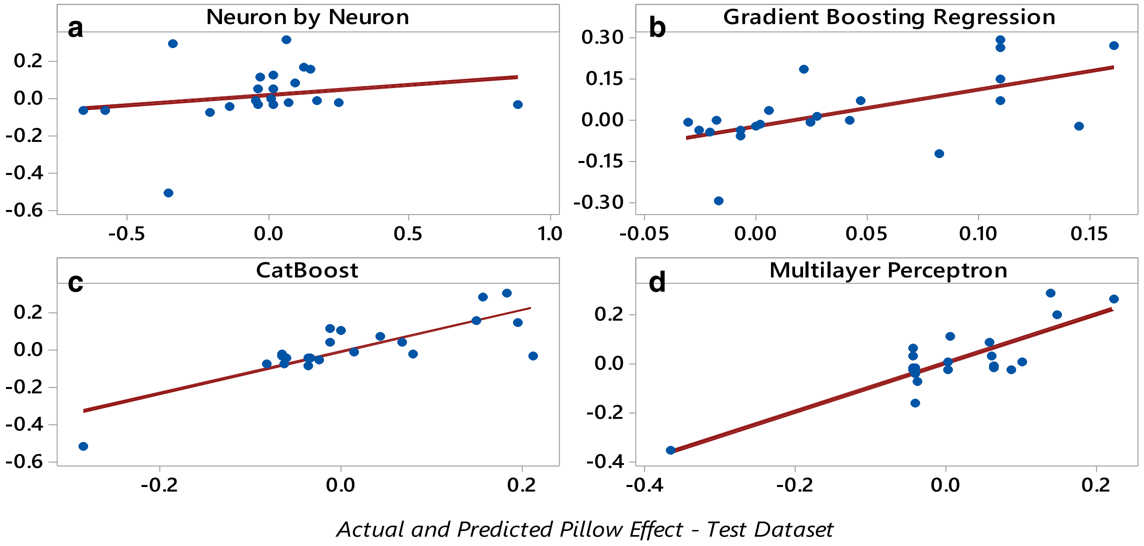


**Fig. 15** Actual and calculated values of pillow effect obtained with ANN models and algorithms in the case of **a** Neuron by neuron (NBN), **b** Gradient boosting regression (GBR), **c** CatBoost, and **d** Multilayer perceptron (MLP)

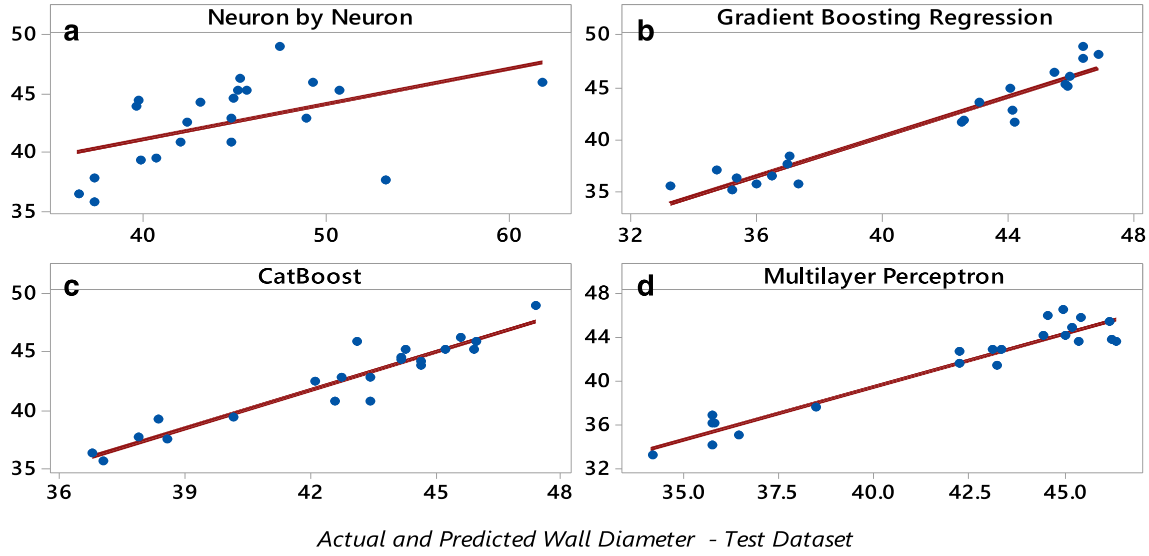


**Fig. 16** Actual and calculated values of wall diameter obtained with ANN models and algorithms in the case of **a** Neuron by neuron (NBN), **b** Gradient boosting regression (GBR), **c** CatBoost, and **d** multilayer perceptron (MLP)

$$Pillow\ Effect_i^{predict} = b_2 + LW \times tansig(b_1 + IW \times x) \tag{40}$$

$$Pillow\ Effect_i^{predict} = b_2 + LW \times \frac{2}{\left(1 + exp^{(-2\times(b_1 + IW\times x))}\right) - 1} \tag{41}$$

$$PillowEffect_i^{predict} = \overset{b_2}{[-4.4799]} + \overset{LW}{[4.8997 \; -2.9949 \; -1.8258 \; -4.4692 \; 2.3491 \; -0.0459 \; 0.9892 \; -10.9224 \; -11.2295 \; -15.6383]} \times \frac{2}{\left(1+exp\left(-2\times\left(\overset{b_1}{\begin{bmatrix}-1.0689\\2.5502\\4.8340\\5.3937\\0.5004\\6.4855\\1.7566\\2.0245\\-1.1407\\6.9462\end{bmatrix}} + \overset{IW}{\begin{bmatrix}-0.3664 & 3.4493 & -0.3515 & 2.7239 & 0.7225 & -1.4367 & 9.6762 & -3.1508 & 3.2435 & -14.1984\\-2.9456 & 1.8466 & -1.7678 & -0.8303 & 4.7506 & 6.6591 & -4.1411 & 6.9449 & -7.4108 & 8.0983\\-0.0801 & 2.2851 & 0.6648 & 4.9030 & 0.1742 & -1.2504 & 11.9422 & -0.5326 & 0.4992 & 3.6904\\-0.0311 & 3.3176 & -2.8060 & 2.1082 & -1.2817 & -3.6789 & 2.6040 & -8.7422 & -3.7858 & 1.1147\\0.1396 & 0.4012 & 2.3888 & -0.2148 & -0.1876 & -0.2164 & 4.8982 & -5.5040 & 6.1299 & -9.5227\end{bmatrix}} \times \overset{x}{\begin{bmatrix}ToolMaterial\\ToolShape\\ToolEndRadius\\ToolRoughness(Ra)\\ToolRoughness(Rz)\end{bmatrix}}\right)\right)\right)} - 1 \tag{42}$$

$$WallDiameter_i^{predict} = \overset{b_2}{[19.2151]} + \overset{LW}{[\,2.5135 \; 22.9679 \; 24.9085 \; 23.8584 \; -135.9051 \; 29.3786 \; 25.9349 \; 16.5500 \; 29.1113 \; -19.6351\,]} \times \frac{exp\left(\overset{b_1}{\begin{bmatrix}-2.1118\\1.0873\\-5.0721\\-4.6389\\-3.1617\\2.2753\\58.4812\\-22.5044\\-11.0040\\-13.1130\end{bmatrix}} + \overset{IW}{\begin{bmatrix}-0.2097 & 11.7171 & 1.5851 & -0.3582 & 0.8478 & 0.5419 & -21.3230 & -0.3622 & 5.2457 & 4.8888\\0.2117 & 2.9757 & -8.5864 & 3.2977 & -5.0426 & -5.9637 & 73.9425 & -17.8875 & -7.9919 & -35.0915\\-1.9420 & -77.4397 & 24.2144 & 25.8808 & 23.6070 & 22.0928 & -85.3340 & 32.3866 & 18.9395 & 20.2970\\-0.6388 & 144.0557 & 44.0436 & 112.4779 & 72.4671 & 73.4602 & -254.2707 & 49.4518 & 32.7593 & 13.5966\\-1.9055 & 7.5593 & 6.3161 & -1.9865 & 2.9167 & 3.1807 & -34.9560 & 7.7291 & 5.1248 & 7.2609\end{bmatrix}} \times \overset{x}{\begin{bmatrix}ToolMaterial\\ToolShape\\ToolEndRadius\\ToolRoughness(Ra)\\ToolRoughness(Rz)\end{bmatrix}}\right)}{\sum\left(exp\left(\overset{b_1}{\begin{bmatrix}-2.1118\\1.0873\\-5.0721\\-4.6389\\-3.1617\\2.2753\\58.4812\\-22.5044\\-11.0040\\-13.1130\end{bmatrix}} + \overset{IW}{\begin{bmatrix}-0.2097 & 11.7171 & 1.5851 & -0.3582 & 0.8478 & 0.5419 & -21.3230 & -0.3622 & 5.2457 & 4.8888\\0.2117 & 2.9757 & -8.5864 & 3.2977 & -5.0426 & -5.9637 & 73.9425 & -17.8875 & -7.9919 & -35.0915\\-1.9420 & -77.4397 & 24.2144 & 25.8808 & 23.6070 & 22.0928 & -85.3340 & 32.3866 & 18.9395 & 20.2970\\-0.6388 & 144.0557 & 44.0436 & 112.4779 & 72.4671 & 73.4602 & -254.2707 & 49.4518 & 32.7593 & 13.5966\\-1.9055 & 7.5593 & 6.3161 & -1.9865 & 2.9167 & 3.1807 & -34.9560 & 7.7291 & 5.1248 & 7.2609\end{bmatrix}} \times \overset{x}{\begin{bmatrix}ToolMaterial\\ToolShape\\ToolEndRadius\\ToolRoughness(Ra)\\ToolRoughness(Rz)\end{bmatrix}}\right)\right)} \tag{45}$$

$$f(x) = softmax(x) = \frac{exp^{(x)}}{\sum\left(exp^{(x)}\right)} \tag{16}$$

$$WallDiameter_i^{predict} = b_2 + LW \times softmax(b_1 + IW \times x) \tag{43}$$

$$WallDiameter_i^{predict} = b_2 + LW \times \frac{exp^{(b_1+IW\times x)}}{\sum\left(exp^{(b_1+IW\times x)}\right)} \tag{44}$$

With respect to relative importance and weights analysis, considerable significant factors affecting pillow effect and wall diameter are shown in Figs. 17a–d and 18a–d. Different methods based on weights and biases for finding the contributions of the input variables affecting output are pillow effect

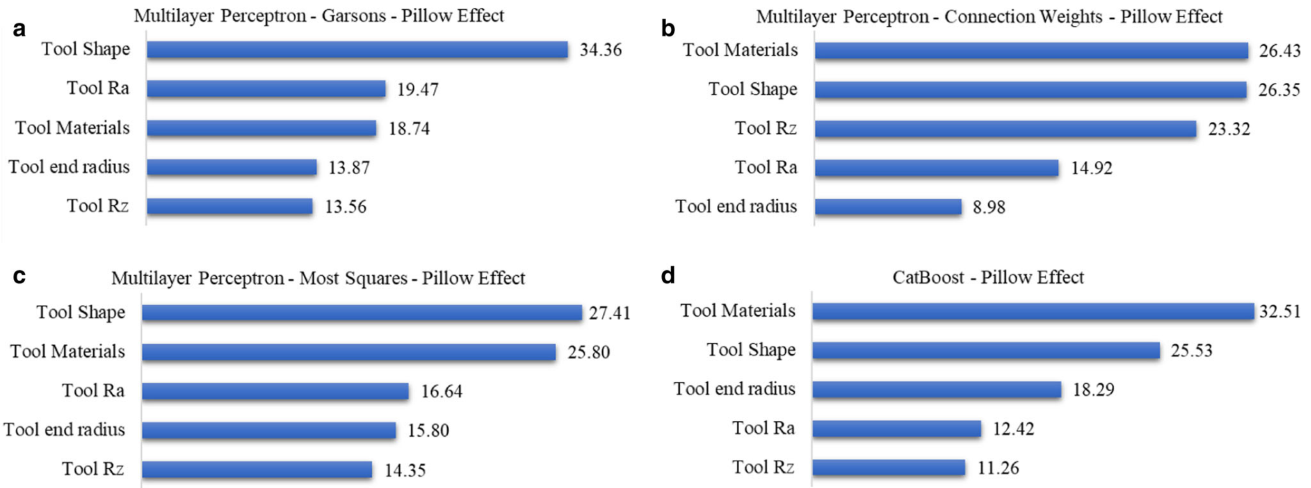


**Fig. 17** Relative importance of different input variables on pillow effect according to **a** Garson, **b** connection weights, **c** most-squares and **d** CatBoost algorithms

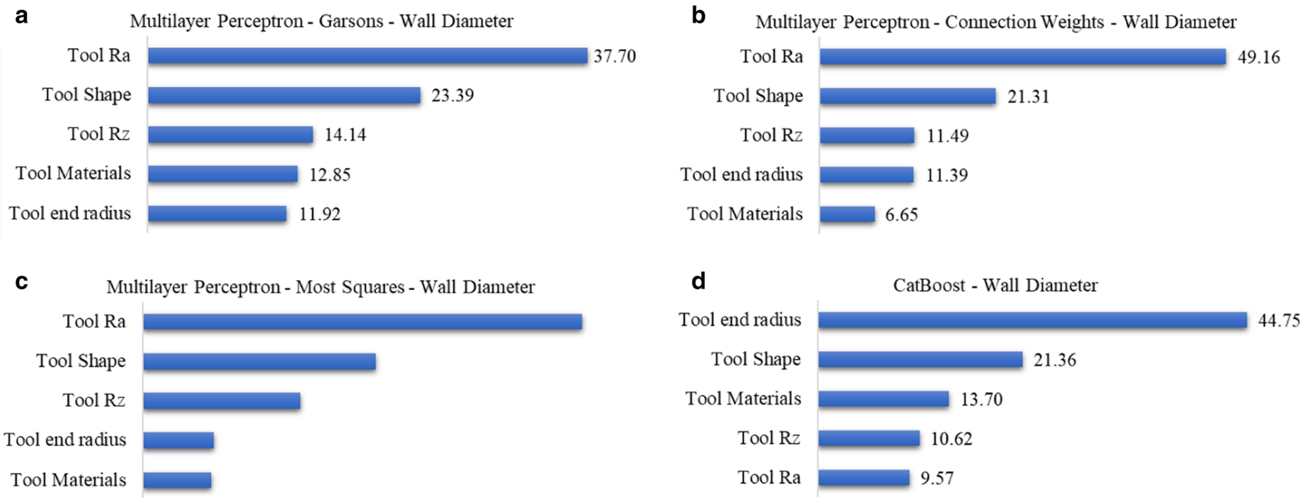


**Fig. 18** Relative importance of different input variables on wall diameter according to **a** Garson, **b** connection weights, **c** most-squares and **d** CatBoost algorithms

and wall diameter. Regarding pillow effect (see Fig. 17a–d), all of the methods show that changes in tool materials and tool shapes have a significant effect with insignificant variance on the Garson method (see Fig. 17a). Tooltip roughness (Rz) is listed at the end of the importance list as tooltip roughness has the lowest impact in all cases except for the connection weights method (see Fig. 17b). Concerning wall diameter (see Fig. 18a–d), tooltip roughness (Ra) has the most significant impact. On the other hand, the Catboost method indicates that tool end radius is the most significant impact on wall diameter (see Fig. 18d). Changes in tool shapes are always affected on the second level of the contribution's impact. Below, tool materials and tool end radius are listed with the exception of the CatBoost method.

