DOI: 10.1007/s12541-016-0185-7 ISSN 2234-7593 (Print) / ISSN 2005-4602 (Online)

Application of an RBF Neural Network for FDM Parts' Surface Roughness Prediction for Enhancing Surface Quality

Ebrahim Vahabli[1] and **Sadegh Rahmati**[1,#]

1 Department of Mechanical and Aerospace Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran
# Corresponding Author / E-mail: Srahmati@srbiau.ac.ir, TEL: +98-912-1350938, FAX: +98-21-44865151

KEYWORDS: Surface roughness, Fused deposition modeling, Neural network, Radial basis function, Imperialist competitive algorithm, Sensitivity analysis

To improve the surface roughness of parts fabricated using fused deposition modeling, modeling of the surface roughness distribution is used before the fabrication process to enable more precise planning of the additive manufacturing process. In this paper, a new methodology based on radial basis function neural networks (RBFNNs) is proposed for estimation of the surface roughness based on experimental results. The effective variables of the RBFNN are optimized using the imperialist competitive algorithm (ICA). The RBFNN-ICA model outperforms considerably comparing to the RBFNN model. A specific test part capable of evaluating the surface roughness distribution for varied surface build angles is built. To demonstrate the advantage of the recommended model, a performance comparison of the most well-known analytical models is carried out. The results of the evaluation confirm the capability of more fitted responses in the proposed modeling. The RBFNN and RBFNN-ICA models have mean absolute percentage error of 7.11% and 3.64%, respectively, and mean squared error of 7.48 and 2.27, respectively. The robustness of the network is studied based on the RBFNN's effective variables evaluation and sensitivity analysis assessment for the contribution of input parameters. Finally, the comprehensive validity assessments confirm improved results using the recommended modeling.

Manuscript received: October 12, 2015 / Revised: April 29, 2016 / Accepted: July 25, 2016

1. Introduction

Additive manufacturing (AM) is one of the most promising and widely used technologies today, and it is used to reduce product development time because the prototype component produced can be used directly in assemblies, product testing or tooling for short or medium run production. AM relates to a rapidly growing number of automated machines or processes such as stereolithography (SL), selective laser sintering (SLS), fused deposition modeling (FDM), laminated object manufacturing (LOM) and shape deposition manufacturing (SDM) which can automatically fabricate three dimensional (3D) solid models from the computer-aided design (CAD) data without use of tooling or human intervention.[1-7]

FDM is one of the AM processes that build parts of any geometry by sequential deposition of material on a layer-by-layer basis. The process uses heated thermoplastic filaments which are extruded from the tip of the nozzle in a prescribed manner, in a semi-molten state, and which solidify at chamber temperature. The part is fabricated by the two-dimensional deposition of layers contoured in the x-y plane. The third dimension (z) results from the stacking of single layers on top of each other but not as a continuous z-coordinate.[8]

The surface finish of the completed part is excessively rough as a consequence of the layered production method and this is unavoidable. This is a problem that occurs widely in different industries that use the FDM process. Poor surface finishing in AM processes is often affected by the tessellation of the original CAD model and the slicing procedure employed during the building process. Slicing a tessellated CAD model causes a containment problem that results in the distortion of the original CAD model of the part as designed. In addition to the containment problem, deposition of the layers in slices results in another issue known as the 'staircase effect' or the 'stair-stepping effect'.[9]

A critical review of the literature suggests that the properties of AM parts are the function of various process related parameters and can be significantly improved with proper adjustment without incurring additional expenses for changing hardware and software.[10] In the context of optimization of the AM process or process parameter selection for

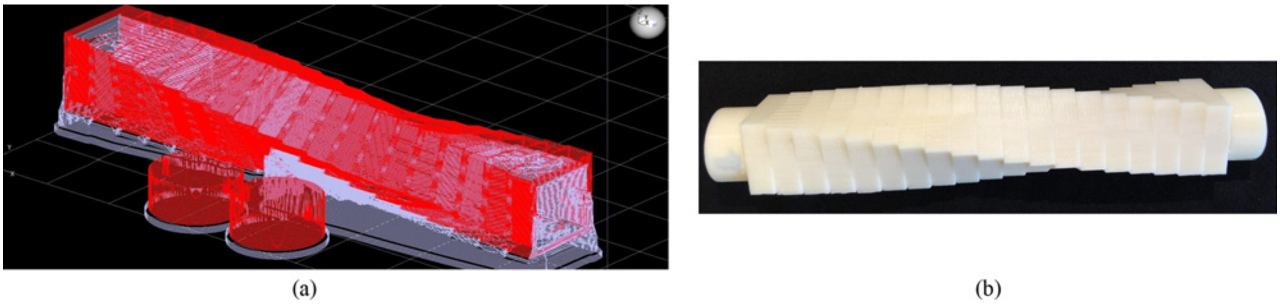

Fig. 1 (a) Isolated view of the test part in CatalystEx, (b) The final test part after washing and assembly

the improvement of performance characteristics of the prototype, the formulation of mathematical models based on a physical understanding of the AM processes plays a major role. Researchers have made several attempts to improve the performance of prototypes via formulation of theoretical models followed by optimization.[11-15] Therefore, surface roughness estimation models may be replaced by trial and error methods to save money and time. Some studies, which focus directly on surface roughness, can be divided into empirical and theoretical methods. Among the empirical methods, Armillotta[16] describes an experimental investigation on the effects of process parameters such as layer thickness, build orientation, road width and raster angle on surface roughness value. Pandey et al.[17,18] have developed a semi-empirical model of the arithmetic-mean-surface roughness (Ra) which considers layer thickness and build orientation as process variables; they approximate the layer profile with a parabola assuming the mean surface to be in the middle. Campbell et al.[19] verified the empirical Ra value proposed by Reeves and Cobb[20] to identify a range of surface angles in which Ra can be reasonably predicted; for FDM they attest that most part surfaces displayed an experimental Ra that was much lower than the one calculated. Ahn et al.[11,21] introduced an equation to express Ra distribution in terms of surface angle using measured Ra data and interpolation. Among the theoretical works, Ahn et al.[11,21] presented a model to express average roughness by assuming that the filament profile was an elliptical curve; cross-sectional shape, surface angle, layer thickness and overlap between adjacent layers are considered to be the main factors affecting surface quality. Luis Perez et al.[22] characterized the geometric roughness arising from a layered manufacturing process and compared models using the SL manufacturing technique to establish two different rapid production strategies for prototypes based on the Ra value, manufacturing with an Ra within a given range of tolerances and production with a constant layer height. Bordoni and Boschetto[23] suggested that layer thickness and surface tilting were the main factors influencing the average roughness. Ahn et al.[24] and Thrimurthulu et al.[15] used these models to determine the optimum part deposition orientation for complex parts.

The model proposed in this study is able to estimate the surface roughness distribution in an FDM processed part with the appropriate process parameters. Accordingly, a reduction in surface roughness is expected because the process parameters are optimized.

Most studies that focused on the surface roughness improvement[9,13,15,18-20,25-27] were not established with empirical data using a comprehensive evaluation at the same time. So that the model estimates surface roughness at all build angle ranges (the main factor of the staircase effect that results in the creation of a rough surface), a specific test part is necessary. Therefore, a specific experimental part was manufactured so the surface roughness could be evaluated at various build angles. Meanwhile, an attempt was made to optimize the process variables of the RBFNN model using imperialist competitive algorithm (ICA). The most well-known analytical models were simulated and then the results were compared with those obtained using the proposed models. The RBFNN and RBFNN-ICA models were simulated for four test parts to demonstrate the improvement achieved for more specific responses in artificial neural network (ANN) modeling. A validity assessment showed that the proposed approach gave the closest estimation to the experimental data. The robustness of the network was studied considering the effective parameters of the RBFNN model and the contribution of the input parameters on output of the network. Altogether, all the evaluations confirm that better results were obtained using the recommended modeling.

2. Experimental Study

Empirical surface roughness (Ra) values were collected for four truncheon test parts made with different FDM machines as follows:

- Two test parts were designed and fabricated with Lt values of 0.254 and 0.3302 mm.
- The reported Ra values of other two test parts were derived from Campbell et al. (2002) with a layer thickness (Lt_c) of 0.253 mm, and Ahn et al. (2008) with a Lt_a of 0.254 mm [the updated values were reported by Ref. 28].

2.1 Test part design

Campbell et al. (2002) and Ahn et al. (2008) presented a more comprehensive test part for assessment of their analytical models. This test part, called a truncheon, was first introduced by Ref. 29. As shown in Fig. 1, the truncheon test part consists of several equivalent and parallel cuboids in which the build angle of each cuboid is increasing by a constant step angle with respect to a reference cuboid. Assuming an upward build direction, the most right cuboid of truncheon is selected as the reference cuboid, and it is designed so that the front side of the reference cuboid has a surface build angle of zero degrees. With this arrangement, as illustrated in Fig. 2(a), the front side of the truncheon part spans build angles between 0 to 90° from the most right