In an attempt to do away with this controversy in the variation of the relative importance of input parameters as yielded by the different calculation methods, Fig. 19 was created, which show the average relative importance of the four previously mentioned methods. It has been concluded that tool materials and shapes have the most significant influence on pillow effect. Importantly, the difference in tool end radius recorded the lowest value. As for wall diameter, the surface roughness of the tool (Ra) is the highest effective value, followed by the change in tool shape. The least influential parameter was change of tool materials.

SHAP is a theoretical method that explains prediction by models. It can estimate and explain how each feature contributes and influences the model. This technique estimates each feature's contribution to each row of the dataset. The

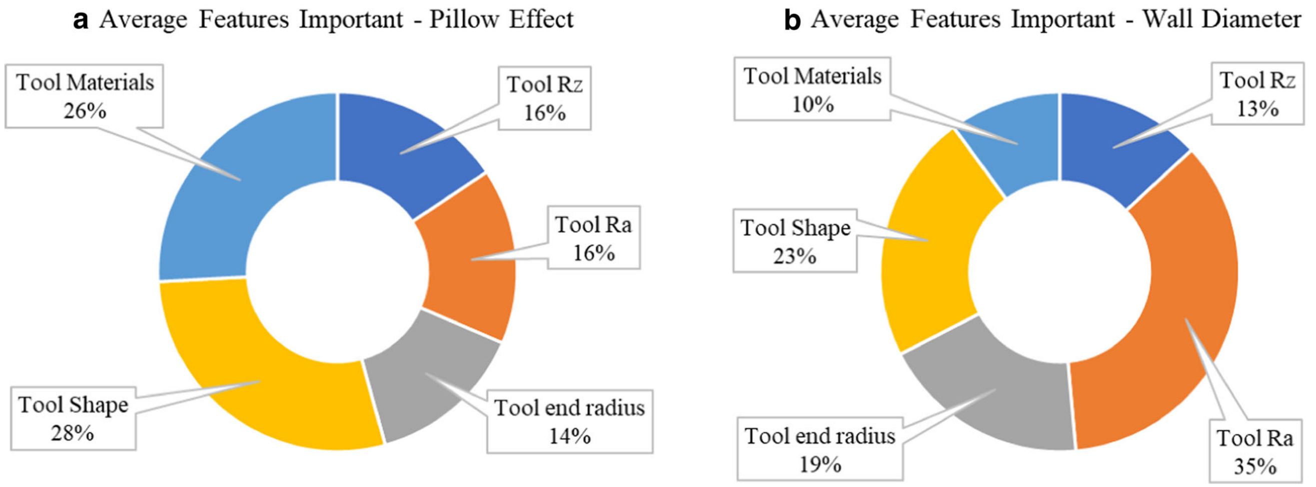


**Fig. 19** Average relative importance of different input variables on **a** pillow effect, and **b** wall diameter

**Fig. 20** Summary plot of SHAP value impact on pillow effect

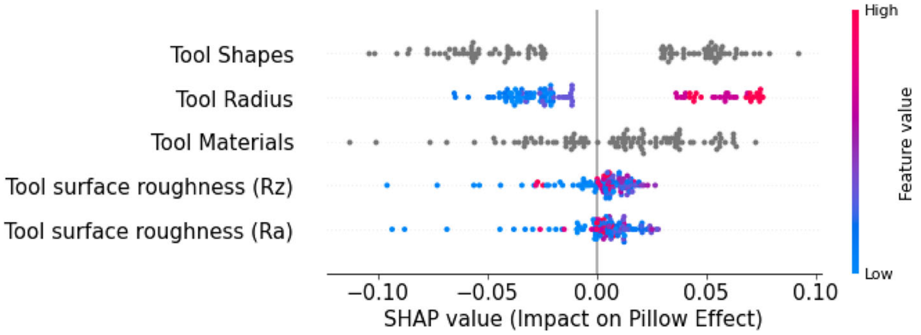
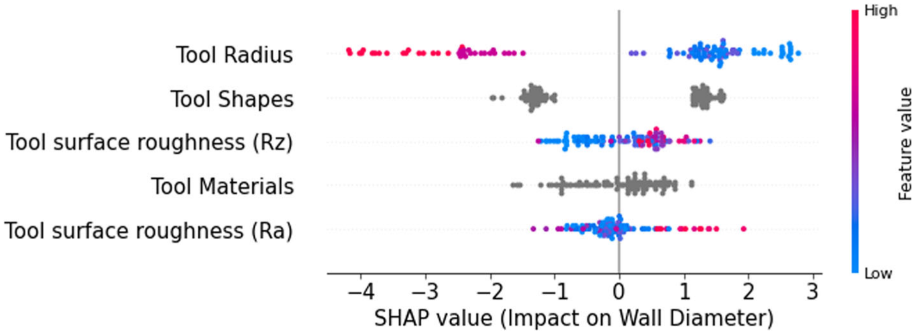


**Fig. 21** Summary plot of SHAP value impact on wall diameter



summary plots in Fig. 20 for pillow effect and Fig. 21 for wall diameter illustrate the individual feature's importance with respect to their feature effects. Each point on the summary plot represents a Shapley value for a feature. Axis Y defines the feature's level, and axis X shows Shapley values. The color of the plots denotes the value of the feature from lower to higher importance. The features are listed based on their importance. Shapley values represent the relative distribution of predictions among the features. It is worth stating that the same values of a certain feature can have different contributions towards an output, as dictated by other feature values for that same row.

Figures 20 and 21 represent every data point in each dataset feature as a single SHAP value on axis X and each feature on axis Y. The color bar indicates the feature's value: red means high values, and blue indicates low values. Grey points represent categorical inputs. Values on the right have a "positive" effect on the output, and values on the left have a "negative" effect. Positive and negative are merely directional terms and are related to the direction in which the model's output is affected. This, however, does not indicate the model's performance. For example, the most distant left point for tool radius is a high value for the tool radius feature in the first raw, therefore it appears in red. This high value for tool radius influenced wall diameter as a model output by

**Table 10** Three different examples of pillow prediction

| Tool Materials | Tool Shapes | Tool Radius | Tool surface roughness (Ra) | Tool surface roughness (Rz) | Actual Pillow Effect | Predict Pillow Effect | Condition |
|---|---|---|---|---|---|---|---|
| Aluminum | Hemispherical | 2 | 0.2 | 2.66 | 0.137 | 0.173 | Red (positive) |
| Copper | Flat | 0.3 | 0.05 | 0.65 | -0.026 | -0.291 | Blue (negative) |
| Polymer | Flat | 0.5 | 0.38 | 6.2 | -0.055 | -0.029 | (positive ± negative) |

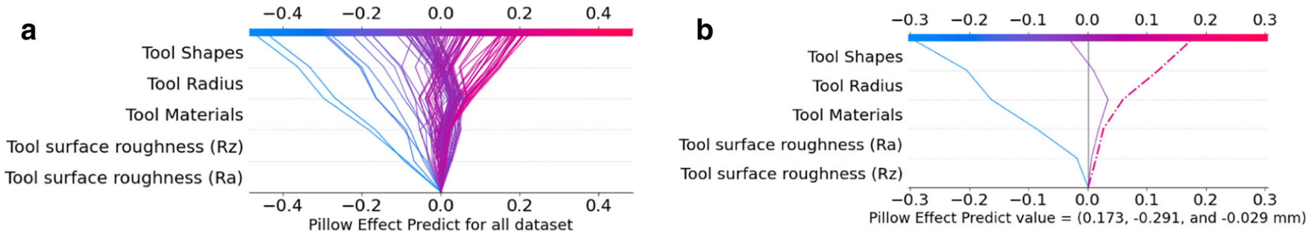


**Fig. 22** SHAP decision plot **a** all prediction values, and **b** three prediction values in Table 10

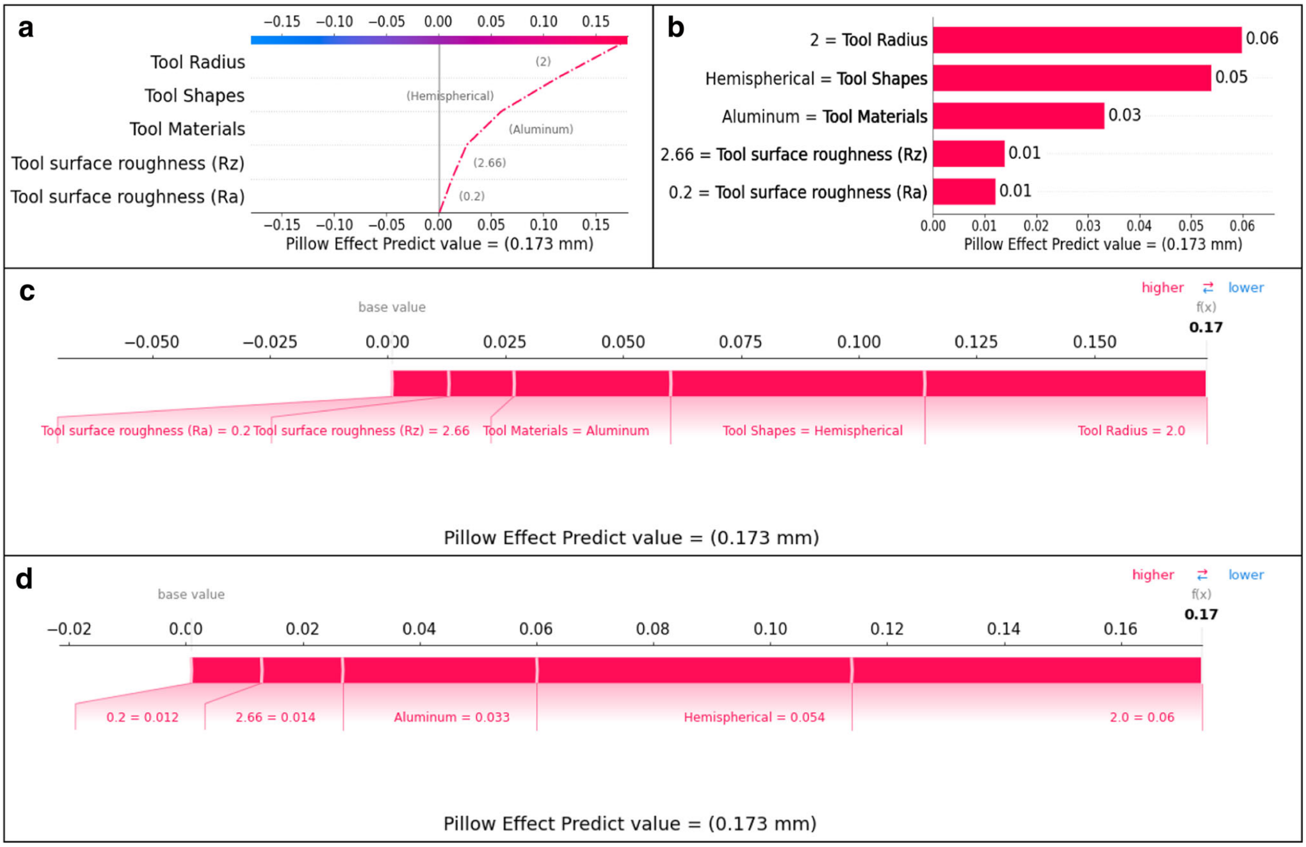


**Fig. 23** Total positive SHAP values: **a** SHAP decision plot, **b** SHAP bar plot, and **c** and **d** SHAP force plot

approximately − 4. The predictive model without that feature would have predicted a value of 4 or higher. Similarly, the rightmost red point of the surface roughness feature (Ra) with a value of 2 means that the absence of this value leads to the prediction of a wall diameter with a value below − 2. That means that broader extensions of the data point indicate the most effective features.

To understand how sharp value changes during the prediction process, three different examples shown in Table 10 illustrate three different behaviors: one on the left side (negative), one on the right side (positive), and one balanced

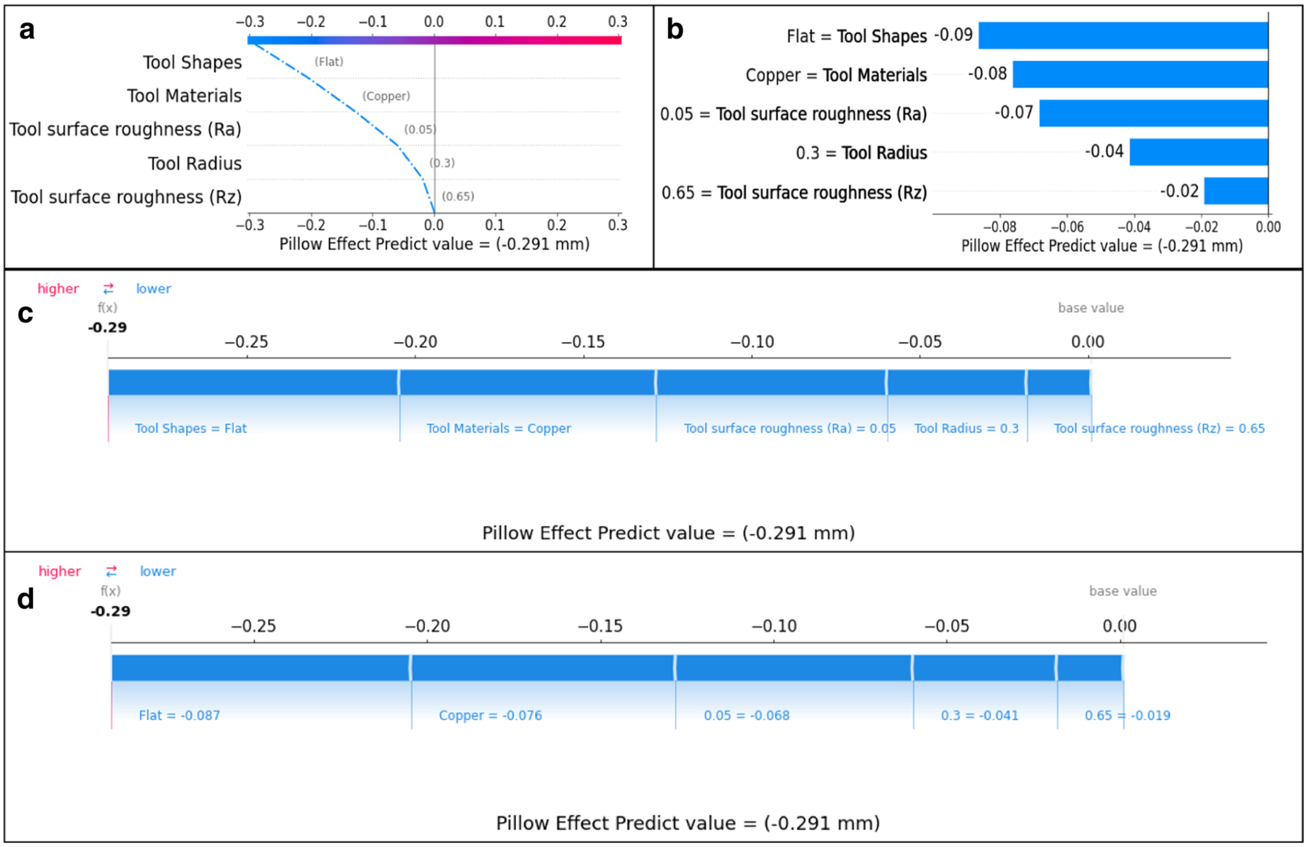


**Fig. 24** Total negative SHAP values: **a** SHAP decision plot, **b** SHAP bar plot, and **c** and **d** SHAP force plot

between the negative and positive sides. The colors used are only for purposes of demonstration.

The prediction models in Table 10 were visualized using a SHAP decision plot, which uses cumulative SHAP values. Each plotted line explains a single model prediction. Figure 22a plots all the pillow effect predictions, and Fig. 22b plots three prediction values as mentioned earlier.

Each value is represented individually: Fig. 23 shows total positive features values, Fig. 24 total negative features values, and Fig. 25 displays both positive and negative features values. The three figures mentioned earlier are presented on (a) a SHAP decision plot, (b), a SHAP bar plot, and (c and d) on a SHAP force plot. The SHAP force plot shows exactly which features had the most extensive influence on the model's prediction concerning individual observation values. The difference between (c and d) for all the figures is that (c) presents the value of the features, and (d) presents the feature SHAP value for each feature value.

As attested by the waterfall plots (Figs. 23, 24, 25) of various SPIF components selected in different conditions, the order of the impacts of parameters varied in the different components. This signifies that the estimation of parameters is the result of multiple factors. One single effective parameter that sufficiently affects output (pillow effect or wall diameter) cannot be captured, given that other parameters could affect the outcome of the model. Furthermore, complexity in the formability of parts using SPIF, which consists in stretching, bending, and shearing with cyclic effects, varies according to conditions, which thereby yields varying orders of impacting parameters in the individual components. Hence, this analysis reveals that each process condition is unique and that multiple factors interact and differently impact outcomes in individual parts. Also, forming tools with various materials show different hardness, which generates different surface strains. The same also holds true regarding different tool geometry. However, it is essential to point out that different effects on components occur if two tools made from two different materials but with identical geometry are used. This phenomenon is caused by different surface strains. Differing hardness values result in varying tool tip surface roughness, which thereby affects parts accuracy in terms of pillow effect and wall diameter. Furthermore, elastic deformation at the tooltip will produce dimensional inaccuracies of the formed part, and the plastic deformations will permanently damage the forming tool thereby excessively impacting the accuracy of the components, as explained by Kiridena et al. (2016).

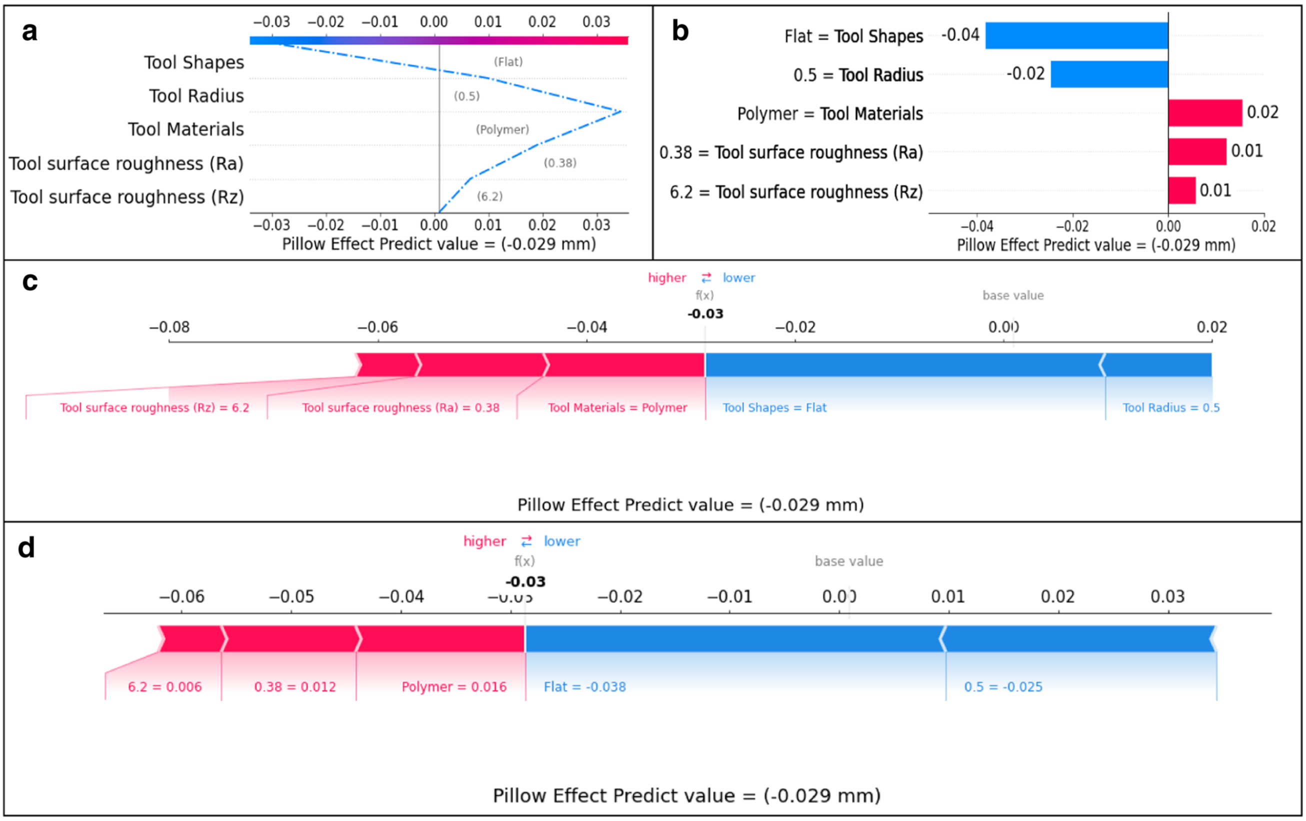


**Fig. 25** Positive and negative SHAP values: **a** SHAP decision plot, **b** SHAP bar plot, and **c** and **d** SHAP force plot

## Conclusions

This paper offered and described different algorithms and model structures of machine learning to predict pillow effect and wall diameter concerning the geometric accuracy of SPIF components produced from foil aluminum alloy sheets. The study's primary goal was to analyze the feature importance of SPIF parameters involved in the forming process with a view to determining the best model and architecture. Consequently, findings were adopted to derive two analytic equations for theoretically calculating pillow effect and wall diameter. The most significant findings of the study are as follows:

1. In the case of the training and testing datasets, with the help of $R^2$ values, CatBoost successfully predicted the wall diameter ranging from 0.9714 to 0.8947, and also successfully predicted pillow effect between the range of 0.6062 and 0.6406. Concerning $R^2$ values between 0.9645 and 0.9082 for wall diameter and between 0.7506 and 0.7129 for pillow effect, respectively, it was shown that the Levenberg–Marquardt training algorithm yielded the best performance as a prediction model based on different validation metrics. NBN offers no notable results, while GBR provides a reliable prediction of the wall diameter.
2. A one-output multilayer perceptron (MLP) solution network showed better results than a network with two outputs.
3. The most promising performance of predicting pillow effect was achieved by way of using BFGS Quasi-Newton (BFG)—Trainbfg as a training function, and Symmetric sigmoid (Tansig) as a transfer function.
4. The best performance of the developed MLP model to predict wall diameter was achieved by way of the Levenberg–Marquardt (LM)—Trainlm training function and softmax transfer function.
5. This research project marks the first time the relative importance (RI) method using SHapley Additive exPlanations (SHAP) was used to assess SPIF factors on outputs.
6. Relative importance (RI) revealed that tool materials and shapes are the most influential factors impacting the pillow effect. Surface roughness of the tool (Ra), followed by changes in tool shapes with the highest effective values on wall diameter.
7. The lowest effective parameter on pillow effect was tool end radius, and the lowest effective parameter for wall diameter was changes in tool materials.

8. Computation of the parameters is achieved by accumulating many factors; in reality, individual parameters are not sufficient enough to affect output (pillow effect or wall diameter). In other words, with respect to other parameter values in the same row, identical values of one parameter may contribute to an outcome in a number of different ways.

**Acknowledgements** The research reported in this paper was carried out in cooperation with the Budapest University of Technology and Economics (BME) and the Institute for Computer Science and Control (SZTAKI) and received support from the European Union project RRF-2.3.1-21-2022-00004 within the framework of the Artificial Intelligence National Laboratory. In addition, Imre Paniti would like to thank the project 'Thematic Excellence Programme – National Challenges Subprogramme – Establishment of the Center of Excellence for Autonomous Transport Systems at Széchenyi István University (TKP2020-NKA-14)' for its support. This research was also supported by Hungary's TEMPUS Public Foundation and Stipendium Hungaricum Scholarship. Also, the corresponding author expresses his gratitude to research engineers Andor Bálint Viharos and Donát Takács at Budapest University of Technology and Economics, Department of Manufacturing Science and Engineering, for their help in the implementation of the CatBoost code.

**Funding** Open access funding provided by Budapest University of Technology and Economics.

**Data availability** Not applicable.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest and have no competing interests.

**Ethical approval** Not applicable.

**Inform consent** Not applicable.

**Consent to publish** Not applicable.

## Appendix

See Tables 11, 12, 13, 14, 15, and 16.