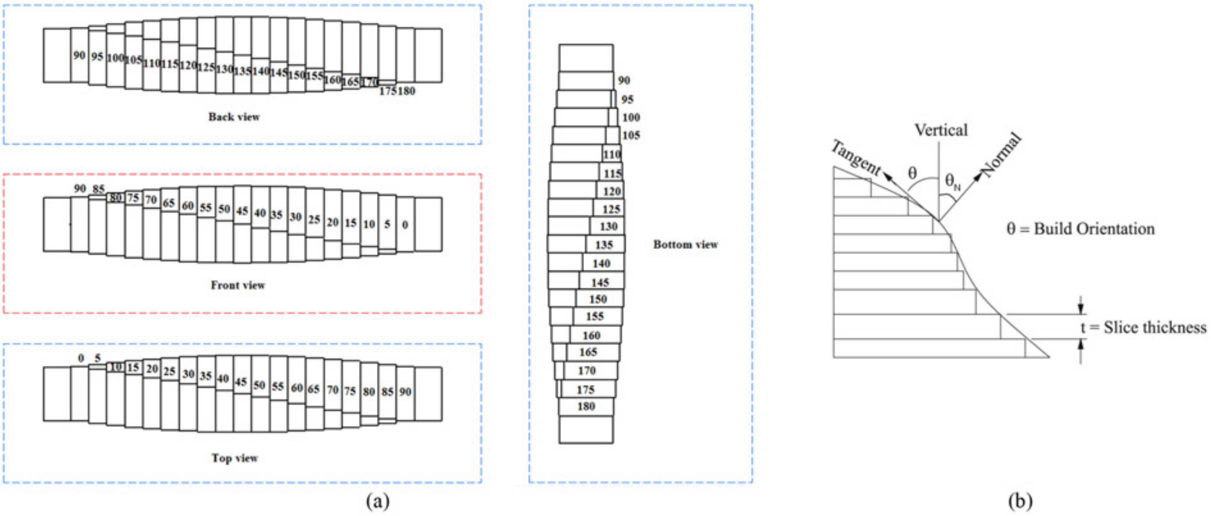

Fig. 2 (a) Front, back, top and bottom view of truncheon with corresponding build angles, (b) the general surface roughness model parameters[30]

to the most left side; the top side of the truncheon part spans build angles between 0 to 90° from the most left to the most right side; the bottom side spans build angles between 90 to 180° from the most right to the most left side; finally, the back side spans build angles between 90 to 180° from the most left to the most right side of the part. The general surface roughness model parameters are illustrated in Fig. 2(b), where θ is the build orientation (build angle), t the layer thickness and θ_N the angle between vertical axis and normal to surface axis. The truncheon part was fabricated considering the general surface roughness model parameters by regarding a constant layer thickness and varied build angles for each cuboid. This comprehensive design enables the complete evaluation of the surface roughness distribution in all build angle intervals which improves the defects of previews designs.

Details of the process parameters of the reported truncheon test part manufactured by an FDM apparatus is given in Ref. 19 which is as follows: Lt_c of 0.253 mm and an incremental surface build angle step of θ_{step} = 2 degrees (total cuboids: 90). In work by Ahn et al.,[26] details of the process parameters of the fabricated part are described as follows: Lt_a of 0.254 mm and an incremental surface build angle step of θ_{step} = 3 degrees (total cuboids: 60). In the present study, an incremental surface build angle (θ_{step} = 5 degrees) was selected and the test part was designed in CATIA^{TM} v5. For accurate measurement using a surface scanner and a machine room size of 254 × 254 × 305 mm^3, the dimensions of each cuboid were set to 10 × 30 × 30 mm^3, and the dimensions of the entire truncheon are 220 × 30 × 30 mm^3.

2.2 Test part fabrication and measurement process

The prototype was manufactured using a Dimension^{TM} sst 1200es 3D printer (Stratasys^{TM}), in which the test part was sliced using CatalystEx^{TM} software and acrylonitrile-butadiene-styrene (ABS*plus*) material was used for manufacturing. The cost of each AM part had a direct relationship with the building time of the part so the prototype was fabricated horizontally. The model interior was solid and a sparse support method was used for fabrication. The test part was solid and fabricated with a sparse support method to reduce the final cost as

Table 1 Empirical surface roughness (Ra) values for each build angle

Build angle (°)	Ra-Report No. Lt = 0.254 mm		Ra_{ave} (µm)	Ra-Report No. Lt = 0.3302 mm		Ra_{ave} (µm)
	Ra-A	Ra-B		Ra-A	Ra-B	
0	17.56	17.82	17.69	16.21	19.27	17.74
5	17.72	19.1	18.41	18.54	24.5	21.52
10	20.37	19.33	19.85	18.21	21.99	20.1
15	21.94	20.56	21.25	19.75	20.01	19.88
20	20.06	22.82	21.44	17.01	17.29	17.15
25	26.98	20.32	23.65	20.35	23.73	22.04
30	24.48	25.32	24.9	22.12	22.4	22.26
35	27.43	27.63	27.53	24.32	23.44	23.88
40	28.56	29.52	29.04	22.36	29.94	26.15
45	32.52	28.74	30.63	30.14	31.54	30.84
50	31.92	32.54	32.23	29.11	28.99	29.05
55	34.52	34.4	34.46	31.66	31.76	31.71
60	39.26	40.1	39.68	35.01	35.57	35.29
65	42.01	39.55	40.78	30.11	36.29	33.2
70	38.42	42.5	40.46	35.12	24.62	29.87
75	31.3	31.4	31.35	35.54	34.68	35.11
80	23.08	20.38	21.73	24.51	24.73	24.62
85	18.27	17.59	17.93	21.04	22.04	21.54
90	1.4	2.72	2.06	3.45	10.03	6.74
95	18.55	17.77	18.16	19.66	19.78	19.72
100	18.2	17.56	17.88	18.92	18.58	18.75
105	18.68	18.4	18.54	18.24	20.16	19.2
110	19.14	20.02	19.58	20.15	21.35	20.75
115	17.14	23.54	20.34	20.04	21.18	20.61
120	21.1	21.22	21.16	25.34	18.92	22.13
125	22.55	22.63	22.59	22.8	22.88	22.84
130	24.63	23.37	24	24.72	23.64	24.18
135	24.66	26.92	25.79	29.35	23.35	26.35
140	26.63	25.45	26.04	30.01	26.29	28.15
145	21.86	26.22	24.04	25.7	25.76	25.73
150	27.83	30.11	28.97	32.24	33.76	33
155	29.61	30.45	30.03	37.77	39.65	38.71
160	32.5	30.82	31.66	41.11	42.75	41.93
165	33.21	35.63	34.42	26.34	24.54	25.44
170	20.22	20.74	20.48	19.28	18.98	19.13
175	18.78	17.92	18.35	14.11	11.41	12.76
180	9.41	9.47	9.44	3.22	3.24	3.23

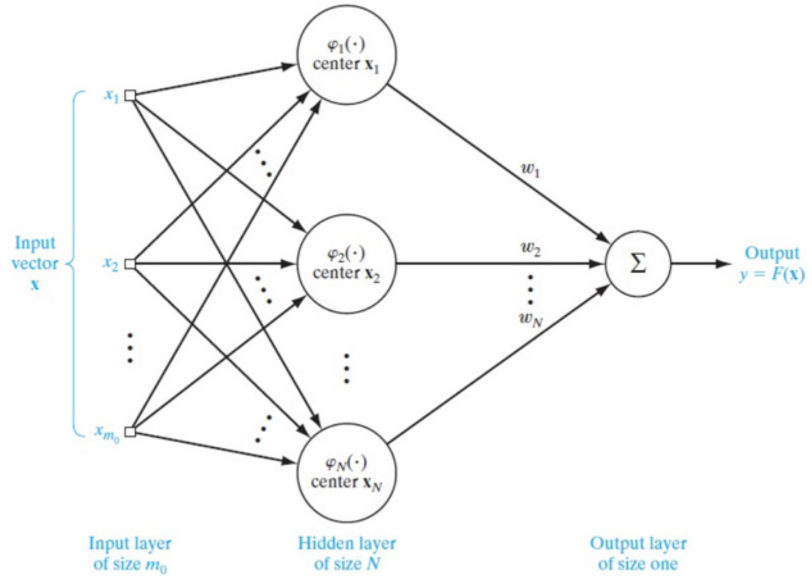
$\varphi_1(\cdot)$ center $\mathbf{x}_1$
$\varphi_2(\cdot)$ center $\mathbf{x}_2$
$\varphi_N(\cdot)$ center $\mathbf{x}_N$
x_1
x_2
x_{m_0}
Input vector $\mathbf{x}$
w_1
w_2
w_N
Σ
Output $y = F(\mathbf{x})$
Input layer of size m_0
Hidden layer of size N
Output layer of size one

Fig. 3 Structure of an RBFNN[31]

shown in Fig. 1(a) (the blue color denotes the sliced support and the red color denotes the sliced part). In the final stage, the test part was immersed in a chemical solution for one day to thoroughly dissolve the support material. In order that Ra may be assessed, no finishing operation was implemented, as shown in Fig. 1(b). The test part surface was measured using a MahrSurf$^{\text{TM}}$ MFW250 surface scanner for subsequent data analysis. The Ra values measured by the scanner for test parts with an Lt of 0.254 mm and 0.3302 mm are shown in Table 1. Two separate roughness measurements were made on the test parts on the same surface for each build angle. Ra$_{\text{ave}}$ denotes the average value between the Ra-A and Ra-B values, which was the Ra value used in the evaluation procedure.

3. Methodology

3.1 Artificial neural networks

A neural network is a massively parallel distributed processor made up of simple processing units that have a natural tendency for storing experiential knowledge and making it available for us.[31] ANNs can be grouped into two major categories: feedforward and feedback (recurrent) networks. In the former network, no loops are formed by the network connections, whereas one or more loops may exist in the latter. The most commonly used family of feedforward networks is a layered network in which neurons are organized into layers with connections strictly in one direction from one layer to another.[32]

3.1.1 Radial basis function (RBF)

RBF neural networks (RBFNN) were independently proposed by many researchers and are a popular alternative to the multi-layer perceptron (MLP). RBFNN are also good at modeling non-linear data and can be trained in one stage rather than using an iterative process as in MLP and also learn the given application quickly. The structure of the RBFNN is similar to that of MLP. It consists of a layer of neurons. The main distinction is that RBF has a hidden layer which contains nodes called RBF units. Each RBF has two key parameters that describe the location of the function's center and its deviation or width. The hidden unit measures the distance between an input data vector and the center of its RBF. The RBF has its peak when the distance between its center and that of the input data vector is zero and declines gradually as this distance increases. There is only a single hidden layer in a RBFNN and there are only two sets of weights, one connecting the hidden layer to the input layer and the other connecting the hidden layer to the output layer. Those weights connected to the input layer contain the parameters of the basis functions. The weights connecting the hidden layer to the output layer are used to form linear combinations of the activations of the basis functions (hidden units) to generate the network outputs. Because the hidden units are non-linear, the outputs of the hidden layer may be combined linearly and so processing is rapid.[33] The hidden neurons are processing units that perform the RBF. Each unit is mathematically defined as in Eq. (1):[34]

$$\varphi(\|x-x_j\|), \quad j = 1,2,\ldots,n \tag{1}$$

In Fig. 3, the neuron in the input layer just propagates input features to the next layer. Each neuron in the hidden layer is associated with a kernel function $\varphi_j(x)$, characterized by a center x_j and a width σ_j (spread). Each output neuron, y, computes a simple weighted summation over the responses of the hidden neurons for a given input pattern, x.