**Table 11** Validation metrics of all multilayer perceptron (MLP) models for predicting pillow effect (one-output structure)

| Transfer function | | Elliotsig | Hardlim | Hardlims | Logsig | Netinv | Poslin | Purelin | Radbas | Radbasn | Satlin | Satlins | Softmax | Tansig | Tribas |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Training function | | Levenberg–Marquardt (LM)—Trainlm | | | | | | | | | | | | | |
| Training | $R^2$ | 0.6069 | 0.4164 | 0.4454 | 0.5286 | 0.4448 | 0.5376 | 0.3273 | 0.6116 | 0.5751 | 0.5701 | 0.5816 | 0.6209 | 0.4752 | 0.6232 |
| | MSE | 0.0136 | 0.0222 | 0.0187 | 0.0164 | 0.0183 | 0.0154 | 0.0214 | 0.0163 | 0.0160 | 0.0175 | 0.0166 | 0.0119 | 0.0170 | 0.0122 |
| Testing | $R^2$ | 0.5616 | 0.3902 | 0.4400 | 0.5044 | 0.4009 | 0.4928 | 0.1477 | 0.5889 | 0.5647 | 0.5272 | 0.4966 | 0.5430 | 0.4590 | 0.5239 |
| | MSE | 0.0175 | 0.0168 | 0.0250 | 0.0198 | 0.0283 | 0.0221 | 0.0434 | 0.0046 | 0.0123 | 0.0059 | 0.0109 | 0.0240 | 0.0267 | 0.0230 |
| Training function | | BFGS Quasi-Newton (BFG)—Trainbfg | | | | | | | | | | | | | |
| Training | $R^2$ | 0.5691 | 0.4484 | 0.4277 | 0.6378 | 0.5379 | 0.5975 | 0.3034 | 0.6156 | 0.5709 | 0.6463 | 0.5422 | 0.6154 | 0.7506 | 0.5031 |
| | MSE | 0.0166 | 0.0217 | 0.0188 | 0.0142 | 0.0176 | 0.0165 | 0.0216 | 0.0165 | 0.0178 | 0.0142 | 0.0148 | 0.0128 | 0.0101 | 0.0186 |
| Testing | $R^2$ | 0.5505 | 0.3638 | 0.3768 | 0.5748 | 0.4694 | 0.5504 | 0.2094 | 0.4406 | 0.5113 | 0.5779 | 0.4732 | 0.5955 | 0.7129 | 0.4809 |
| | MSE | 0.0113 | 0.0144 | 0.0300 | 0.0097 | 0.0147 | 0.0063 | 0.0427 | 0.0039 | 0.0071 | 0.0081 | 0.0261 | 0.0186 | 0.0050 | 0.0154 |
| Training function | | Resilient Backpropagation (RP)—Trainrp | | | | | | | | | | | | | |
| Training | $R^2$ | 0.5745 | 0.3992 | 0.4786 | 0.5864 | 0.6319 | 0.5540 | 0.3155 | 0.5787 | 0.6092 | 0.5383 | 0.5231 | 0.7374 | 0.6142 | 0.6152 |
| | MSE | 0.0140 | 0.0190 | 0.0220 | 0.0135 | 0.0142 | 0.0151 | 0.0225 | 0.0140 | 0.0163 | 0.0190 | 0.0177 | 0.0107 | 0.0124 | 0.0129 |
| Testing | R2 | 0.5528 | 0.3838 | 0.4300 | 0.5057 | 0.6117 | 0.5004 | 0.1356 | 0.4985 | 0.5741 | 0.4869 | 0.5167 | 0.6429 | 0.4799 | 0.6091 |
| | MSE | 0.0211 | 0.0323 | 0.0060 | 0.0242 | 0.0098 | 0.0220 | 0.0393 | 0.0235 | 0.0055 | 0.0069 | 0.0148 | 0.0063 | 0.0262 | 0.0175 |
| Training function | | Scaled Conjugate Gradient (SCG)—Trainscg | | | | | | | | | | | | | |
| Training | $R^2$ | 0.6235 | 0.4478 | 0.4277 | 0.5687 | 0.4018 | 0.4780 | 0.3071 | 0.5756 | 0.6524 | 0.5119 | 0.5169 | 0.5891 | 0.5629 | 0.5197 |
| | MSE | 0.0117 | 0.0217 | 0.0188 | 0.0183 | 0.0190 | 0.0201 | 0.0236 | 0.0163 | 0.0131 | 0.0209 | 0.0194 | 0.0148 | 0.0169 | 0.0174 |
| Testing | $R^2$ | 0.5969 | 0.4089 | 0.3933 | 0.5598 | 0.3943 | 0.4475 | 0.1162 | 0.5323 | 0.5693 | 0.4905 | 0.4891 | 0.5507 | 0.5336 | 0.4631 |
| | MSE | 0.0218 | 0.0133 | 0.0292 | 0.0048 | 0.0312 | 0.0142 | 0.0381 | 0.0120 | 0.0120 | 0.0045 | 0.0097 | 0.0157 | 0.0114 | 0.0183 |
| Training function | | Conjugate Gradient with Powell/Beale Restarts (CGB)—Traincgb | | | | | | | | | | | | | |
| Training | $R^2$ | 0.6208 | 0.3974 | 0.4461 | 0.5726 | 0.4133 | 0.5407 | 0.3137 | 0.5558 | 0.5772 | 0.5878 | 0.5611 | 0.7211 | 0.5744 | 0.5416 |
| | MSE | 0.0124 | 0.0180 | 0.0229 | 0.0165 | 0.0139 | 0.0177 | 0.0221 | 0.0178 | 0.0163 | 0.0177 | 0.0180 | 0.0111 | 0.0161 | 0.0181 |
| Testing | $R^2$ | 0.5576 | 0.3665 | 0.4172 | 0.5676 | 0.2155 | 0.4862 | 0.1744 | 0.4783 | 0.5396 | 0.5448 | 0.4693 | 0.6311 | 0.4757 | 0.4497 |
| | MSE | 0.0213 | 0.0377 | 0.0081 | 0.0110 | 0.0644 | 0.0134 | 0.0418 | 0.0104 | 0.0108 | 0.0036 | 0.0082 | 0.0078 | 0.0148 | 0.0124 |
| Training function | | Fletcher-Powell Conjugate Gradient (CGF)—Traincgf | | | | | | | | | | | | | |
| Training | $R^2$ | 0.5926 | 0.4076 | 0.4139 | 0.5738 | 0.3392 | 0.5757 | 0.2855 | 0.5429 | 0.6176 | 0.4781 | 0.5643 | 0.6167 | 0.5902 | 0.5670 |
| | MSE | 0.0161 | 0.0213 | 0.0193 | 0.0154 | 0.0271 | 0.0144 | 0.0235 | 0.0177 | 0.0145 | 0.0206 | 0.0175 | 0.0138 | 0.0168 | 0.0178 |
| Testing | $R^2$ | 0.5021 | 0.3460 | 0.3776 | 0.5287 | 0.2862 | 0.4809 | 0.2713 | 0.5371 | 0.5717 | 0.4721 | 0.5087 | 0.5281 | 0.5559 | 0.4573 |
| | MSE | 0.0074 | 0.0230 | 0.0296 | 0.0156 | 0.0107 | 0.0225 | 0.0346 | 0.0115 | 0.0104 | 0.0112 | 0.0092 | 0.0166 | 0.0066 | 0.0078 |
| Training function | | Polak-Ribiére Conjugate Gradient (CGP)—Traincgp | | | | | | | | | | | | | |
| Training | $R^2$ | 0.5169 | 0.4061 | 0.4043 | 0.6285 | 0.4037 | 0.4707 | 0.3107 | 0.5428 | 0.5659 | 0.5260 | 0.5102 | 0.7037 | 0.5416 | 0.4582 |

**Table 11** continued

| Transfer function | | Elliotsig | Hardlim | Hardlims | Logsig | Netinv | Poslin | Purelin | Radbas | Radbasn | Satlin | Satlins | Softmax | Tansig | Tribas |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | 0.0195 | 0.0192 | 0.0224 | 0.0135 | 0.0236 | 0.0188 | 0.0216 | 0.0190 | 0.0163 | 0.0116 | 0.0191 | 0.0121 | 0.0177 | 0.0221 |
| Testing | $R^2$ | 0.4775 | 0.3961 | 0.4010 | 0.5726 | 0.3017 | 0.4442 | 0.2186 | 0.5337 | 0.5620 | 0.4445 | 0.5071 | 0.6569 | 0.5225 | 0.4494 |
| | MSE | 0.0089 | 0.0301 | 0.0176 | 0.0148 | 0.0114 | 0.0208 | 0.0419 | 0.0063 | 0.0130 | 0.0447 | 0.0113 | 0.0055 | 0.0119 | 0.0092 |
| Training function | | One Step Secant (OSS)—Trainoss | | | | | | | | | | | | | |
| Training | $R^2$ | 0.5636 | 0.4284 | 0.3971 | 0.5526 | 0.4980 | 0.4344 | 0.2842 | 0.4905 | 0.6073 | 0.5125 | 0.4711 | 0.6282 | 0.5879 | 0.5470 |
| | MSE | 0.0171 | 0.0227 | 0.0200 | 0.0155 | 0.0188 | 0.0199 | 0.0244 | 0.0204 | 0.0120 | 0.0171 | 0.0215 | 0.0156 | 0.0132 | 0.0152 |
| Testing | $R^2$ | 0.5018 | 0.3980 | 0.3457 | 0.5321 | 0.4473 | 0.4117 | 0.2828 | 0.4569 | 0.5815 | 0.4932 | 0.4634 | 0.5761 | 0.5433 | 0.4725 |
| | MSE | 0.0087 | 0.0128 | 0.0308 | 0.0186 | 0.0164 | 0.0214 | 0.0311 | 0.0099 | 0.0222 | 0.0189 | 0.0085 | 0.0046 | 0.0232 | 0.0238 |
| Training function | | Variable Learning Rate Backpropagation (GDX)—Traingdx | | | | | | | | | | | | | |
| Training | $R^2$ | 0.4670 | 0.4107 | 0.4272 | 0.4890 | 0.4399 | 0.3926 | 0.2838 | 0.6090 | 0.6041 | 0.5486 | 0.5421 | 0.4926 | 0.5176 | 0.5546 |
| | MSE | 0.0153 | 0.0170 | 0.0228 | 0.0203 | 0.0217 | 0.0248 | 0.0291 | 0.0154 | 0.0154 | 0.0147 | 0.0158 | 0.0178 | 0.0187 | 0.0176 |
| Testing | $R^2$ | 0.4275 | 0.3328 | 0.3894 | 0.4777 | 0.4369 | 0.3871 | 0.2438 | 0.5185 | 0.5354 | 0.5051 | 0.5337 | 0.4462 | 0.4880 | 0.5388 |
| | MSE | 0.0368 | 0.0418 | 0.0125 | 0.0111 | 0.0137 | 0.0089 | 0.0127 | 0.0095 | 0.0111 | 0.0242 | 0.0185 | 0.0214 | 0.0127 | 0.0100 |
| Training function | | Bayesian Regularization Backpropagation (BRB)—Ttrainbr | | | | | | | | | | | | | |
| Training | $R^2$ | 0.5014 | 0.3846 | 0.4647 | 0.4062 | 0.5465 | 0.3673 | 0.2811 | 0.4342 | 0.4920 | 0.3920 | 0.3769 | 0.4155 | 0.3900 | 0.4189 |
| | MSE | 0.0161 | 0.0248 | 0.0203 | 0.0240 | 0.0114 | 0.0223 | 0.0212 | 0.0174 | 0.0211 | 0.0266 | 0.0184 | 0.0243 | 0.0202 | 0.0227 |
| Testing | $R^2$ | 0.4045 | 0.3728 | 0.4646 | 0.3952 | 0.5437 | 0.3121 | 0.2736 | 0.3851 | 0.4692 | 0.3437 | 0.3357 | 0.3726 | 0.3674 | 0.3554 |
| | MSE | 0.0293 | 0.0120 | 0.0152 | 0.0077 | 0.0356 | 0.0262 | 0.0439 | 0.0336 | 0.0073 | 0.0036 | 0.0404 | 0.0078 | 0.0295 | 0.0151 |

**Table 12** Validation metrics of all multilayer perceptron (MLP) models for predicting wall diameter (one-output structure)