The j^{th} input data point x_j denotes the center of the RBF, and the vector x is the pattern applied to input layer. Selecting the basis function is not crucial to the performance of the network and the most common one used being the Gaussian basis function which is used in this study. It is defined in Eq. (2):[34]

$$\varphi_j(x) = \exp(-1/2\sigma_i^2(\|x-x_j\|^2)), \quad j = 1,2,\ldots,n \tag{2}$$

The output neuron is a summing unit to produce the output as a weighted sum of the hidden layer outputs as shown in Eq. (3):[34]

$$F(x) = \Sigma_{j=1}^{n} w_j \varphi_j(x) \tag{3}$$

where $\varphi_j(x)$ is the response of the j^{th} hidden node resulting from all input data, w_j is the connecting weight between the j^{th} hidden node and output node, and n is the number of hidden nodes. The center vectors, $\vec{x}$, the output weights w_j and the width parameter x_j are adjusted adaptively during the training of RBFNN in order for the data to fit well. The weights are optimized using least mean square (LMS) algorithm once the centers of the RBF units are determined. The centers can be chosen randomly or clustering algorithms can be used. In this study, centers were randomly selected from the data set.

In order to explain the above mentioned procedure in software platform, the total process of the RBFNN is described as below:

The proposed modeling was coded in MATLAB® (v.R2014b) software. The radial basis function neural network (RBFNN) was coded using *newrb* function, where it creates a two-layer network. The first layer has *radbas* neurons, and calculates its weighted inputs with *dist* and its net input with *netprod*. The *radbas* transfer function uses Gaussian basis function to find the optimum centers. Therefore, the proposed model applied the Gaussian function due to the *radbas* transfer function's default configuration.[31,35,36] The RBFNN conventionally uses the Gaussian function as the RBF. It is notable to indicate that the commonly suggested function in the approximation scope of RBFNN is Gaussian basis function which have been investigated and discussed in the literature.[31,35] The second layer has *purelin* neurons, and calculates its weighted input with *dotprod* and its net inputs with *netsum*. Both layers have biases. Initially the *radbas* layer has no neurons. The following steps are repeated until the network's mean squared error falls below desired goal: 1) The network is simulated. 2) The input vector with the greatest error is found. 3) A *radbas* neuron is added with weights equal to that vector. 4) The *purelin* layer weights are redesigned to minimize error.[36]

To use an RBFNN, an intelligent training algorithm is absolutely necessary for determining the network parameters. ICA was used to optimize the effective process variables accurately.

3.1.2 Imperialist Competitive Algorithm (ICA)

Imperialist competitive algorithm (ICA) is a new evolutionary algorithm in the evolutionary computation field based on human being's sociopolitical evolution that was proposed by Atashpaz-Gargari and Lucas (2007).[37] In ICA, the countries can be viewed as population individuals and basically divided into two groups based on their power, i.e., colonies and imperialists. Also, one empire is formed by one imperialist with its colonies. Furthermore, two operators called assimilation and revolution and one strategy called imperialistic competition are the main building blocks that employed in ICA. The implementation procedures are described as below:[37]

A) Initializing phase:

A.1. Like other optimization algorithms, ICA requires some initial population to be created. Each solution (i.e., country) that in form of an array can be defined via Eq. (4):[37]

$$Country = [P_1, P_2, P_3, \ldots, P_{N_{var}}] \tag{4}$$

where $P_i s$ represents different variables based on various socio-political characteristics (such as culture, language, and economical policy), and N_{var} denotes the total number of the characteristics (i.e., n-dimension of the problems) to be optimized. Corresponding to the word country in imperialist competitive algorithm is "chromosome" in genetic algorithm (GA) terminology.

A.2. Creating the cost function: In order to evaluate the cost of countries, the cost function can be defined via Eq. (5):[37]

$$Cost = f(Country) = f([P_1, P_2, P_3, \ldots, P_{N_{var}}]) \tag{5}$$

A.3. Initializing the empires: Based on cost values, a certain number of countries that have the lowest cost are selected as imperialist (N_{imp}) and the rest, known as colonies (N_{col}), are divided among these imperialists. Each imperialist and its allocated colonies form an empire. The power of an empire assigns the initial number of its colonies. Therefore the colonies should be divided among imperialists based on the imperialist's power. For this purpose the normalized cost of an imperialist is defined as follows:

$$C_n = c_n - max_i\{c_i\} \tag{6}$$

In the above equation, c_n is the cost of n-th imperialist and C_n is its normalized cost. Normally, two methods can be used to divide colonies among imperialist: (1) from the imperialists' point of view which based on the power of each imperialist; (2) from the colonies' point of view which based on the relationship with the imperialist (i.e., the colonies should be possessed by the imperialist according to the power). Both methods are given via Eqs. (7) and (8), respectively.[37]

$$P_n = \left| \frac{c_n}{\Sigma_{i=1}^{n_{imp}} c_i} \right| \tag{7}$$

$$N \cdot C_n = round\{P_n \cdot N_{col}\} \tag{8}$$

where P_n is the normalized power of each imperialist, N_{col} and N_{imp} represent the number of all colonies and imperialists, respectively, and $N.C_n$ is the initial number of colonies of n-th empire.

B) Moving phase:

B.1. Assimilation: The imperialist countries try to absorb their colonies toward themselves. For this purpose the assimilation policy is considered in ICA. In this process the colonies move toward their proper imperialist along different optimization axis. The direction of the movement is the vector from the colony to the imperialist in which the colony moves toward the imperialist by x units. The distance between the initial position of colony and imperialist is defined by d and the position of colony after movement is defined by a random parameter with uniform distribution (x). By considering the assimilation coefficient (β) as a number greater than one, x can be formulated as:

$$X \sim U(0, \beta \times d) \tag{9}$$

In assimilation policy it is not necessary to move colonies along the vector from colony to the imperialist and the movements in this direct path without any deviation just increase the exploitation ability of the algorithm. Also the lack of diversity often leads to stagnation, as the model finds itself trapped in local optima. Therefore, it is vital to increase the search area around the imperialist by adding a random amount of deviation to the direction of movement. This new direction defined by a deviation angle (θ) which is a random number with uniform distribution as follows:

$$\theta \sim U(-\gamma, \gamma) \tag{10}$$

where γ is a parameter that adjusts the deviation from the original direction. Nevertheless, the values of β and γ are arbitrary; in most of the implementations a value of about 2 for β and about π/4 (rad) for γ results in good convergence of countries to the global minimum.

B.2. Revolution: In addition to assimilation policy, the revolution policy is considered that is a sudden change in the position of countries. The revolution increases the exploration of the algorithm and prevents the early convergence of countries to local minimums.[37]

C) Exchanging phase: Based on the cost function, when the new position of a colony is better than that of the corresponding imperialist, the imperialist and the colony change their positions and the new location with lower cost becomes the imperialist.

D) Imperialistic competition phase:

D.1. The total power of an empire: It is influenced by the power of imperialist country and the colonies of an empire via Eq. (11):[37]

$$T \cdot C_n = Cost(imperialst_n) + \xi mean\{Cost(colony\ of\ empire_n)\} \tag{11}$$

where $T.C_n$ is the total cost of the n-th empire and ξ is a positive small number that is considered to be less than 1.

D.2. Imperialistic competition: all empires try to take the possession of the colonies of other empires and control them. This imperialistic competition gradually brings about a decrease in the power of weaker empires and an increase in the power of more powerful ones. This imperialistic challenge is approached by just selecting some (usually one) of the weakest colonies of the weakest empire and making a competition among all empires to possess colonies (or colony). The possession probability (P_p) of each empire is related to their total power in the competition process. Based on this theory, these colonies will not definitely be possessed by the most powerful empires, but these empires will be more likely to possess them. The normalized total cost of an empire is determined as follows:[37]

$$N \cdot T \cdot C_n = T \cdot C_n - max_i\{T \cdot C_i\} \tag{12}$$

where $N.T.C_n$ and $T.C_n$ are the total cost and the normalized total cost of n-th empire, respectively. Having the normalized total cost, the possession probability of each empire is obtained by the following equation:[37]

$$P_{P_n} = \left| \frac{N \cdot T \cdot C_n}{\Sigma_{i=1}^{N_{imp}} N \cdot T \cdot C_i} \right| \tag{13}$$

To distribute the mentioned colonies among empires vector P is formed as follows:

$$P = [P_{P_1}, P_{P_2}, P_{P_3}, \ldots, P_{P_{N_{imp}}}] \tag{14}$$

Then, vector R with random numbers and uniform distribution is created which has the same size as P as follows:[37]

$$R = [r_1, r_2, r_3, \ldots, r_{N_{imp}}] \tag{15}$$

After that, by subtracting R from P vector D is formed:

$$D = P - R = [D_1, D_2, D_3, \ldots, D_{N_{imp}}] \tag{16}$$

Referring to vector D, the mentioned colony (colonies) is controlled to an empire whose corresponding index in D is maximized.[37]

E) Eliminating phase: When an empire loses all its colonies (i.e., their colonies will be divided among other empires), it is assumed to be collapsed and will be eliminated.

F) Convergence phase: At the end, all the empires except the most powerful one will be collapsed and the colonies will be handled by this unique empire. It should be noted that imperialist and colonies have the same position and power in this stage.

Taking into account the key phases described above, the steps of implementing ICA can be summarized as follows:[37]

- Step 1: Defining the optimization problem.
- Step 2: Generating initial empires by pick some random points on the function.
- Step 3: Move the colonies towards imperialist states in different directions (i.e., assimilation).
- Step 4: Random changes occur in the characteristics of some countries (i.e., revolution).
- Step 5: Position exchange between a colony and imperialist.
- Step 6: Compute the total cost of all empires.
- Step 7: Use imperialistic competition and pick the weakest colony from the weakest empire.
- Step 8: Eliminate the powerless empires.
- Step 9: Check if maximum iteration is reached; go to Step 3 for new beginning. If a specified termination criterion is satisfied stop and return the best solution.

3.2 Selection of input and output parameters

In developing an RBFNN model prediction of surface roughness distribution in an FDM built part, one should consider the main factors involved in the creation of the rough surface. Based on former research, it was assumed that the variables of build orientation (angle), and layer thickness were the inputs, and the arithmetic mean surface roughness (Ra) was the output. The important reasons for selecting appropriate input parameters are:

(1) In an FDM process, five parameters (layer thickness, build angle (or build orientation), the raster angle, the raster width and the air gap) are known to be important to the creation of roughness. Previous studies have concluded that according to the evaluation of the most well-known models, layer thickness and the build angle have a significant effect on surface roughness and varying the other parameters slightly modifies the output.[11,17,19-21,25]

(2) Because of the specific way in which it functions, the ANN must be trained with empirical data derived from identical test parts under the same experimental conditions. Only two studies investigated identical test parts and different designs to express surface roughness estimation using analytical modeling.[19,26]

(3) To more effectively demonstrate the advancement achieved by the model proposed in this paper, the ANN model was established in the same way as the most frequently cited estimation models.[9,13,19,26,27,38]

3.3 Training and test data

Table 2 Selected parameters

RBFNN		
Transfer function	*Radbas-Purelin*	
Number of Layers	2	
Number of Weight Elements	341	
Maximum Neuron Number	74	
Data Division	Random	
Performance Goal	0	
Spread	1	
Optimized parameters of RBFNN by ICA		
Number of Weight Elements	297	
Maximum Neuron Number	55	
Spread	0.3913	
Data Division Determination	Train	80%
	Test	20%

Table 3 Parameters of ICA approach

Parameter	Value
Number of initial countries	80
Number of initial imperialists	20
Number of decades	40
Revolution rate	0.3
Assimilation Coefficient (β)	2
Assimilation Angle (γ)	0.02

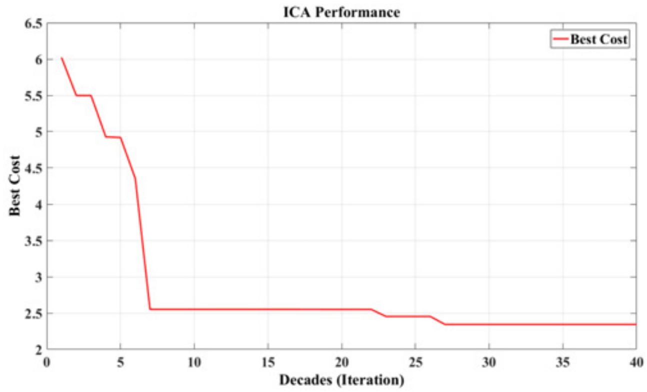

Fig. 4 Convergence of the ICA to the best cost

The research reported in this paper uses experimental data from two fabricated test parts (74 data sets) and two other test parts (32 data sets) from reported data. The input data were normalized in the interval of [0, 1]. The performance of the RBFNN is heavily affected by the centers and widths of the kernel functions in the hidden layer. The RBFNN was developed initially to model the surface roughness distribution using important process parameters which were studied in the experiments. The principle is that *newrb* creates neurons one at a time. At each iteration, the input vector that results in lowering the network error the most, is used to create a *radbas* neuron. The error of the new network is checked, and if it is low enough, *newrb* is finished. Otherwise the next neuron is added. This procedure is repeated until the error goal is met, or the maximum number of neurons is reached.[36]

The added neuron is a radially symmetric Gaussian with specified spread centered at the training vector that maximizes the correlation between the hidden nodes and the output layer targets.

Table 4 Performance of the RBFNN and RBFNN-ICA for the training, testing, and all data sets

Data set	Model	MSE	MAE	MAPE (%)	R
Training	RBFNN	2.31	1.12	6.30	0.9822
	RBFNN-ICA	0.28	0.27	2.62	0.9978
Testing	RBFNN	28.39	3.54	10.41	0.9036
	RBFNN-ICA	8.87	2.01	7.19	0.9325
All	RBFNN	7.48	1.60	7.11	0.9507
	RBFNN-ICA	2.27	0.67	3.64	0.9851

It was assumed that the desired approximation goal error is an MSE = 0 and the training process increases the number of hidden neurons from 1 to 74 (by steps equal to one). The network tries to converge to the predefined value (MSE = 0) without getting over-trained. Several trials have to be carried out to determine the process variables of the RBFNN properly. Therefore, an optimization method was implemented to apply the smart training algorithm for determining the network parameters. ICA optimizes the effective variables including connection weights, biases, maximum neuron numbers, spread, and the precise data division. For each network parameter considering the MSE, MAE, MAPE, and R values, the iteration number is greater than 40. The main process variables of RBFNN and the optimized values after ICA implementation are represented in Table 2. Considering the relevant literature to ICA,[37] it is extremely vital to assign the used parameters of ICA, accurately. Initializing of the ICA parameters is represented in Table 3.

Number of weight element indicates the number of weight and bias values in the network. It is the sum of the number of elements in the matrices stored in these three cell arrays of matrices which comprised of *IW*, *LW*, and *b*. *IW* that defines the weight matrices of weights going to layers from network inputs, *LW* which defines the weight matrices of weights going to layers from other layers, and *b* which denotes biases.[36] As depicted in Fig. 4, after 27 decades the value of the cost reaches to minimum cost and remains constant. Note that the mean squared error (MSE) of training data was implemented as a cost function in the ICA.

3.4 Training process

As shown in Figs. 5(a) to 5(d) and 6(a) to 6(d), prediction performance and correlation (R) plots of the RBFNN and RBFNN-ICA models for training and testing data demonstrates that the accuracy of the RBFNN model for desired outputs is in good accordance with the actual data. Meanwhile, the accuracy of the RBFNN-ICA model for train and test data performs better than the RBFNN model. Table 4 represents the proposed approaches prediction performance representing with different criteria. Regarding the results in Table 4, it is concluded that the accuracy of both the RBFNN and RBFNN-ICA models in training, testing, and all data sets is reasonably acceptable. However, the RBFNN-ICA model outperforms in training, testing, and all data sets due to the intelligent optimization of the process variables. Altogether, all assessments confirm the strength of the recommended methods in surface roughness prediction.

It is noteworthy to describe that the correlation coefficient (R value) determines the association among outputs and target values of the model - an R value of 1 and 0 indicates a strong and random association,

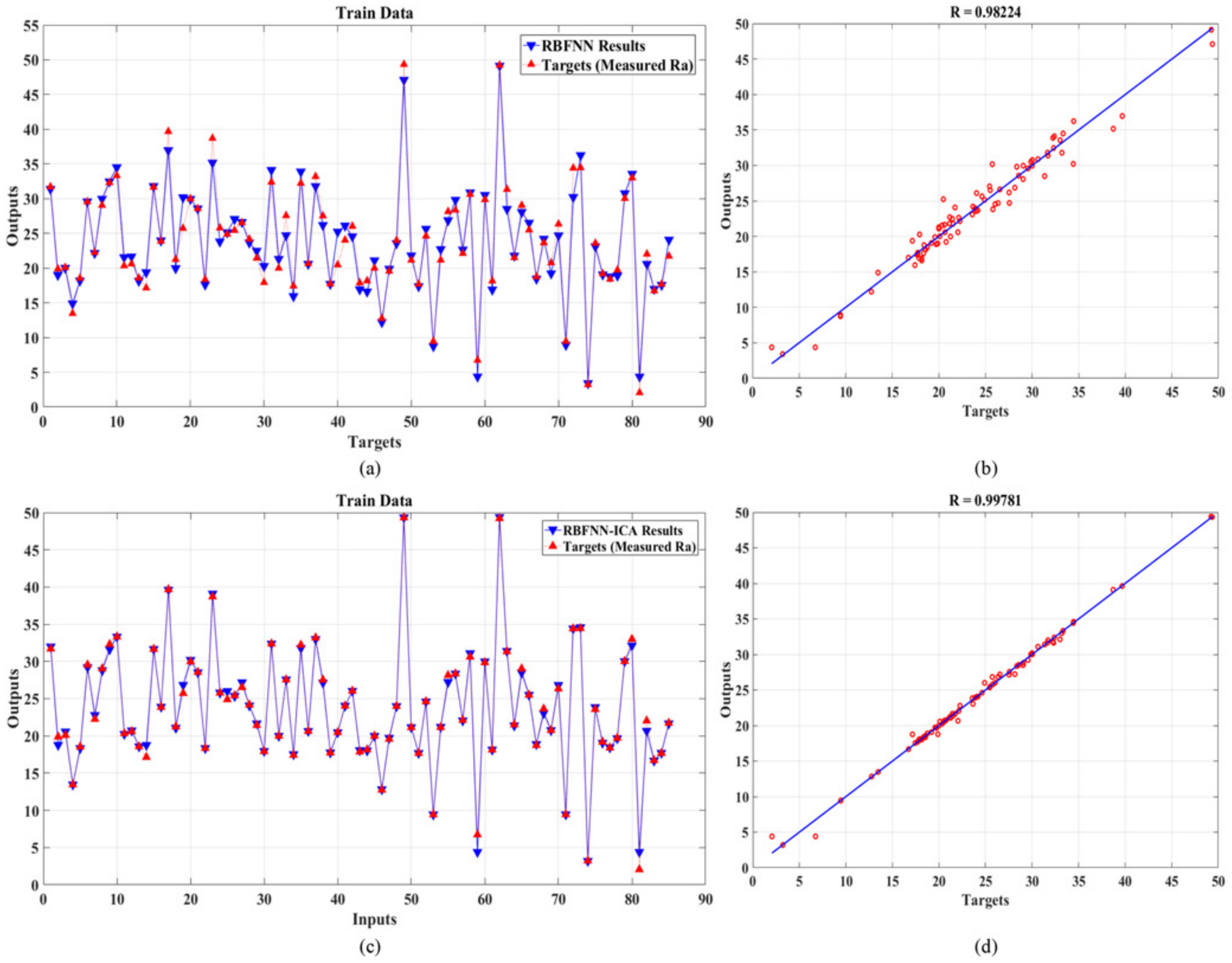

Fig. 5 the accuracy of the models for training data in surface roughness prediction, (a) the performance of the RBFNN, (b) the correlation of the RBFNN, (c) the performance of the RBFNN-ICA, (d) the correlation of the RBFNN-ICA

respectively. A perfect fit indicates that the data should fall along a line with a slope close to 1 (45 degrees), meaning that the network output tries to be converged to the desired target values which denotes the minimum prediction error. The R value was 0.9507 and 0.9851 for the RBFNN and RBFNN-ICA, respectively (all data set), indicating that each model can predict an Ra value close to the measured values.