| Transfer function | | Elliotsig | Hardlim | Hardlims | Logsig | Netinv | Poslin | Purelin | Radbas | Radbasn | Satlin | Satlins | Softmax | Tansig | Tribas |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Training function | | Levenberg–Marquardt (LM)—Trainlm | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9739 | 0.7813 | 0.8017 | 0.9754 | 0.7406 | 0.9270 | 0.7757 | 0.9085 | 0.8441 | 0.9176 | 0.8934 | 0.9645 | 0.9352 | 0.8813 |
| | MSE | 0.4168 | 3.4296 | 3.0891 | 0.3681 | 4.1758 | 1.2513 | 3.7503 | 1.4843 | 2.5128 | 1.4151 | 1.7518 | 0.5711 | 0.9973 | 1.8589 |
| Testing | $R^2$ | 0.8875 | 0.7699 | 0.7769 | 0.9049 | 0.7395 | 0.8583 | 0.7202 | 0.8574 | 0.7763 | 0.8757 | 0.8190 | 0.9082 | 0.8408 | 0.8571 |
| | MSE | 1.9503 | 4.1066 | 3.8937 | 2.0939 | 4.4728 | 1.8877 | 4.1884 | 2.4542 | 3.9164 | 1.5737 | 2.9857 | 1.5923 | 3.2663 | 2.7849 |
| Training function | | BFGS Quasi-Newton (BFG)—Trainbfg | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9539 | 0.7864 | 0.8195 | 0.9127 | 0.8424 | 0.8774 | 0.7782 | 0.8351 | 0.8819 | 0.7268 | 0.7906 | 0.8179 | 0.8646 | 0.7968 |
| | MSE | 0.7548 | 3.6113 | 2.9868 | 1.4368 | 2.4213 | 1.9760 | 3.3980 | 2.7406 | 1.8229 | 4.6529 | 3.2035 | 2.9821 | 2.1010 | 3.1879 |
| Testing | $R^2$ | 0.8828 | 0.7548 | 0.7836 | 0.8982 | 0.8293 | 0.8337 | 0.7117 | 0.7879 | 0.8379 | 0.6830 | 0.7881 | 0.7720 | 0.8494 | 0.7433 |
| | MSE | 1.8532 | 3.5133 | 3.2382 | 1.6402 | 3.4846 | 2.7878 | 5.9122 | 3.3141 | 3.1298 | 4.4670 | 4.3261 | 3.8099 | 2.9513 | 4.9723 |
| Training function | | Resilient Backpropagation (RP)—Trainrp | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9020 | 0.7772 | 0.7924 | 0.8960 | 0.8923 | 0.8567 | 0.7822 | 0.9641 | 0.8938 | 0.0000 | 0.7921 | 0.9332 | 0.8695 | 0.9233 |
| | MSE | 1.6266 | 3.5073 | 3.3432 | 1.6610 | 1.7017 | 2.3433 | 3.3454 | 0.6066 | 1.6927 | 17.4277 | 3.3642 | 1.0977 | 2.1212 | 1.1203 |
| Testing | $R^2$ | 0.8713 | 0.7677 | 0.7888 | 0.8827 | 0.8282 | 0.8474 | 0.6985 | 0.8953 | 0.8928 | 0.0000 | 0.7586 | 0.8408 | 0.8165 | 0.8690 |
| | MSE | 1.9143 | 4.3934 | 3.4769 | 2.1144 | 3.2539 | 2.1747 | 6.1829 | 1.5393 | 1.9366 | 12.5997 | 4.2176 | 2.6084 | 3.0704 | 3.0777 |
| Training function | | Scaled Conjugate Gradient (SCG)—Trainscg | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9461 | 0.8072 | 0.7892 | 0.9191 | – | 0.8031 | 0.7736 | 0.9311 | 0.8510 | 0.8034 | 0.7920 | 0.8695 | 0.9220 | 0.7362 |
| | MSE | 0.8372 | 3.2033 | 3.1442 | 1.2641 | – | 3.2589 | 3.9171 | 1.0754 | 2.3434 | 3.4135 | 3.5430 | 2.1061 | 1.2861 | 4.2042 |
| Testing | $R^2$ | 0.8539 | 0.7443 | 0.7482 | 0.8872 | – | 0.7877 | 0.7209 | 0.8859 | 0.8336 | 0.7966 | 0.7753 | 0.8243 | 0.8806 | 0.7185 |
| | MSE | 2.7108 | 4.0322 | 5.4987 | 2.2087 | – | 3.3951 | 3.6300 | 2.2414 | 3.2059 | 2.6148 | 3.0700 | 2.8376 | 1.9273 | 5.0874 |
| Training function | | Conjugate Gradient with Powell/Beale Restarts (CGB)—Traincgb | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9372 | 0.8013 | 0.8135 | 0.9436 | − 92.6822 | 0.8641 | 0.7964 | 0.8272 | 0.9019 | 0.7712 | 0.7738 | 0.8684 | 0.8782 | 0.7631 |
| | MSE | 0.9116 | 3.0843 | 3.2646 | 0.9412 | 1510.7393 | 2.0223 | 3.2727 | 2.8304 | 1.5584 | 3.4939 | 3.4923 | 2.2532 | 1.9552 | 3.9186 |
| Testing | $R^2$ | 0.8596 | 0.7979 | 0.7669 | 0.9044 | − 92.7471 | 0.8443 | 0.6047 | 0.8083 | 0.8871 | 0.7449 | 0.7411 | 0.8143 | 0.8337 | 0.7378 |
| | MSE | 2.9088 | 4.0518 | 2.8495 | 1.4280 | 1657.5713 | 3.4552 | 7.0541 | 3.1981 | 2.0255 | 5.3675 | 5.2447 | 2.5592 | 2.6887 | 4.0222 |
| Training function | | Fletcher-Powell Conjugate Gradient (CGF)—Traincgf | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9335 | 0.7718 | 0.7960 | 0.9364 | 0.7927 | 0.8564 | 0.7863 | 0.8099 | 0.8654 | 0.7623 | 0.7696 | 0.7747 | 0.8953 | 0.4537 |
| | MSE | 1.0702 | 3.9998 | 3.2288 | 1.0082 | 3.4882 | 2.4051 | 3.2547 | 3.0454 | 2.2851 | 4.0107 | 3.7607 | 3.4999 | 1.6175 | 8.8139 |
| Testing | $R^2$ | 0.8819 | 0.7703 | 0.7568 | 0.9046 | 0.6232 | 0.7965 | 0.7028 | 0.7533 | 0.7891 | 0.6987 | 0.7012 | 0.7139 | 0.8736 | 0.4092 |
| | MSE | 1.9902 | 2.7895 | 4.4917 | 1.6273 | 4.6690 | 2.8726 | 6.2983 | 4.4119 | 2.9833 | 4.4115 | 5.0248 | 5.7090 | 2.5713 | 10.4220 |
| Training function | | Polak-Ribiére Conjugate Gradient (CGP)—Traincgp | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9370 | 0.7959 | 0.7760 | 0.9424 | 0.8397 | 0.8654 | 0.7900 | 0.8406 | 0.8877 | 0.7664 | 0.8168 | 0.8804 | 0.9068 | 0.7358 |

**Table 12** continued

| Transfer function | | Elliotsig | Hardlim | Hardlims | Logsig | Netinv | Poslin | Purelin | Radbas | Radbasn | Satlin | Satlins | Softmax | Tansig | Tribas |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | 0.9963 | 3.3185 | 3.7280 | 0.9149 | 2.6346 | 2.2043 | 3.1077 | 2.6087 | 1.8038 | 3.9113 | 3.1721 | 2.0479 | 1.4414 | 4.1465 |
| Testing | $R^2$ | 0.9116 | 0.7875 | 0.7699 | 0.8907 | 0.8025 | 0.8273 | 0.7001 | 0.7809 | 0.8218 | 0.7067 | 0.8054 | 0.7949 | 0.8513 | 0.6881 |
| | MSE | 1.6690 | 3.5895 | 3.4755 | 2.0147 | 3.2524 | 2.5274 | 6.7425 | 3.5149 | 3.1615 | 4.2516 | 2.4530 | 2.6316 | 2.9971 | 6.0469 |
| Training function | | One Step Secant (OSS)—Trainoss | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9251 | 0.8148 | 0.8183 | 0.9278 | 0.8288 | 0.8302 | 0.7780 | 0.8087 | 0.8792 | 0.6920 | 0.7650 | 0.8495 | 0.8844 | 0.7917 |
| | MSE | 1.2774 | 2.8125 | 2.5908 | 1.1428 | 2.6992 | 2.8275 | 3.5820 | 3.0984 | 2.0460 | 4.7352 | 3.7642 | 2.2914 | 1.7778 | 3.4426 |
| Testing | $R^2$ | 0.8335 | 0.7454 | 0.7674 | 0.8709 | 0.8223 | 0.8259 | 0.7195 | 0.7254 | 0.8447 | 0.6276 | 0.7624 | 0.7711 | 0.8156 | 0.7693 |
| | MSE | 2.1661 | 5.4052 | 5.6189 | 2.2701 | 3.2326 | 2.6909 | 4.9303 | 4.7130 | 2.1318 | 7.6485 | 3.9696 | 3.9472 | 3.7436 | 3.7003 |
| Training function | | Variable Learning Rate Backpropagation (GDX)—Traingdx | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9133 | 0.8306 | 0.8131 | 0.9014 | 0.7690 | 0.8275 | 0.7596 | 0.9090 | 0.8536 | 0.6476 | 0.7962 | 0.7956 | 0.8556 | 0.8392 |
| | MSE | 1.4948 | 2.8233 | 3.0309 | 1.5750 | 3.8536 | 2.7909 | 3.6495 | 1.5171 | 2.5016 | 5.7460 | 3.3165 | 3.3443 | 2.4180 | 2.7252 |
| Testing | $R^2$ | 0.8145 | 0.7426 | 0.7564 | 0.8729 | 0.7438 | 0.7764 | 0.7553 | 0.8698 | 0.8420 | 0.6321 | 0.7776 | 0.7624 | 0.8280 | 0.8124 |
| | MSE | 2.4712 | 4.0109 | 3.7416 | 2.3217 | 3.9283 | 3.1915 | 5.1905 | 1.9479 | 2.1203 | 6.2480 | 3.8075 | 3.9211 | 2.5151 | 2.2809 |
| Training function | | Bayesian Regularization Backpropagation (BRB)—Ttrainbr | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9349 | 0.8653 | 0.8405 | 0.9538 | 0.9175 | 0.7663 | 0.7672 | 0.9550 | 0.9556 | 0.8906 | 0.8792 | 0.9533 | 0.9487 | 0.8780 |
| | MSE | 1.0627 | 2.2395 | 2.5979 | 0.7203 | 1.2703 | 3.4480 | 3.8854 | 0.6698 | 0.6859 | 1.8307 | 1.8631 | 0.7606 | 0.8701 | 1.9836 |
| Testing | $R^2$ | 0.9254 | 0.7763 | 0.8015 | 0.9053 | 0.8532 | 0.7607 | 0.7581 | 0.8879 | 0.9042 | 0.7714 | 0.8443 | 0.9170 | 0.8770 | 0.8234 |
| | MSE | 1.2541 | 3.0992 | 3.3374 | 1.8521 | 2.9703 | 5.4996 | 3.7460 | 2.4989 | 1.8950 | 3.0856 | 3.1689 | 1.4122 | 1.7511 | 2.8279 |

**Table 13** Validation metrics of all multilayer perceptron (MLP) models for predicting pillow effect (two-output structure)

| Transfer function | | Elliotsig | Hardlim | Hardlims | Logsig | Netinv | Poslin | Purelin | Radbas | Radbasn | Satlin | Satlins | Softmax | Tansig | Tribas |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Training function | | Levenberg–Marquardt (LM)—Trainlm | | | | | | | | | | | | | |
| Training | $R^2$ | 0.2834 | 0.2998 | 0.2566 | 0.3172 | – | 0.2982 | 0.1751 | 0.2342 | 0.1854 | 0.3071 | 0.2366 | 0.2651 | 0.2920 | 0.2683 |
| | MSE | 0.0760 | 0.0744 | 0.0224 | 0.0267 | – | 0.0208 | 0.0876 | 0.0281 | 0.0282 | 0.0200 | 0.0792 | 0.0279 | 0.0206 | 0.0759 |
| Testing | $R^2$ | 0.1854 | 0.1575 | 0.0974 | 0.1235 | – | 0.1201 | 0.0920 | 0.0346 | 0.0822 | 0.1115 | 0.1347 | 0.0784 | 0.0981 | 0.1587 |
| | MSE | 0.0302 | 0.0304 | 0.2969 | 0.2556 | – | 0.2913 | 0.0331 | 0.2845 | 0.2883 | 0.2898 | 0.0399 | 0.2774 | 0.2974 | 0.0388 |
| Training function | | BFGS Quasi-Newton (BFG)—Trainbfg | | | | | | | | | | | | | |
| Training | $R^2$ | 0.2419 | 0.2232 | 0.3234 | 0.2120 | 0.1571 | 0.3291 | 0.0990 | 0.2400 | 0.2391 | − 0.0001 | 0.0000 | 0.2286 | 0.1569 | 0.2012 |
| | MSE | 0.0786 | 0.0300 | 0.0255 | 0.0823 | 0.0946 | 0.0257 | 0.0940 | 0.0295 | 0.0284 | 0.0953 | 0.0953 | 0.0305 | 0.0891 | 0.0242 |
| Testing | $R^2$ | 0.0761 | 0.0498 | 0.1375 | 0.1847 | 0.0090 | 0.1733 | − 0.0171 | 0.0642 | 0.0714 | − 0.0049 | 0.0000 | 0.0458 | 0.1172 | 0.0972 |
| | MSE | 0.0427 | 0.2769 | 0.2610 | 0.0349 | 0.0109 | 0.2432 | 0.0446 | 0.2790 | 0.2783 | 0.0790 | 0.0793 | 0.2804 | 0.0341 | 0.2882 |
| Training function | | Resilient Backpropagation (RP)—Trainrp | | | | | | | | | | | | | |
| Training | $R^2$ | 0.2794 | 0.3001 | 0.2375 | 0.2824 | 0.2291 | 0.2412 | 0.1748 | 0.2035 | 0.2735 | 0.0000 | 0.1243 | 0.2486 | 0.1297 | 0.2256 |
| | MSE | 0.0269 | 0.0266 | 0.0857 | 0.0271 | 0.0278 | 0.0280 | 0.0872 | 0.0786 | 0.0271 | 0.1022 | 0.0846 | 0.0282 | 0.0893 | 0.0292 |
| Testing | $R^2$ | 0.1584 | 0.1343 | 0.1946 | 0.0961 | 0.0857 | 0.0728 | 0.0488 | 0.1812 | 0.0789 | − 0.0020 | 0.1173 | 0.1020 | 0.1292 | 0.1443 |
| | MSE | 0.2542 | 0.2563 | 0.0093 | 0.2714 | 0.2830 | 0.2802 | 0.0366 | 0.0536 | 0.2764 | 0.0521 | 0.0645 | 0.2688 | 0.0425 | 0.2583 |
| Training function | | Scaled Conjugate Gradient (SCG)—Trainscg | | | | | | | | | | | | | |
| Training | $R^2$ | 0.2222 | 0.2087 | 0.2562 | 0.2006 | – | 0.1367 | 0.1513 | 0.2929 | 0.2768 | 0.2112 | 0.1344 | 0.1208 | 0.1513 | 0.2309 |
| | MSE | 0.0827 | 0.0867 | 0.0834 | 0.0889 | – | 0.0336 | 0.0906 | 0.0220 | 0.0272 | 0.0280 | 0.0926 | 0.0852 | 0.0903 | 0.0287 |
| Testing | $R^2$ | 0.1126 | 0.1774 | 0.1122 | 0.1213 | – | 0.0419 | 0.0839 | 0.1242 | 0.1302 | 0.0525 | 0.0858 | 0.1178 | 0.1461 | 0.0723 |
| | MSE | 0.0320 | 0.0136 | 0.0118 | 0.0145 | – | 0.2831 | 0.0311 | 0.2792 | 0.2600 | 0.2880 | 0.0304 | 0.0635 | 0.0304 | 0.2813 |
| Training function | | Conjugate Gradient with Powell/Beale Restarts (CGB)—Traincgb | | | | | | | | | | | | | |
| Training | $R^2$ | 0.2374 | 0.3494 | 0.3173 | 0.2578 | – | 0.2227 | 0.1542 | 0.2285 | 0.2005 | 0.2080 | 0.2293 | 0.1392 | 0.2141 | 0.0754 |
| | MSE | 0.0824 | 0.0239 | 0.0265 | 0.0712 | – | 0.0806 | 0.0890 | 0.0288 | 0.0294 | 0.0295 | 0.0286 | 0.0956 | 0.0312 | 0.0969 |
| Testing | $R^2$ | 0.1839 | 0.1900 | 0.1307 | 0.1966 | – | 0.1562 | 0.1437 | 0.0775 | 0.0374 | 0.0865 | 0.0641 | − 0.0024 | 0.0183 | 0.0257 |
| | MSE | 0.0147 | 0.2478 | 0.2571 | 0.0612 | – | 0.0377 | 0.0342 | 0.2814 | 0.2907 | 0.2731 | 0.2853 | 0.0148 | 0.2849 | 0.0410 |
| Training function | | Fletcher-Powell Conjugate Gradient (CGF)—Traincgf | | | | | | | | | | | | | |
| Training | $R^2$ | 0.2930 | 0.2522 | 0.2498 | 0.2256 | – | 0.0625 | 0.1764 | 0.1225 | 0.2797 | 0.1029 | 0.1276 | 0.1164 | 0.1757 | − 0.0002 |
| | MSE | 0.0268 | 0.0284 | 0.0270 | 0.0812 | – | 0.0367 | 0.0879 | 0.0980 | 0.0788 | 0.0938 | 0.0935 | 0.0985 | 0.0881 | 0.1044 |
| Testing | $R^2$ | 0.1704 | 0.0863 | 0.0962 | 0.2100 | – | 0.0222 | 0.1337 | 0.0334 | 0.1841 | 0.0612 | 0.0489 | 0.0459 | 0.1033 | − 0.0005 |
| | MSE | 0.2459 | 0.2726 | 0.2772 | 0.0329 | – | 0.2885 | 0.0300 | 0.0139 | 0.0128 | 0.0396 | 0.0313 | 0.0148 | 0.0301 | 0.0435 |
| Training function | | Polak-Ribiére Conjugate Gradient (CGP)—Traincgp | | | | | | | | | | | | | |
| Training | $R^2$ | 0.2338 | 0.2333 | 0.1983 | 0.2029 | – | − 0.0272 | 0.2623 | 0.1976 | 0.1779 | 0.1335 | 0.1305 | 0.1173 | 0.1426 | 0.1664 |