3.4.1 Error criteria

The different criteria used in this study are: MSE, MAE, MAPE, R, and R^2.

The regression index (R) which denotes the correlation between outputs and targets. R^2 is the coefficient of determination which indicates the model accuracy and the MAPE which both of these criteria were used for network simulation assessments. The MAPE gives a normalized error, which allows an efficient comparison of nets derived from different data sets, as shown by Eq. (17):

$$MAPE(\%) = \frac{1}{N}\Sigma_i\left|\frac{t_i - o_i}{t_i}\right| \times 100 \tag{17}$$

The coefficient of determination, R^2 is calculated using Eq. (18):

$$R^2 = 1 - \left(\frac{\Sigma_i(t_i - o_i)^2}{\Sigma_i(o_i)^2}\right) \tag{18}$$

The MSE at the end of each epoch taken from all the patterns is calculated using Eq. (19):

$$MSE = \frac{1}{N}\Sigma_i(t_i - o_i)^2 \tag{19}$$

The training process is terminated when the specified MSE or the maximum number of epochs is achieved.

The mean absolute error (MAE) is a quantity used to measure how close outputs of the model or predictions are to the actual values. The mean absolute error is given by Eq. (20)

$$MAE = \frac{1}{n}\Sigma_{i=1}^{n} o_i - t_i \tag{20}$$

As the name suggests, the mean absolute error is an average of the absolute errors $|e_i| = |o_i - t_i|$, where t_i is the target which denotes the actual value of the Ra, o_i is the output denoting the network estimation value of the *Ra*, and *N* is the number of training patterns.

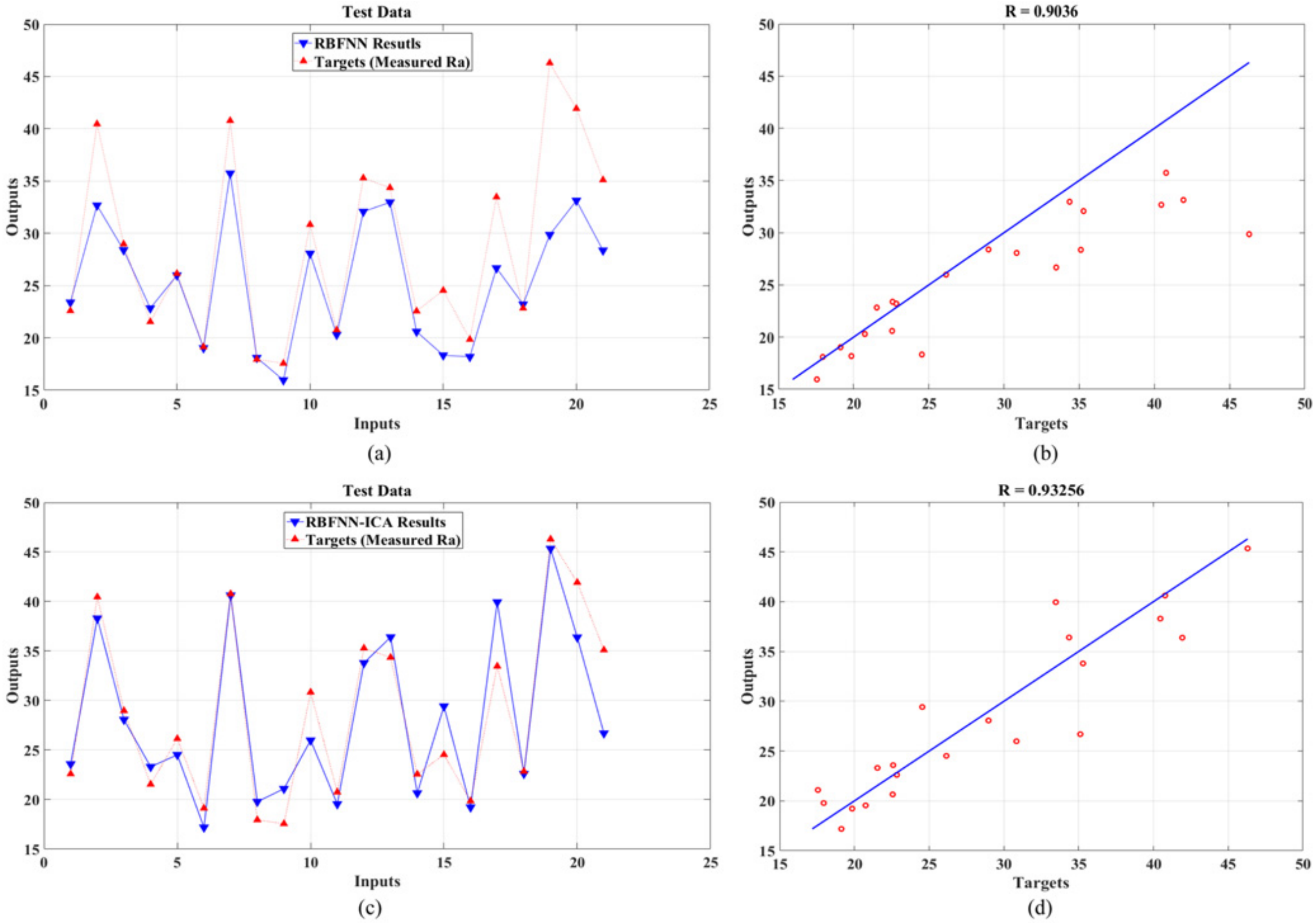

Fig. 6 the accuracy of the models for testing data in surface roughness prediction, (a) the performance of the RBFNN, (b) the correlation of the RBFNN, (c) the performance of the RBFNN-ICA, (d) the correlation of the RBFNN-ICA

4. Results and Discussion

4.1 Model performance

To represent the progress achieved with the recommended models, the most well-known analytical models, which were evaluated in a recent paper,[30] are compared with the RBFNN model. Table 5 shows the model performances (MAPE criteria) calculated at four build angle ranges in MATLAB® (v.R2014b) software, including 0° to 45°, 45° to 90°, 90° to 135° and 135° to 180°, which provide more details about the estimation error for different build angles.

It is concluded that a significant improvement was achieved using this method for surface roughness estimation in comparison with other analytical models. In test parts with Lt of 0.3302 mm, 0.254 mm, 0.253 (Lt_c), and 0.254 (Lt_a) the RBFNN modeling reached the minimum estimation error when compared to other models in all build angle intervals. Moreover, the intelligent RBFNN based on ICA, improves the overall performance of (MAPE) the network remarkably. Therefore, this assessment displays the advantage of neural networks and intelligent modeling application in surface roughness prediction.

4.2 Validity

Two different assessments were performed for the RBFNN and RBFNN-ICA models validation which are the simulation of the proposed models for truncheon test parts and 3D general parts.

4.2.1 Simulation for truncheon test part

The RBFNN and RBFNN-ICA models were simulated for four distinct parts as illustrated in Figs. 7(a) to 7(d). The MAPE and R^2 values for the network for each of the four different parts are shown in Table 6. Figs. 7(a) to 7(d), indicate that, for test parts with Lt of 0.254 mm, 0.3302 mm, 0.253 mm (Lt_c), and 0.254 (Lt_a) the RBFNN-ICA gave the closest estimations to the actual values.

According to Figs. 7(a) to 7(d), Table 5, and Table 6 it is concluded that the proposed models has an acceptable error close to that of the actual data and it also shows a significant predictability of the fitted surface roughness; thus, allowing process parameter optimization in the process planning stage. Consequently, the capability of this modeling to remedy the defects of analytical models has been validated.

4.2.2 Simulation for 3D general parts

In order to extend the ability of the proposed model to the 3D general components, a comprehensive validation on 3D parts with various conditions was carried out. The verification has been performed over the specimens from the literature[25,39] in this scope which were manufactured with different process parameters, shapes, production systems, and material. The required data for surface roughness

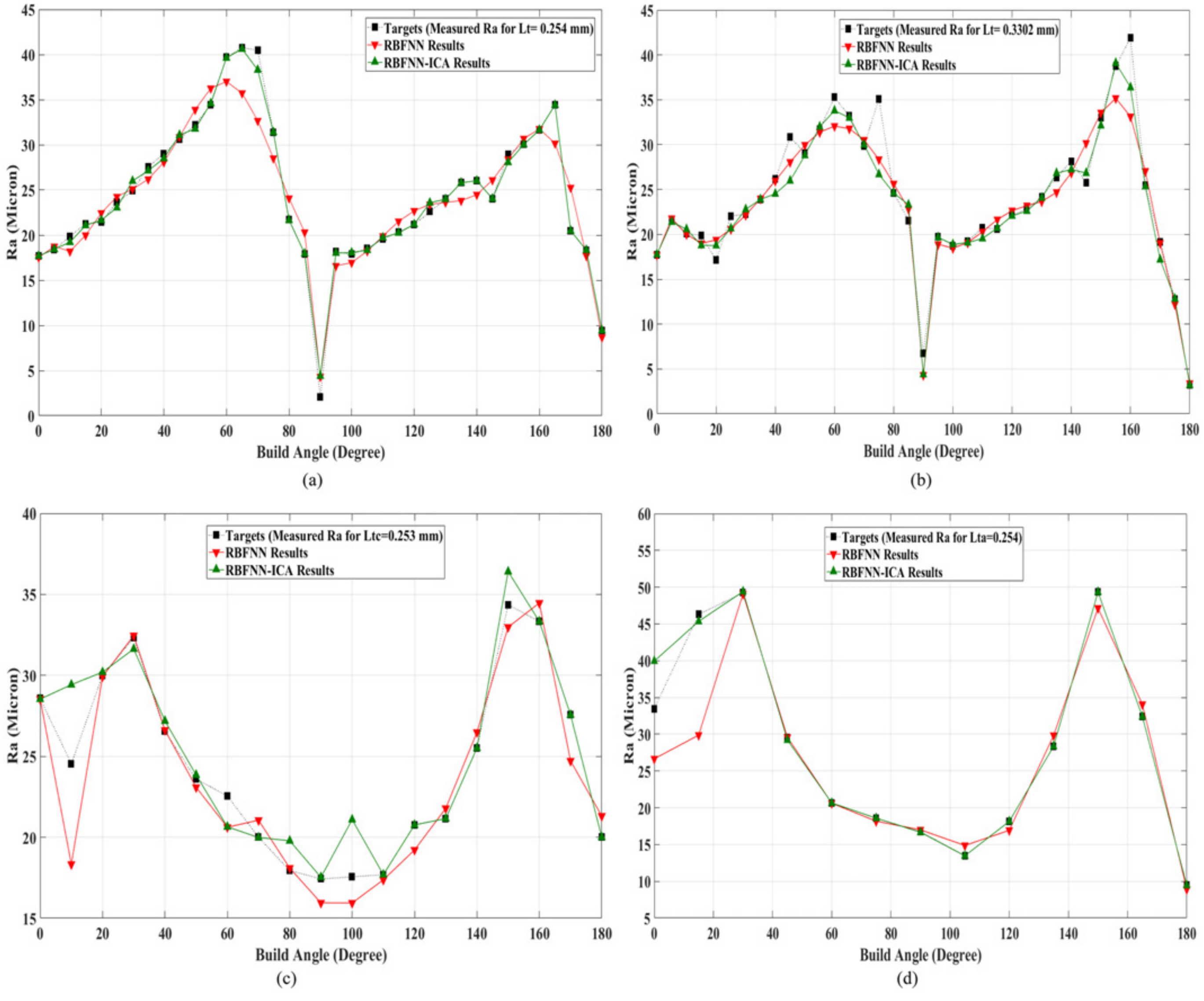

Fig. 7 Simulation results of the RBFNN and RBFNN-ICA models in different truncheon parts; (a) truncheon part with Lt=0.254 mm, (b) truncheon part with Lt=0.3302 mm, (c) truncheon part with Lt_c = 0.253 mm, and (d) truncheon part with Lt_a = 0.254 mm

assessment were derived from the literature.[25,39] This assessment permits to extend the capability of the model to most diffused material and to different apparatuses. Fig. 8(a) shows the specimens that were fabricated with ABS (FDM machine's specification: Stratasys, Dimension BST 768), ABSplus (FDM machine's specification: Stratasys, Fortus 400), ULTEM 9085 (FDM machine's specification: Stratasys, Fortus 400), and Polycarbonate (FDM machine's specification: Stratasys, Fortus 360) with Lt of 0.254 mm.[25] Fig. 8(b) shows the specimen with Lt of 0.254 mm, fabricated (FDM machine's specification: Stratasys, Fortus 400) for surface roughness evaluation in 19 different build angles of ABS material (named as PS in the following assessments).[39]

Figs. 9(a) to 9(e) represent the surface roughness estimation capability of RBFNN and RBFNN-ICA models for different parts fabricated in varied conditions. As observed, a good accordance between empirical data and the results of the proposed models is obtained. This is confirmed by performance criteria; in particular, the MAPE values of RBFNN and RBFNN-ICA models which are around or lower than 10% for all components. As represented in Table 7, regarding the MAPE, MSE, and MAE criteria, the overall performance of the RBFNN-ICA is better than the RBFNN which confirms the general strength of the ICA for proper optimization of the RBFNN parameters. However, it is noteworthy to indicate that the performance of RBFNN-ICA model can be further improved for general approaches by changing the algorithm's variables. Therefore, the proposed models can be efficiently used to predict the surface quality of a prototype, in an early stage of process planning. Altogether, these assessments demonstrate the strength of the recommended approaches for roughness estimation in varied conditions.

4.3 Robustness of networks

In order to evaluate the robustness of the proposed models, the effective parameters of the RBFNN were studied. Moreover, the input variables contribution on the output of the network was derived to determine the most dominant input variable in the rough surface creation.

Table 5 MAPE values of surface roughness prediction for different models

Model	Lt (mm)	Build angle (°)			
		$0 \le \theta \le 45$	$45 < \theta \le 90$	$90 < \theta \le 135$	$135 < \theta \le 180$
Pandey	0.3302	20.78	118.22	48.75	275.97
Campbell		1683.25	65.64	70.13	2221.69
Byun		70.15	150.85	244.09	47.20
Ahn		226.85	645.07	563.52	122.85
Mason		188.83	629.43	589.98	115.63
RBFNN		*3.20*	*21.19*	*4.66*	*7.25*
RBFNN-ICA		*1.71*	*13.25*	*0.98*	*0.35*
Pandey	0.254	13.56	183.03	16.54	99.49
Campbell		1279.15	67.88	59.28	1299.83
Byun		39.88	96.13	175.93	45.57
Ahn		138.33	950.35	434.21	76.30
Mason		112.91	932.79	451.24	89.29
RBFNN		*5.04*	*9.43*	*2.30*	*6.82*
RBFNN-ICA		*4.87*	*8.22*	*1.29*	*4.05*
Pandey	0.253 (Lt_c)	29.49	87.69	26.48	60.44
Campbell		723.36	64.16	56.51	648.42
Byun		47.69	158.48	195.88	52.52
Ahn		107.56	498.97	468.61	69.52
Mason		91.37	482.11	491.76	90.59
RBFNN		*5.28*	*5.06*	*5.36*	*5.63*
RBFNN-ICA		*4.80*	*4.09*	*5.04*	*1.19*
Pandey	0.254 (Lt_a)	45.66	98.93	33.91	121.43
Campbell		252.29	61.19	48.97	485.68
Byun		62.98	165.46	205.53	61.65
Ahn		88.42	568.92	488.19	19.86
Mason		90.45	550.59	511.09	43.39
RBFNN		*14.85*	*1.90*	*7.64*	*5.62*
RBFNN-ICA		*5.73*	*0.12*	$7.927e^{-05}$	$2.8121e^{-05}$

Table 6 Estimation of the performance of the network for different test parts

Network	Lt = 0.254 mm		Lt = 0.3302 mm		Lt_c = 0.253 mm		Lt_a = 0.254 mm	
	MAPE (%)	R^2	MAPE (%)	R^2	MAPE (%)	R^2	MAPE (%)	R^2
RBFNN	9.05	0.9375	5.89	0.9402	5.33	94.18	7.50	0.9385
RBFNN-ICA	4.08	0.9852	4.61	0.9808	3.78	97.04	1.79	0.9828

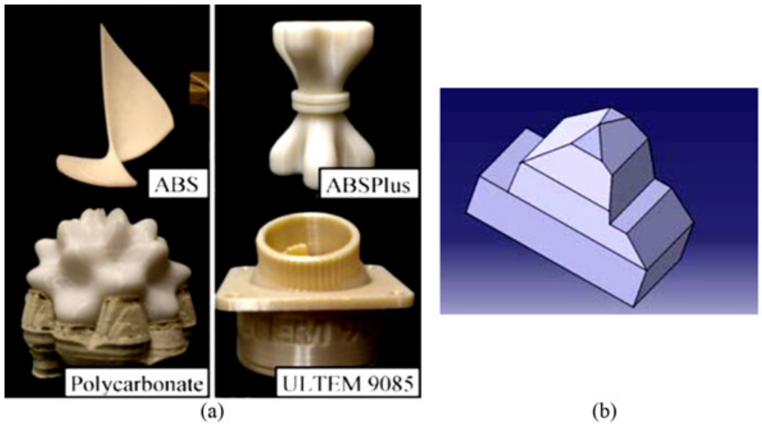

Fig. 8 Specimens for model verification, (a) different materials with complex shapes,[25] (b) the part with different slopes[39]

4.3.1 Sensitivity analysis

Sensitivity Analysis is used to determine how “sensitive” a model is with respect to the changes in the parameter values and to the changes in the model structure. The sensitivity coefficients describe the change in the system outputs due to variations in the parameters that affect the system. The weights method (Garson’s algorithm) is used in