**Table 13** continued

| Transfer function | | Elliotsig | Hardlim | Hardlims | Logsig | Netinv | Poslin | Purelin | Radbas | Radbasn | Satlin | Satlins | Softmax | Tansig | Tribas |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | 0.0802 | 0.0849 | 0.0893 | 0.0830 | – | 0.1020 | 0.0216 | 0.0836 | 0.0863 | 0.0860 | 0.0959 | 0.0861 | 0.0954 | 0.0815 |
| Testing | R$^2$ | 0.1945 | 0.0360 | 0.0837 | 0.1624 | – | − 0.0422 | 0.1025 | 0.0908 | 0.1093 | 0.0988 | − 0.0280 | 0.0020 | 0.0492 | 0.0240 |
| | MSE | 0.0342 | 0.0164 | 0.0131 | 0.0372 | – | 0.0653 | 0.2969 | 0.0400 | 0.0368 | 0.0567 | 0.0203 | 0.0677 | 0.0118 | 0.0677 |
| Training function | | One Step Secant (OSS)—Trainoss | | | | | | | | | | | | | |
| Training | R$^2$ | 0.1863 | 0.4232 | 0.3006 | 0.1842 | 0.1137 | 0.1736 | 0.1961 | 0.2259 | 0.1412 | − 0.0001 | 0.0000 | 0.2250 | 0.1947 | 0.1080 |
| | MSE | 0.0842 | 0.0168 | 0.0257 | 0.0901 | 0.0989 | 0.0911 | 0.0310 | 0.0281 | 0.0916 | 0.1022 | 0.0966 | 0.0295 | 0.0219 | 0.0970 |
| Testing | R$^2$ | 0.1634 | 0.2360 | 0.1263 | 0.1366 | − 0.0320 | 0.1212 | 0.0132 | 0.0943 | 0.1378 | − 0.0060 | − 0.0185 | 0.0511 | 0.0272 | 0.0616 |
| | MSE | 0.0395 | 0.2465 | 0.2661 | 0.0154 | 0.0127 | 0.0150 | 0.2923 | 0.2770 | 0.0299 | 0.0522 | 0.0743 | 0.2780 | 0.3279 | 0.0247 |
| Training function | | Variable Learning Rate Backpropagation (GDX)—Traingdx | | | | | | | | | | | | | |
| Training | R$^2$ | 0.2127 | 0.2087 | 0.1761 | 0.2817 | 0.1317 | 0.1763 | 0.1177 | − 0.2559 | 0.2122 | 0.1327 | 0.1366 | 0.1345 | 0.1263 | 0.1184 |
| | MSE | 0.0804 | 0.0839 | 0.0879 | 0.0271 | 0.0965 | 0.0897 | 0.0975 | 0.1197 | 0.0180 | 0.0891 | 0.0923 | 0.0919 | 0.0313 | 0.0967 |
| Testing | R$^2$ | 0.1848 | 0.1942 | 0.1515 | 0.0845 | 0.0383 | 0.0255 | 0.0211 | − 0.4005 | 0.0509 | 0.1086 | 0.0972 | 0.0889 | 0.0961 | 0.0461 |
| | MSE | 0.0417 | 0.0300 | 0.0294 | 0.2746 | 0.0145 | 0.0245 | 0.0157 | 0.1104 | 0.3208 | 0.0441 | 0.0304 | 0.0335 | 0.2815 | 0.0176 |
| Training function | | Bayesian Regularization Backpropagation (BRB)—Ttrainbr | | | | | | | | | | | | | |
| Training | R$^2$ | 0.2823 | 0.2409 | 0.2322 | 0.3168 | 0.2333 | 0.2316 | 0.2242 | 0.2394 | 0.2687 | 0.2540 | 0.2326 | 0.1440 | 0.2727 | 0.1434 |
| | MSE | 0.0270 | 0.0796 | 0.0801 | 0.0205 | 0.0824 | 0.0203 | 0.0298 | 0.0790 | 0.0273 | 0.0291 | 0.0284 | 0.0905 | 0.0281 | 0.0903 |
| Testing | R$^2$ | 0.1076 | 0.0674 | 0.2138 | 0.1575 | 0.2150 | 0.0845 | 0.1103 | 0.2080 | 0.1106 | 0.0563 | 0.0924 | 0.1403 | 0.1016 | 0.1172 |
| | MSE | 0.2715 | 0.0387 | 0.0337 | 0.2729 | 0.0244 | 0.3125 | 0.2651 | 0.0359 | 0.2690 | 0.2759 | 0.2745 | 0.0329 | 0.2669 | 0.0349 |

**Table 14** Validation metrics of all multilayer perceptron (MLP) models for predicting wall diameter (two-output structure)

| Transfer function | | Elliotsig | Hardlim | Hardlims | Logsig | Netinv | Poslin | Purelin | Radbas | Radbasn | Satlin | Satlins | Softmax | Tansig | Tribas |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Training function | | Levenberg–Marquardt (LM)—Trainlm | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9486 | 0.7936 | 0.7140 | 0.8720 | – | 0.9233 | 0.7924 | 0.9771 | 0.9750 | 0.9178 | 0.9121 | 0.9186 | 0.9764 | 0.8811 |
| | MSE | 0.8884 | 3.5224 | 4.8086 | 2.1768 | – | 1.1936 | 3.4987 | 0.3346 | 0.4063 | 1.3198 | 1.4320 | 1.3186 | 0.4021 | 1.9642 |
| Testing | $R^2$ | 0.7973 | 0.7812 | 0.6444 | 0.8264 | – | 0.8472 | 0.6417 | 0.8234 | 0.8034 | 0.7738 | 0.8537 | 0.7873 | 0.9020 | 0.8222 |
| | MSE | 2.5381 | 2.9832 | 4.9774 | 2.4691 | – | 2.8015 | 5.3051 | 4.0098 | 3.3227 | 4.0554 | 2.4790 | 3.4807 | 1.3959 | 2.7165 |
| Training function | | BFGS Quasi-Newton (BFG)—Trainbfg | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9636 | 0.6656 | 0.7801 | 0.9097 | 0.8214 | 0.8590 | 0.7753 | 0.8004 | 0.8057 | – 0.0003 | 0.0000 | 0.7670 | 0.7647 | 0.5446 |
| | MSE | 0.5233 | 5.4678 | 3.7569 | 1.5153 | 2.8993 | 2.2897 | 3.5001 | 3.0018 | 3.2379 | 15.8342 | 16.3426 | 3.8868 | 3.8193 | 7.4966 |
| Testing | $R^2$ | 0.9060 | 0.6153 | 0.7432 | 0.7981 | 0.7741 | 0.7231 | 0.6828 | 0.7460 | 0.6829 | – 0.0023 | -0.0191 | 0.7491 | 0.6580 | 0.4434 |
| | MSE | 2.2658 | 6.3818 | 3.5380 | 2.9499 | 3.7787 | 4.3209 | 5.8728 | 5.4646 | 4.8493 | 18.8295 | 16.9199 | 3.8681 | 5.8774 | 7.6693 |
| Training function | | Resilient Backpropagation (RP)—Trainrp | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9091 | 0.7069 | 0.7770 | 0.8970 | 0.8737 | 0.8057 | 0.7851 | 0.9466 | 0.8433 | – 0.0094 | 0.7973 | 0.8579 | 0.7870 | 0.8895 |
| | MSE | 1.4999 | 4.7026 | 3.8004 | 1.7248 | 2.0462 | 3.1402 | 3.6210 | 0.8256 | 2.7021 | 17.4064 | 3.5062 | 2.3886 | 3.4721 | 1.6749 |
| Testing | $R^2$ | 0.8285 | 0.6705 | 0.7319 | 0.8275 | 0.7266 | 0.7394 | 0.6568 | 0.8305 | 0.7365 | – 0.0456 | 0.7361 | 0.8567 | 0.7159 | 0.8490 |
| | MSE | 2.7313 | 5.8855 | 3.7410 | 2.4319 | 4.5162 | 4.4992 | 4.7300 | 2.9276 | 3.5100 | 13.8117 | 3.2163 | 2.1325 | 4.7419 | 3.1871 |
| Training function | | Scaled Conjugate Gradient (SCG)—Trainscg | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9299 | 0.7872 | 0.8090 | 0.9448 | – | 0.7818 | 0.7906 | 0.9543 | 0.8464 | 0.7788 | 0.7649 | 0.8229 | 0.8866 | 0.7292 |
| | MSE | 1.1767 | 3.5600 | 3.0563 | 0.8851 | – | 3.5724 | 3.3645 | 0.7828 | 2.2308 | 3.6911 | 3.8929 | 2.9074 | 1.8375 | 4.5923 |
| Testing | $R^2$ | 0.8784 | 0.7649 | 0.6940 | 0.9017 | – | 0.7493 | 0.6715 | 0.8548 | 0.7777 | 0.7216 | 0.7236 | 0.7285 | 0.8823 | 0.5343 |
| | MSE | 1.7376 | 3.1582 | 5.3264 | 1.7171 | – | 4.1505 | 5.8751 | 1.9898 | 5.2154 | 3.8934 | 4.3635 | 4.4934 | 2.0448 | 6.6622 |
| Training function | | Conjugate Gradient with Powell/Beale Restarts (CGB)—Traincgb | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9232 | 0.7612 | 0.7621 | 0.9299 | – | 0.8770 | 0.7722 | 0.7471 | 0.8022 | 0.7179 | 0.7344 | 0.6945 | 0.8848 | 0.6053 |
| | MSE | 1.3124 | 3.9341 | 3.8826 | 1.1467 | – | 2.0733 | 3.6260 | 4.2613 | 3.2290 | 4.5961 | 4.5686 | 5.1770 | 1.9501 | 6.6277 |
| Testing | $R^2$ | 0.7891 | 0.7317 | 0.6107 | 0.9007 | – | 0.8349 | 0.7259 | 0.5834 | 0.7765 | 0.5539 | 0.5737 | 0.6190 | 0.7196 | 0.5040 |
| | MSE | 2.6485 | 4.1726 | 6.5326 | 1.6613 | – | 2.4437 | 4.7387 | 6.0172 | 3.7463 | 7.1545 | 5.7479 | 5.4692 | 3.9212 | 7.0373 |
| Training function | | Fletcher-Powell Conjugate Gradient (CGF)—Traincgf | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9244 | 0.7543 | 0.7918 | 0.9440 | – | 0.8607 | 0.7789 | 0.7859 | 0.8187 | 0.6840 | 0.7099 | 0.7869 | 0.8401 | 0.0000 |
| | MSE | 1.1911 | 4.2275 | 3.4527 | 0.8909 | – | 2.3069 | 3.6660 | 3.3036 | 2.8287 | 5.7485 | 5.1715 | 3.3774 | 2.5907 | 16.5412 |
| Testing | $R^2$ | 0.8237 | 0.6943 | 0.7377 | 0.8932 | – | 0.6938 | 0.7018 | 0.7627 | 0.7986 | 0.6050 | 0.7086 | 0.7791 | 0.7524 | -0.0121 |
| | MSE | 3.3678 | 3.8042 | 4.1484 | 1.9435 | – | 4.7990 | 4.6454 | 4.8335 | 3.7055 | 3.5087 | 3.2159 | 4.1252 | 4.2896 | 16.0993 |
| Training function | | Polak-Ribiére Conjugate Gradient (CGP)—Traincgp | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9255 | 0.7827 | 0.7185 | 0.9184 | – | 0.8288 | 0.7843 | 0.7834 | 0.7906 | 0.5254 | 0.8137 | 0.8043 | 0.8406 | 0.5096 |