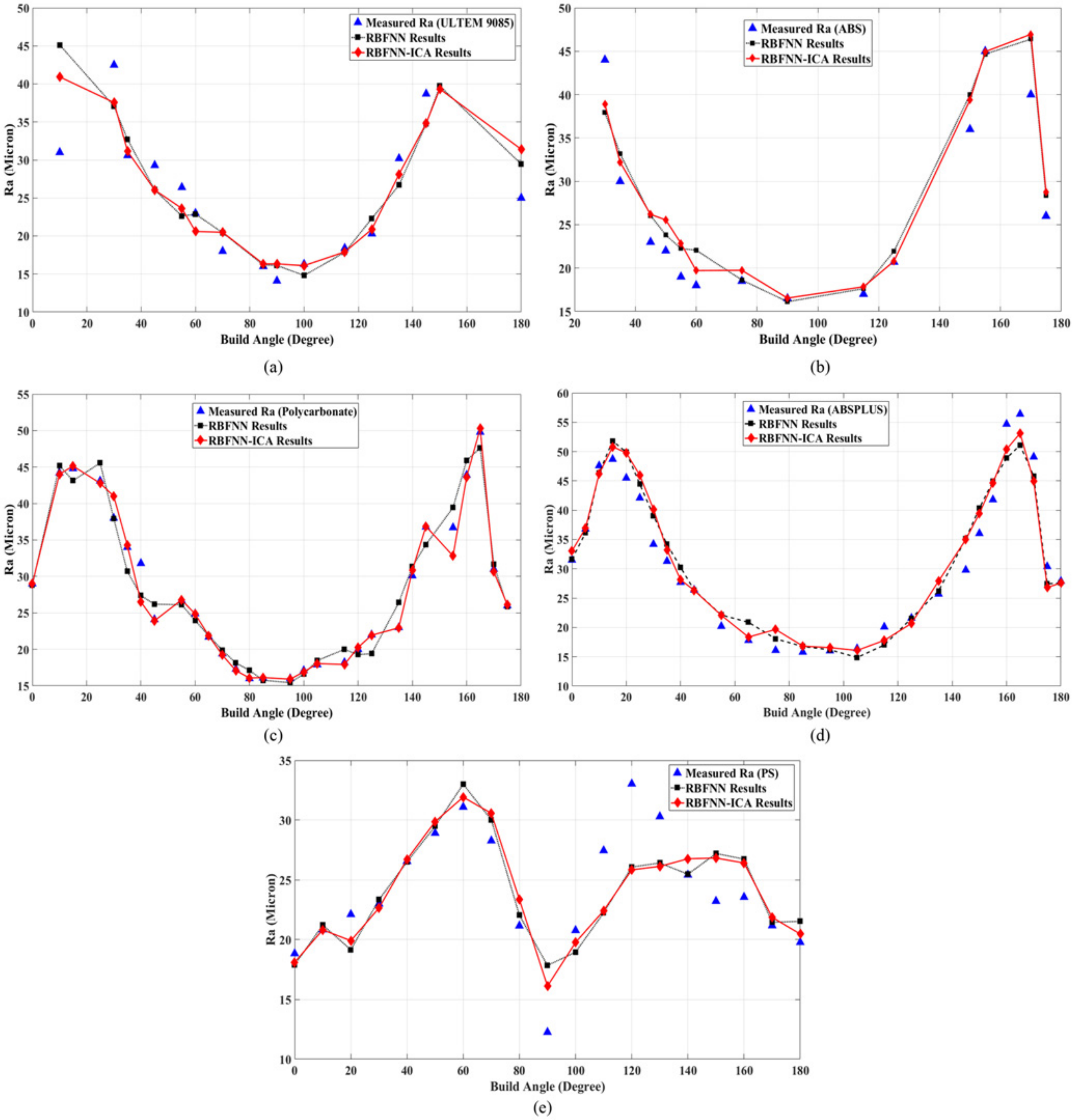

Fig. 9 Simulation of the RBFNN and RBFNN-ICA models for general 3D parts; (a) part built with ULTEM 9085, (b) part built with ABS, (c) part built with Polycarbonate, (d) part built with ABSPLUS, (e) part built with different slopes

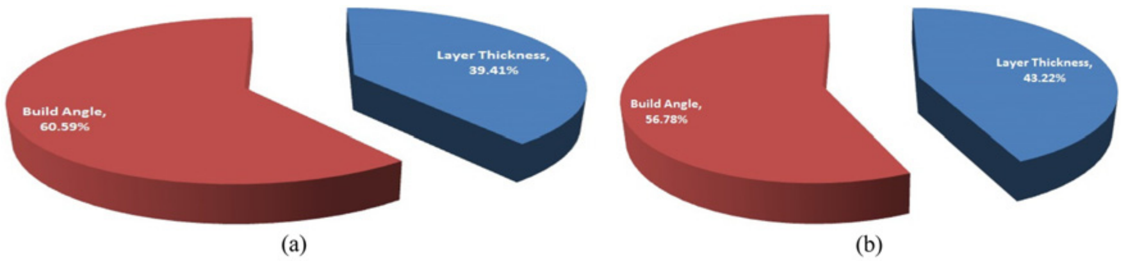

Fig. 10 Relative importance of (a) RBFNN-ICA and (b) RBFNN

Table 7 The performance of the RBFNN and RBFNN-ICA models for different parts

Test parts	Model	MSE	MAE	MAPE (%)
ABSPLUS	RBFNN	9.02	2.44	7.68
	RBFNN-ICA	6.86	1.97	6.21
Polycarbonate	RBFNN	3.29	1.42	5.09
	RBFNN-ICA	1.85	0.60	1.88
ABS	RBFNN	10.80	2.63	9.60
	RBFNN-ICA	8.66	2.14	7.82
ULTEM9085	RBFNN	24.54	3.61	12.46
	RBFNN-ICA	18.52	3.20	11.05
PS	RBFNN	9.05	2.23	9.92
	RBFNN-ICA	7.97	2.11	8.97

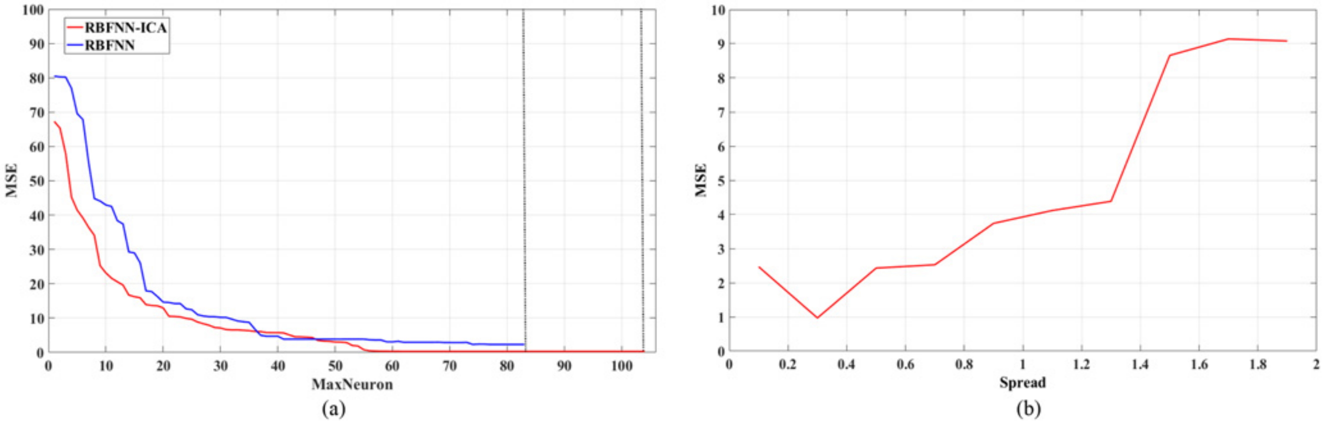

Fig. 11 (a) The effect of the number of neurons on the RBFNN and RBFNN-ICA performance (MSE versus the maximum neuron), (b) the effect of the spread parameter on the RBFNN-ICA performance (MSE versus spread)

this study as a procedure for partitioning the connection weights to calculate the relative importance of the various inputs and was first proposed by Garson et al.[40] and repeated by Goh et al..[41] The input variables that do not have a significant effect on the ANN performance can be excluded from the input variables. The relative importance (RI) of the input variables can be written as Eq. (21).

$$RI(\%)_i = \frac{\sum_{h=1}^{n_h}\left[\left(\frac{W_{hi}}{\sum_{i=1}^{n_i}W_{hi}}\right)W_{ho}\right]}{\sum_{i=1}^{n_i}\left[\sum_{h=1}^{n_h}\left(\frac{W_{hi}}{\sum_{i=1}^{n_i}W_{hi}}\right)W_{ho}\right]} \tag{21}$$

where n_i and n_h are the number of neurons in the input and the hidden layers, respectively. W_{hi} is the absolute value of connection weights between the input and the hidden layers and W_{ho} is the absolute value of the connection weights between the hidden layers and the output layers. The *RI* of input variables in the RBFNN and RBFNN-ICA models are illustrated in Fig. 10. Fig. 10 shows that the input parameter build angle has the significant effect on the output of the RBFNN and RBFNN-ICA models with 56.78% and 60.59% of *RI*, respectively. This outcome reveals the dominant effect of build angle variation in rough surface creation.

4.3.2 Network parameters analysis

According to the literature,[31,35,42] the performance of the RBFNN is highly sensitive to the size of neuron number and the value of the spread variables. Therefore, the effect of the number of neurons and spread variables were studied on the proposed networks performances (performance was determined by MSE criterion in training data sets).

In order to further study the influence of the number of maximum neurons on robustness of the RBF network, the performance (MSE) of the RBFNN and RBFN-ICA models were illustrated against the variation of neuron number. The RBFNN was trained with the optimized data division determined by ICA which was defined as the training and testing data percentage of 80% and 20%, respectively. As observed in Fig. 11(a), the general trend for the prediction error of training data sets increases by decreasing the number of neurons. This could be interpreted as the RBF network wanting to allocate one neuron for each input data in the early stages. The RBFNN and RBFNN-ICA models reached to the optimum goal by using 74 and 55 neurons, respectively. This attempt gave the performance (MSE) of 2.31 and 0.28 for RBFNN and RBFNN-ICA models, respectively. Fig. 11(b) shows the performance of the RBFNN-ICA in finding the optimum spread value where this parameter's default value was set to 1 in the RBFNN. It is notable to indicate that the RBFNN model performs well in the spread value of around 1 and the optimum values cannot be determined without a large number of trials, so this duty was performed by intelligent algorithm. Therefore, for this variable, the parameter analysis includes only the determination of the optimum values for RBFNN-ICA model. As depicted in Fig. 11(b), the general trend for this parameter represents that the prediction error increases by increasing the spread value. Therefore, it is concluded that the best spread value is 0.3913. Altogether, the ICA optimization can effectively improve the performance of the RBFNN by improving the maximum neuron size and spread variables.

5. Conclusion

The limiting aspect of FDM is the surface quality of the fabricated parts which is initiated from the layered manufacturing principle. To improve the surface roughness of the part fabricated by FDM, modeling of the surface roughness distribution for optimizing the effective parameters before the fabrication process is used for more precise planning of the AM process. Analytical estimation of surface roughness distribution has been applied to improve the surface quality. However, such estimates may often not accurately predict the surface roughness for all the ranges of the surface build angles. Therefore, in this paper, an RBFNN was applied using empirical data derived from a specific test part. Precise adjustment of the process variables (such as weight connections, biases, spread, and the number of neurons of the RBFNN model) is the most important part of the intelligent training algorithm. ICA is one of the optimization algorithms with high performance of convergence, fast speed and limited number of parameters. In this direction, Using ICA the effective parameters were optimized accordingly. The RBFNN-ICA model had the best accuracy and precision in estimating the surface roughness of the FDM built parts. The MAPE values of the RBFNN and RBFNN-ICA models were 7.11% and 3.64%, respectively, and the MSE were 7.48 and 2.27, respectively.