**Table 14** continued

| Transfer function | | Elliotsig | Hardlim | Hardlims | Logsig | Netinv | Poslin | Purelin | Radbas | Radbasn | Satlin | Satlins | Softmax | Tansig | Tribas |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | 1.1373 | 3.5112 | 4.7617 | 1.3561 | – | 2.7748 | 3.6430 | 3.6217 | 3.3624 | 7.7538 | 3.2067 | 3.2360 | 2.5704 | 8.3034 |
| Testing | $R^2$ | 0.8575 | 0.7638 | 0.6184 | 0.7737 | – | 0.6873 | 0.6621 | 0.6942 | 0.7839 | 0.4332 | 0.6333 | 0.6181 | 0.7439 | 0.4435 |
| | MSE | 2.9861 | 4.1352 | 5.4426 | 3.2920 | – | 5.0056 | 4.9207 | 4.6891 | 3.8480 | 9.4174 | 4.5441 | 5.9696 | 4.1253 | 7.5217 |
| Training function | | One Step Secant (OSS)—Trainoss | | | | | | | | | | | | | |
| Training | $R^2$ | 0.8135 | 0.6721 | 0.7035 | 0.9227 | 0.7710 | 0.7911 | 0.7749 | 0.7870 | 0.8498 | 0.0000 | -0.0022 | 0.8002 | 0.7730 | 0.5918 |
| | MSE | 3.2585 | 5.3760 | 4.8231 | 1.1757 | 3.6232 | 3.1966 | 3.4535 | 3.4595 | 2.4141 | 14.9927 | 16.2938 | 3.2602 | 3.6721 | 6.7295 |
| Testing | $R^2$ | 0.7903 | 0.6140 | 0.6830 | 0.8992 | 0.7265 | 0.7152 | 0.7244 | 0.7786 | 0.8397 | 0.0000 | -0.0155 | 0.6433 | 0.7341 | 0.5283 |
| | MSE | 2.6031 | 6.4083 | 5.1049 | 2.1261 | 5.0310 | 5.9013 | 5.6786 | 3.7967 | 2.8694 | 22.1187 | 17.3556 | 5.7912 | 4.5024 | 6.7972 |
| Training function | | Variable Learning Rate Backpropagation (GDX)—Traingdx | | | | | | | | | | | | | |
| Training | $R^2$ | 0.8771 | 0.7039 | 0.7772 | 0.8766 | 0.6692 | 0.7410 | 0.7602 | 0.8901 | 0.8793 | 0.4797 | 0.7718 | 0.7637 | 0.7829 | 0.6181 |
| | MSE | 2.0124 | 4.7329 | 4.0211 | 2.1858 | 4.8649 | 3.9782 | 4.0478 | 1.6068 | 2.0628 | 8.7483 | 3.8178 | 3.8704 | 3.3021 | 6.6961 |
| Testing | $R^2$ | 0.7984 | 0.6727 | 0.6575 | 0.8210 | 0.6527 | 0.7371 | 0.7327 | 0.8444 | 0.8281 | 0.4518 | 0.7092 | 0.6510 | 0.7693 | 0.4445 |
| | MSE | 3.3303 | 5.9067 | 3.2423 | 2.0509 | 7.8229 | 5.3882 | 3.8492 | 3.3088 | 2.3310 | 7.9031 | 4.4105 | 5.2580 | 4.7037 | 6.7597 |
| Training function | | Bayesian Regularization Backpropagation (BRB)—Ttrainbr | | | | | | | | | | | | | |
| Training | $R^2$ | 0.9221 | 0.7955 | 0.7778 | 0.9631 | 0.8718 | 0.8283 | 0.7588 | 0.9439 | 0.8673 | 0.8712 | 0.8526 | 0.9001 | 0.8835 | 0.8781 |
| | MSE | 1.3452 | 3.2747 | 3.4220 | 0.6111 | 2.0802 | 2.8301 | 3.9895 | 0.9206 | 2.0848 | 2.0121 | 2.4967 | 1.6578 | 1.8761 | 2.0434 |
| Testing | $R^2$ | 0.8269 | 0.7727 | 0.7468 | 0.8303 | 0.7128 | 0.8074 | 0.7431 | 0.7669 | 0.7938 | 0.8452 | 0.8337 | 0.7174 | 0.7970 | 0.7360 |
| | MSE | 2.2688 | 4.0841 | 4.8589 | 2.6796 | 4.9347 | 3.1223 | 4.0774 | 3.8412 | 3.4407 | 3.0457 | 2.4119 | 4.3601 | 3.5879 | 4.0061 |

**Table 15** Weights and biases of the best MLP model for predicting pillow effect (Trainbfg with Tansig)—one-output structure

| $b_1$ | $b_2$ | $IW$ | | | | | $LW$ |
|---|---|---|---|---|---|---|---|
| − 1.0689 | − 4.4799 | − 0.3664 | − 2.9456 | − 0.0801 | − 0.0311 | 0.1396 | 4.8997 |
| 2.5502 | | 3.4493 | 1.8466 | 2.2851 | 3.3176 | 0.4012 | − 2.9949 |
| 4.8340 | | − 0.3515 | − 1.7678 | 0.6648 | − 2.8060 | 2.3888 | − 1.8258 |
| 5.3937 | | 2.7239 | − 0.8303 | 4.9030 | 2.1082 | − 0.2148 | − 4.4692 |
| 0.5004 | | 0.7225 | 4.7506 | 0.1742 | − 1.2817 | − 0.1876 | 2.3491 |
| 6.4855 | | − 1.4367 | 6.6591 | − 1.2504 | − 3.6789 | − 0.2164 | − 0.0459 |
| 1.7566 | | 9.6762 | − 4.1411 | 11.9422 | 2.6040 | 4.8982 | 0.9892 |
| 2.0245 | | − 3.1508 | 6.9449 | − 0.5326 | − 8.7422 | − 5.5040 | − 10.9224 |
| − 1.1407 | | 3.2435 | − 7.4108 | 0.4992 | − 3.7858 | 6.1299 | − 11.2295 |
| 6.9462 | | − 14.1984 | 8.0983 | 3.6904 | 1.1147 | − 9.5227 | − 15.6383 |

**Table 16** Weights and biases of the best MLP model for predicting wall diameter (Trainlm with Softmax) – one-output structure

| $b_1$ | $b_2$ | $IW$ | | | | | $LW$ |
|---|---|---|---|---|---|---|---|
| − 2.1118 | 19.2151 | − 0.2097 | 0.2117 | − 1.9420 | − 0.6388 | − 1.9055 | 2.5135 |
| 1.0873 | | 11.7171 | 2.9757 | − 77.4397 | − 144.0557 | 7.5593 | 22.9679 |
| − 5.0721 | | 1.5851 | − 8.5864 | 24.2144 | 44.0436 | 6.3161 | 24.9085 |
| − 4.6389 | | − 0.3582 | 3.2977 | 25.8808 | 112.4779 | − 1.9865 | 23.8584 |
| − 3.1617 | | 0.8478 | − 5.0426 | 23.6070 | 72.4671 | 2.9167 | − 135.9051 |
| 2.2753 | | 0.5419 | − 5.9637 | 22.0928 | 73.4602 | 3.1807 | 29.3786 |
| 58.4812 | | − 21.3230 | 73.9425 | − 85.3340 | − 254.2707 | − 34.9560 | 25.9349 |
| − 22.5044 | | − 0.3622 | − 17.8875 | 32.3866 | 49.4518 | 7.7291 | 16.5500 |
| − 11.0040 | | 5.2457 | − 7.9919 | 18.9395 | 32.7593 | 5.1248 | 29.1113 |
| − 13.1130 | | 4.8888 | − 35.0915 | 20.2970 | 13.5966 | 7.2609 | − 19.6351 |

## References

A. C. M. & Guido, S. (1997). *Introduction to with python learning machine*.

Afzal, M. J. (2021). Study on the single point incremental sheet forming of AISI 321 variable wall angle geometry. https://doi.org/10.21203/rs.3.rs-836822/v1.

Al-Ghamdi, K., & Hussain, G. (2015). The pillowing tendency of materials in single-point incremental forming: Experimental and finite element analyses. *Journal of Engineering Manufacture, 229*(5), 744–753. https://doi.org/10.1177/0954405414530906

Amala Justus Selvam, M., Velu, R., & Dheerankumar, T. (2017). Study of the influence of the process variables on formability and strain distribution in incremental sheet metal working of AA 1050 Sheets, pp. 493–505.

Ambrogio, G., Cozza, V., Filice, L., & Micari, F. (2007). An analytical model for improving precision in single point incremental forming. *Journal of Materials Processing Technology, 191*(1–3), 92–95. https://doi.org/10.1016/j.jmatprotec.2007.03.079

Amirjan, M., Khorsand, H., Siadati, M. H., & Eslami Farsani, R. (2013). Artificial neural network prediction of Cu–$Al_2O_3$ composite properties prepared by powder metallurgy method. *The Journal of Materials Research and Technology, 2*(4), 351–355. https://doi.org/10.1016/j.jmrt.2013.08.001

Azaouzi, M., & Lebaal, N. (2012). Tool path optimization for single point incremental sheet forming using response surface method. *Simulation Modelling Practice and Theory, 24*, 49–58. https://doi.org/10.1016/j.simpat.2012.01.008

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE, 10*(7), e0130140. https://doi.org/10.1371/journal.pone.0130140

Bai, L., Li, Y., Yang, M., Lin, Y., Yuan, Q., & Zhao, R. (2019). Modeling and analysis of single point incremental forming force with static pressure support and ultrasonic vibration. *Materials (basel), 12*(12), 1899. https://doi.org/10.3390/ma12121899

Beale, M. H., Hagan, M., & Demuth, H. (2019). Deep Learning Toolbox Getting Started Guide. *Deep Learn. Toolbox*. https://doi.org/10.1016/j.neunet.2005.10.002

Beale, M. H., Hagan, M. T., & Demuth, H. B. (2013). Neural network toolbox$^{TM}$ user's guide R2013b. *Mathworks Inc*.

Beale, M. H., Hagan, M. T., & Demuth, H. B. (2020) Deep learning toolbox$^{TM}$ user's guide how to contact MathWorks.

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford University Press Inc.

Chen, M., *et al.* (2021). Task-wise split gradient boosting trees for multi-center diabetes prediction. In Proc. ACM SIGKDD int. conf. knowl. discov. data min., pp. 2663–2673. https://doi.org/10.1145/3447548.3467123.

Davis, J. R. (2001). Aluminum and aluminum alloys. *ASM International, 42*(351–416), 2001. https://doi.org/10.1136/oem.42.11.746

Demuth, H. (2000). *Neural network toolbox$^{TM}$6 user's guide*, vol. 9, no. 4.

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. *International Journal of Molecular Science, 20*(23), 6104. https://doi.org/10.3390/ijms20236104

Dorogush, A. V., Ershov, V., & Gulin, A. (2018). CatBoost: Gradient boosting with categorical features support. pp. 1–7. http://arxiv.org/abs/1810.11363.

Edwards, W. L., Grimm, T. J., Ragai, I., & Roth, J. T. (2017). Optimum process parameters for Springback reduction of single point incrementally formed polycarbonate. *Procedia Manufacturing, 10*, 329–338. https://doi.org/10.1016/j.promfg.2017.07.002

Essa, K., & Hartley, P. (2011). An assessment of various process strategies for improving precision in single point incremental forming. *International Journal of Material Forming, 4*(4), 401–412. https://doi.org/10.1007/s12289-010-1004-9

Ezugwu, E. O., Fadare, D. A., Bonney, J., Da Silva, R. B., & Sales, W. F. (2005). Modelling the correlation between cutting and process parameters in high-speed machining of Inconel 718 alloy using an artificial neural network. *International Journal of Machine Tools and Manufacture, 45*(12–13), 1375–1385. https://doi.org/10.1016/j.ijmachtools.2005.02.004

Garson, G. D. (1991). Interpreting neural-network connection weights. *AI Expert, 6*(4), 46–51.

Gatea, S., Ou, H., & McCartney, G. (2016). Review on the influence of process parameters in incremental sheet forming. *International Journal of Advanced Manufacturing Technology, 87*(1–4), 479–499. https://doi.org/10.1007/s00170-016-8426-6

Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesús, O. (2014). *Neural network design*. Martin Hagan.

Hartmann, C., Opritescu, D., & Volk, W. (2019). An artificial neural network approach for tool path generation in incremental sheet metal free-forming. *Journal of Intelligent Manufacturing, 30*(2), 757–770. https://doi.org/10.1007/s10845-016-1279-x

Hirsch, J. (2006). *AlMn1Mg1 for beverage cans*. Weinheim Wiley-VCH Verlag.

Hussaini, S. M., Singh, S. K., & Gupta, A. K. (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

Ibragimov, B., & Gusev, G. (2019). Minimal variance sampling in stochastic gradient boosting. *Advances in Neural Information Processing Systems, 32*, 15061–15071.

Ibrahim, O. M. (2013). A comparison of methods for assessing the relative importance of input variables in artificial neural networks. *Journal of Applied Sciences Research, 9*(11), 5692–5700.

Isidore, B. B. L. (2014). Controlling pillow defect in single point incremental forming through varying tool geometry

Isidore, B. B. L., Hussain, G., Shamchi, S. P., & Khan, W. A. (2016). Prediction and control of pillow defect in single point incremental forming using numerical simulations. *Journal of Mechanical Science and Technology, 30*(5), 2151–2161. https://doi.org/10.1007/s12206-016-0422-0

Jackson, K., & Allwood, J. (2009). The mechanics of incremental sheet forming. *Journal of Materials Processing Technology, 209*(3), 1158–1174. https://doi.org/10.1016/j.jmatprotec.2008.03.025

Kaufman, J. G. (2000). Applications for aluminum alloys and tempers. *Introduction to Aluminum Alloys and Their Tempers, 1100*, 242.

Khan, M. S., Coenen, F., Dixon, C., El-Salhi, S., Penalva, M., & Rivero, A. (2015). An intelligent process model: Predicting springback in single point incremental forming. *International Journal of Advanced Manufacturing Technology, 76*(9–12), 2071–2082. https://doi.org/10.1007/s00170-014-6431-1

Kiridena, V. S., Xia, Z. C., & Ren, F. (2016). High stiffness and high access forming tool for incremental sheet forming, United States, Patent Application Publication [US8021501B2]. US8021501B2, vol. 1, no. 19.

Kondayya, D., & Gopala Krishna, A. (2013). An integrated evolutionary approach for modelling and optimization of laser beam cutting process. *International Journal of Advanced Manufacturing Technology, 65*(1–4), 259–274. https://doi.org/10.1007/s00170-012-4165-5

Kumar, A., & Gulati, V. (2020). Optimization and investigation of process parameters in single point incremental forming. *Indian Journal of Engineering and Materials Science, 27*(2), 246–255.

Kumar, Y., & Kumar, S. (2015). Incremental sheet forming (ISF), pp. 29–46.

Lela, B., Bajić, D., & Jozić, S. (2009). Regression analysis, support vector machines, and Bayesian neural network approaches to modeling surface roughness in face milling. *International Journal of Advanced Manufacturing Technology, 42*(11–12), 1082–1088. https://doi.org/10.1007/s00170-008-1678-z

Leszak, E. (1967). Apparatus and process for incremental dieless forming, [US3342051A].

Li, E. (2013). Reduction of Springback by intelligent sampling-based LSSVR metamodel-based optimization. *International Journal of Material Forming, 6*(1), 103–114. https://doi.org/10.1007/s12289-011-1076-1

Low, D. W. W., Chaudhari, A., Kumar, D., & Kumar, A. S. (2022). Convolutional neural networks for prediction of geometrical errors in incremental sheet metal forming. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-022-01932-1

Lu, H. B., Le Li, Y., Liu, Z. B., Liu, S., & Meehan, P. A. (2014). Study on step depth for part accuracy improvement in incremental sheet forming process. *Advances in Materials Research, 939*, 274–280. https://doi.org/10.4028/www.scientific.net/AMR.939.274

Lundberg, S. M., Erion, G. G., & Lee, S.-I. (2018). Consistent individualized feature attribution for tree ensembles, no. 2, 2018. http://arxiv.org/abs/1802.03888.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, vol. 30. https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

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. *International Journal of Material Forming, 5*(2), 147–155. https://doi.org/10.1007/s12289-011-1035-x

Matsubara, S. (1994). Incremental backward bulge forming of a sheet metal with a hemispherical head tool-a study of a numerical control forming system II. *Journal of Japan Society for Technology of Plasticity, 35*(406), 1311–1316.

Mease, D., & Wyner, A. (2008). Evidence contrary to the statistical view of boosting. *Journal of Machine Learning Research, 9*, 131–156.

Mezher, M. T., Barrak, O. S., Nama, S. A., & Shakir, R. A. (2021a). Predication of forming limit diagram and spring-back during SPIF process of AA1050 and DC04 sheet metals. *Journal of Mechanical Engineering Research and Developments, 44*(1), 337–345.

Mezher, M. T., Khazaal, S. M., Namer, N. S. M., & Shakir, R. A. (2021b). A comparative analysis study of hole flanging by incremental sheet forming process of AA1060 and DC01 sheet metals. *Journal of Engineering Science and Technology, 16*(6), 4383–4403.

Mezher, M. T., Namer, N. S. M., & Nama, S. A. (2018). Numerical and experimental investigation of using lubricant with nano powder additives in SPIF process. *International Journal of Mechanical Engineering and Technology, 9*(13), 968–977.

Micari, F., Ambrogio, G., & Filice, L. (2007). Shape and dimensional accuracy in single point incremental forming: State of the art and future trends. *Journal of Materials Processing Technology, 191*(1–3), 390–395. https://doi.org/10.1016/j.jmatprotec.2007.03.066

Misra, S., & He, J. (2020). Stacked neural network architecture to model the multifrequency conductivity/permittivity responses of subsurface shale formations. In *Machine learning for subsurface characterization* (pp. 103–127). Elsevier.

Mulay, A., Ben, B. S., 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

Nabipour, M., & Keshavarz, P. (2017). Modeling surface tension of pure refrigerants using feed-forward back-propagation neural networks. *International Journal of Refrigeration, 75*, 217–227. https://doi.org/10.1016/j.ijrefrig.2016.12.011

Nagargoje, A., Kankar, P. K., Jain, P. K., & Tandon, P. (2021). Application of artificial intelligence techniques in incremental forming: A state-of-the-art review. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-021-01868-y

Najm, S. M., & Paniti, I. (2018). Experimental investigation on the single point incremental forming of AlMn1Mg1 foils using flat end tools. In *IOP conf. ser. mater. sci. eng.*, vol. 448, p. 012032. https://doi.org/10.1088/1757-899X/448/1/012032

Najm, S. M., & Paniti, I. (2020). Study on Effecting Parameters of Flat and Hemispherical end Tools in SPIF of Aluminium Foils. *Tehnički vjesnik – Technical Gazette*. https://doi.org/10.17559/TV-20190513181910.

Najm, S. M., & Paniti, I. (2021a). 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. *International Journal of Advanced Manufacturing Technology, 114*(9–10), 2591–2615. https://doi.org/10.1007/s00170-021-06712-4

Najm, S. M., & Paniti, I. (2021b). Predict the effects of forming tool characteristics on surface roughness of aluminum foil components formed by SPIF using ANN and SVR. *International Journal of Precision Engineering and Manufacturing, 22*(1), 13–26. https://doi.org/10.1007/s12541-020-00434-5

Najm, S. M., Paniti, I., Trzepieciński, T., Nama, S. A., Viharos, Z. J., & Jacso, A. (2021). Parametric effects of single point incremental forming on hardness of AA1100 aluminium alloy sheets. *Materials (basel), 14*(23), 7263. https://doi.org/10.3390/ma14237263

Najm, S. M., Paniti, I., & Viharos, Z. J. (2020). Lubricants and affecting parameters on hardness in SPIF of AA1100 aluminium. In *17th IMEKO TC 10 EUROLAB Virtual Conf. "global trends testing, diagnostics insp. 2030,"* pp. 387–392.

Nasulea, D., & Oancea, G. (2021). Achieving accuracy improvements for single-point incremental forming process using a circumferential hammering tool. *Metals (basel), 11*(3), 482. https://doi.org/10.3390/met11030482

Olden, J. D., & Jackson, D. A. (2002). Illuminating the 'black box': A randomization approach for understanding variable contributions in artificial neural networks. *Ecol. Modell., 154*(1–2), 135–150. https://doi.org/10.1016/S0304-3800(02)00064-9

Paniti, I., Viharos, Z. J., Harangozó, D., & Najm, S. M. (2020). Experimental and numerical investigation of the single-point incremental forming of aluminium alloy foils. *Acta IMEKO, 9*(1), 25–31.

Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in {P}ython". *The Journal of Machine Learning Research, 12*, 2825–2830.

Principe, J., Euliano, N. R., & Lefebvre, W. C. (1997). Neural and adaptive systems: Fundamentals through simulation: multilayer perceptrons. In *Neural adapt. syst. fundam. through simulation©*, pp. 1–108. https://doi.org/10.1002/ejoc.201200111.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., & Gulin, A. (2018). Catboost: Unbiased boosting with categorical features. *Advances in Neural Information Processing Systems., 208*(4), 6638–6648.

Rezakazemi, M., Razavi, S., Mohammadi, T., & Nazari, A. G. (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. *Journal of Membrane Science, 379*(1–2), 224–232. https://doi.org/10.1016/j.memsci.2011.05.070

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Why should I trust you?. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135–1144. https://doi.org/10.1145/2939672.2939778.

Riedmiller, P. M. *Machine learning: Multi layer perceptrons*. Albert-Ludwigs-University Freibg. AG Maschinelles Lernen. Retrieved from http://ml.informatik.uni-freiburg.de/_media/documents/teaching/ss09/ml/mlps.pdf.

Shabanzadeh, P., Yusof, R., & Shameli, K. (2015). Artificial neural network for modeling the size of silver nanoparticles' prepared in montmorillonite/starch bionanocomposites. *Journal of Industrial and Engineering Chemistry, 24*, 42–50. https://doi.org/10.1016/j.jiec.2014.09.007

Shahin, M., Maier, H. R., & Jaksa, M. B. (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.

Skjoedt, M., Hancock, M. H., & Bay, N. (2007). Creating helical tool paths for single point incremental forming. *Key Engineering Materials, 344*, 583–590. https://doi.org/10.4028/www.scientific.net/KEM.344.583

Štrumbelj, E., & Kononenko, I. (2014). Explaining prediction models and individual predictions with feature contributions. *Knowledge and Information Systems, 41*(3), 647–665. https://doi.org/10.1007/s10115-013-0679-x

Szpunar, M., Ostrowski, R., Trzepieciński, T., & Kaščák, Ľ. (2021). Central composite design optimisation in single point incremental forming of truncated cones from commercially pure titanium grade 2 sheet metals. *Materials (basel), 14*(13), 3634. https://doi.org/10.3390/ma14133634

Trzepieciński, T., Kubit, A., Dzierwa, A., Krasowski, B., & Jurczak, W. (2021c). Surface finish analysis in single point incremental sheet forming of rib-stiffened 2024–T3 and 7075–T6 alclad aluminium alloy panels. *Materials (basel), 14*(7), 1640. https://doi.org/10.3390/ma14071640

Trzepieciński, T., Najm, S. M., Oleksik, V., Vasilca, D., Paniti, I., & Szpunar, M. (2022a). Recent developments and future challenges in incremental sheet forming of aluminium and aluminium alloy sheets. *Metals (basel), 12*(1), 124. https://doi.org/10.3390/met12010124

Trzepieciński, T., Najm, S. M., Sbayti, M., Belhadjsalah, H., Szpunar, M., & Lemu, H. G. (2021b). New advances and future possibilities in forming technology of hybrid metal-polymer composites used in aerospace applications. *Journal of Composites Science, 5*(8), 217. https://doi.org/10.3390/jcs5080217

Trzepieciński, T., Oleksik, V., Pepelnjak, T., Najm, S. M., Paniti, I., & Maji, K. (2021a). Emerging trends in single point incremental sheet forming of lightweight metals. *Metals (basel), 11*(8), 1188. https://doi.org/10.3390/met11081188

Trzepieciński, T., Szpunar, M., & Ostrowski, R. (2022b). Split-plot I-optimal design optimisation of combined oil-based and friction stir rotation-assisted heating in SPIF of Ti-6Al-4V titanium alloy sheet under variable oil pressure. *Metals (basel), 12*(1), 113. https://doi.org/10.3390/met12010113

Vatankhah, E., Semnani, D., Prabhakaran, M. P., Tadayon, M., Razavi, S., & Ramakrishna, S. (2014). Artificial neural network for modeling the elastic modulus of electrospun polycaprolactone/gelatin scaffolds. *Acta Biomaterialia, 10*(2), 709–721. https://doi.org/10.1016/j.actbio.2013.09.015

Wei, H., Hussain, G., Shi, X., Isidore, B. B. L., Alkahtani, M., & Abidi, M. H. (2020). Formability of materials with small tools in incremental forming. *Chinese Journal of Mechanical Engineering, 33*(1), 55. https://doi.org/10.1186/s10033-020-00474-y

Yu, H., & Wilamowski, B. M. (2009a). C++ implementation of neural networks trainer. In *Proc. - 2009a int. conf. intell. eng. syst. INES 2009a*, pp. 257–262. https://doi.org/10.1109/INES.2009.4924772.

Yu, H., & Wilamowski, B. M. (2009b) Efficient and reliable training of neural networks. In *Proc. - 2009b 2nd conf. hum. syst. interact. HSI '09*, pp. 109–115. https://doi.org/10.1109/HSI.2009.5090963.

Zain, A. M., Haron, H., & Sharif, S. (2010). Prediction of surface roughness in the end milling machining using Artificial Neural Network. *Expert Systems with Applications, 37*(2), 1755–1768. https://doi.org/10.1016/j.eswa.2009.07.033

Zarei, M. J., Ansari, H. R., Keshavarz, P., & Zerafat, M. M. (2020). Prediction of pool boiling heat transfer coefficient for various nano-refrigerants utilizing artificial neural networks. *Journal of Thermal Analysis and Calorimetry, 139*(6), 3757–3768. https://doi.org/10.1007/s10973-019-08746-z

Zhang, Z., et al. (2016). Springback reduction by annealing for incremental sheet forming. *Procedia Manuf., 5*, 696–706. https://doi.org/10.1016/j.promfg.2016.08.057

Zhang, G., Eddy-Patuwo, B., & Hu, M. Y. (1998). Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting, 14*(1), 35–62.

Zhang, H., Lu, B., Chen, J., Feng, S., Li, Z., & Long, H. (2017). Thickness control in a new flexible hybrid incremental sheet forming process. *Journal of Engineering Manufacture, 231*(5), 779–791. https://doi.org/10.1177/0954405417694061

Zhou, B., et al. (2015). Relative importance analysis of a refined multi-parameter phosphorus index employed in a strongly agriculturally influenced watershed. *Water, Air, and Soil Pollution, 226*(3), 25. https://doi.org/10.1007/s11270-014-2218-0

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

## Author and Affiliations

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

Imre Paniti
imre.paniti@sztaki.hu

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

[2] Kirkuk Technical Institute, Northern Technical University,
36001 Kirkuk, Iraq

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

[4] Research Center of Vehicle Industry, Széchenyi István
University, Győr, Hungary