The comparison of the proposed models with analytical models demonstrated the improvement achieved in the performance of the estimation. The significant improvement in the estimation of surface roughness was achieved at all ranges. The results of the simulation confirmed the accuracy of more fitted responses in the RBFNN-ICA model. The robustness of the RBF network was studied using an assessment over effective variables of RBFNN model and the sensitivity analysis evaluation of the input parameters, where the proposed models showed the best performance at the optimum values of 55 for maximum neuron number and the spread value of 0.3913. The proposed neural network model based on intelligent optimization is the best and most appropriate method to predict surface roughness because it uses empirical data that simultaneously contains all of the factors that affect surface roughness, such as roughness from the staircase effect, support burrs, material properties and other factors.

REFERENCES

1. Bernard, A. and Fischer, A., “New Trends in Rapid Product Development,” CIRP Annals-Manufacturing Technology, Vol. 51, No. 2, pp. 635-652, 2002.

2. Hopkinson, N., Hague, R., and Dickens, P., “Rapid Manufacturing: An Industrial Revolution for the Digital Age,” John Wiley & Sons, 2006.

3. Upcraft, S. and Fletcher, R., “The Rapid Prototyping Technologies,” Assembly Automation, Vol. 23, No. 4, pp. 318-330, 2003.

4. Mansour, S. and Hague, R., “Impact of Rapid Manufacturing on Design for Manufacture for Injection Moulding,” Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, Vol. 217, No. 4, pp. 453-461, 2003.

5. Pham, D. T. and Gault, R. S., “A Comparison of Rapid Prototyping Technologies,” International Journal of Machine Tools and Manufacture, Vol. 38, No. 10, pp. 1257-1287, 1998.

6. Wiedemann, B. and Jantzen, H.-A., “Strategies and Applications for Rapid Product and Process Development in Daimler-Benz AG,” Computers in Industry, Vol. 39, No. 1, pp. 11-25, 1999.

7. Yan, X. and Gu, P., “A Review of Rapid Prototyping Technologies and Systems,” Computer-Aided Design, Vol. 28, No. 4, pp. 307-318, 1996.

8. Onuh, S. O. and Yusuf, Y. Y., “Rapid Prototyping Technology: Applications and Benefits for Rapid Product Development,” Journal of Intelligent Manufacturing, Vol. 10, No. 3-4, pp. 301-311, 1999.

9. Pandey, P. M., Reddy, N. V., and Dhande, S. G., “Slicing Procedures in Layered Manufacturing: A Review,” Rapid Prototyping Journal, Vol. 9, No. 5, pp. 274-288, 2003.

10. Sood, A. K., Equbal, A., Toppo, V., Ohdar, R. K., and Mahapatra, S. S., “An Investigation on Sliding Wear of FDM Built Parts,” CIRP Journal of Manufacturing Science and Technology, Vol. 5, No. 1, pp. 48-54, 2012.

11. Ahn, D., Kweon, J.-H., Kwon, S., Song, J., and Lee, S., “Representation of Surface Roughness in Fused Deposition Modeling,” Journal of Materials Processing Technology, Vol. 209, No. 15, pp. 5593-5600, 2009.

12. Li, A., Zhang, Z., Wang, D., and Yang, J., “Optimization Method to Fabrication Orientation of Parts in Fused Deposition Modeling Rapid Prototyping,” Proc. of International Conference on Mechanic Automation and Control Engineering (MACE), pp. 416-419, 2010.

13. Byun, H. S. and Lee, K. H., “Determination of Optimal Build Direction in Rapid Prototyping with Variable Slicing,” The International Journal of Advanced Manufacturing Technology, Vol. 28, No. 3-4, pp. 307-313, 2006.

14. Chang, D.-Y. and Huang, B.-H., “Studies on Profile Error and Extruding Aperture for the RP Parts using the Fused Deposition Modeling Process,” The International Journal of Advanced Manufacturing Technology, Vol. 53, No. 9-12, pp. 1027-1037, 2011.

15. Thrimurthulu, K., Pandey, P. M., and Reddy, N. V., “Optimum Part Deposition Orientation in Fused Deposition Modeling,” International Journal of Machine Tools and Manufacture, Vol. 44, No. 6, pp. 585-594, 2004.

16. Armillotta, A., “Assessment of surface Quality on Textured FDM Prototypes,” Rapid Prototyping Journal, Vol. 12, No. 1, pp. 35-41, 2006.

17. Pandey, P. M., Reddy, N. V., and Dhande, S. G., “Real Time Adaptive

Slicing for Fused Deposition Modelling," International Journal of Machine Tools and Manufacture, Vol. 43, No. 1, pp. 61-71, 2003.

18. Pandey, P. M., Reddy, N. V., and Dhande, S. G., "Improvement of Surface Finish by Staircase Machining in Fused Deposition Modeling," Journal of Materials Processing Technology, Vol. 132, No. 1, pp. 323-331, 2003.

19. Campbell, R. I., Martorelli, M., and Lee, H. S., "Surface Roughness Visualisation for Rapid Prototyping Models," Computer-Aided Design, Vol. 34, No. 10, pp. 717-725, 2002.

20. Reeves, P. E. and Cobb, R. C., "Surface Deviation Modelling of LTM Processes - A Comparative Analysis," Proc. of 5th European Conference on Rapid Prototyping and Manufacturing, 1995.

21. Ahn, D., Kim, H., and Lee, S., "Surface Roughness Prediction Using Measured Data and Interpolation in Layered Manufacturing," Journal of Materials Processing Technology, Vol. 209, No. 2, pp. 664-671, 2009.

22. Perez, C. L., Vivancos, J., and Sebastian, M. A., "Surface Roughness Analysis in Layered Forming Processes," Precision Engineering, Vol. 25, No. 1, pp. 1-12, 2001.

23. Bordoni, M. and Boschetto, A., "Thickening of Surfaces for Direct Additive Manufacturing Fabrication," Rapid Prototyping Journal, Vol. 18, No. 4, pp. 308-318, 2012.

24. Ahn, D., Kim, H., and Lee, S., "Fabrication Direction Optimization to Minimize Post-Machining in Layered Manufacturing," International Journal of Machine Tools and Manufacture, Vol. 47, No. 3, pp. 593-606, 2007.

25. Boschetto, A., Giordano, V., and Veniali, F., "Surface Roughness Prediction in Fused Deposition Modelling by Neural Networks," The International Journal of Advanced Manufacturing Technology, Vol. 67, No. 9-12, pp. 2727-2742, 2013.

26. Ahn, D.-K., Kwon, S.-M., and Lee, S.-H., "Expression for Surface Roughness Distribution of FDM Processed Parts," Proc. of International Conference on Smart Manufacturing Application, pp. 490-493, 2008.

27. Mason, A., "Multi-Axis Hybrid Rapid Prototyping using Fusion Deposition Modeling," M.Sc. Thesis, Department of Mechanical Engineering, Ryerson University, 2006.

28. Nourghassemi, B., "Surface Roughness Estimation for FDM Systems," M.Sc. Thesis, Department of Mechanical Engineering, Ryerson University, 2011.

29. Reeves, P. E. and Cobb, R. C., "Reducing the Surface Deviation of Stereolithography Using in-Process Techniques," Rapid Prototyping Journal, Vol. 3, No. 1, pp. 20-31, 1997.

30. Rahmati, S. and Vahabli, E., "Evaluation of Analytical Modeling for Improvement of Surface Roughness of FDM Test Part using Measurement Results," The International Journal of Advanced Manufacturing Technology, Vol. 79, No. 5-8, pp. 823-829, 2015.

31. Haykin, S. S., Haykin, S. S., Haykin, S. S., and Haykin, S. S., "Neural Networks and Learning Machines," Pearson Upper Saddle River, 2009.

32. Jain, A. K., Mao, J., and Mohiuddin, K. M., "Artificial Neural Networks: A Tutorial," IEEE Computer, Vol. 29, No. 3, pp. 31-44, 1996.

33. Foody, G. M., "Supervised Image Classification by MLP and RBF Neural Networks with and without an Exhaustively Defined Set of Classes," International Journal of Remote Sensing, Vol. 25, No. 15, pp. 3091-3104, 2004.

34. Assi, A. H., "Engineering Education and Research using Matlab," InTech, 2011.

35. Du, K.-L. and Swamy, M. N., "Neural Networks in a Softcomputing Framework," Springer Science & Business Media, 2006.

36. Beale, M. H., Hagan, M. T., and Demuth, H. B., "Neural Netword Toolbox User Guide," https://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf (Accessed 10 NOV 2016)

37. Atashpaz-Gargari, E. and Lucas, C., "Imperialist Competitive Algorithm: An Algorithm for Optimization Inspired by Imperialistic Competition," Proc. of IEEE Congress on Evolutionary Computation, pp. 4661-4667, 2007.

38. Pandey, P. M., Thrimurthulu, K., and Reddy, N. V., "Optimal Part Deposition Orientation in FDM by using a Multicriteria Genetic Algorithm," International Journal of Production Research, Vol. 42, No. 19, pp. 4069-4089, 2004.

39. Sreedhar, P., MathikumarManikandan, C., and Jothi, G., "Experimental Investigation of Surface Roughness for Fused Deposition Modeled Part with Different Angular Orientation," Technology, Vol. 5, No. 3, pp. 21-28, 2012.

40. Garson, D. G., "Interpreting Neural Network Connection Weights," AI Expert, Vol. 6, No. 7, pp. 47-51, 1991.

41. Goh, A. T. C., "Back-Propagation Neural Networks for Modeling Complex Systems," Artificial Intelligence in Engineering, Vol. 9, No. 3, pp. 143-151, 1995.

42. Huang, G.-B., Saratchandran, P., and Sundararajan, N., "A Generalized Growing and Pruning RBF (GGAP-RBF) Neural Network for Function Approximation," IEEE Transactions on Neural Networks, Vol. 16, No. 1, pp. 57-67, 2005.