Int J Adv Manuf Technol (2015) 78:781–793
DOI 10.1007/s00170-014-6679-5

ORIGINAL ARTICLE

Process characterisation of 3D-printed FDM components using improved evolutionary computational approach

V. Vijayaraghavan · A. Garg · Jasmine Siu Lee Lam · B. Panda · S. S. Mahapatra

Received: 14 May 2014 / Accepted: 1 December 2014 / Published online: 13 December 2014

Abstract Fused deposition modelling (FDM) is an additive manufacturing technique deployed to fabricate the functional components leading to shorter product development times with less human intervention. Typical characteristics such as surface roughness, mechanical strength and dimensional accuracy are found to influence the wear strength of FDM fabricated components. It would be useful to determine an explicit numerical model to describe the correlation between various output process parameters and input parameters. In this paper, we have proposed an improved approach of multi-gene genetic programming (Im-MGGP) to formulate the functional relationship between wear strength and input process variables of the FDM process. It was found that the improved approach performs better than MGGP, SVR and ANN models and is able to generalise wear strength of the FDM prototype satisfactorily. Further, sensitivity and parametric analysis is conducted to study the influence of each input variable on the wear strength of the FDM fabricated components. It was found that the input parameter, air gap, has the maximum influence on the wear strength of the FDM fabricated component.

V. Vijayaraghavan
School of Mechanical and Aerospace Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore

A. Garg · J. S. L. Lam (✉)
School of Civil and Environmental Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798, Singapore
e-mail: SLLam@ntu.edu.sg

S. S. Mahapatra
Department of Mechanical Engineering, National Institute of Technology, Rourkela 769008, India

B. Panda
Department of Production Engineering, Veer Surendra Sai University of Technology (VSSUT), Sambalpur 768018, India

Keywords Wear prediction · Fused deposition modelling · Rapid prototyping · Evolutionary computation

1 Introduction

Fused deposition modelling (FDM) is a three-dimensional (3D) printing process that makes use of rapid prototyping (RP) technology to build 3D solid complex parts from the computer-aided design data. The process is entirely automatic without the use of tooling and involves less human intervention. The FDM process has hence resulted in applications in functional prototype development in the medical sector, automobile industries, construction industries, space applications and tool and die making [1, 2].

Literature reveals that the properties of the FDM fabricated parts such as wear strength, tensile strength, compressive strength and surface roughness are function of input process variables and can be significantly improved with its proper adjustment [3, 4]. For an appropriate selection of input process variables as well as with an advent of development of capital intensive FDM machines, the desire for mathematical modelling has been strengthened. One route is to develop new materials, but this may require one to have expert knowledge about the characteristics of materials at different operating conditions [5–7]. Another route is to develop mathematical models that can be used as an alternative for studying the FDM process. In this context, several physics-based models have been formulated [8–11]. The formulation of the physics-based models requires in-depth understanding of the process and is not an easy task in presence of partial information about the process. Therefore, researchers shifted focus on developing models based on only the given data.

To develop models based on only the given data, several well-known computational intelligence (CI) methods such as artificial neural networks (ANNs), fuzzy logic, adaptive-

network-based fuzzy inference system, genetic programming (GP) and support vector regression (SVR) have been applied to formulate the relationship between output and input process variables of the FDM process [12–16]. Among these methods, GP possesses the ability to evolve models structure and its coefficients automatically [12, 17–20]. The most popular variant of GP used recently is multi-gene genetic programming (MGGP) [21–23]. Despite of good number applications of MGGP in solving symbolic regression problems [24–27], it has a limitation for producing models that *over-fit* on the testing data. This indicates that the underlying relationships of the whole data were not learned, and instead, a set of relationships existing only on training cases were learned, but these have no correspondence over the whole possible set of cases. The poor performance of models on the testing data is undesirable and is likely to give falsify information about the process. *Over-fitting* in GP is the popular problem among researchers and have been paid less attention [28, 29].

Therefore, the present work proposed an improved approach of MGGP (Im-MGGP) in modelling of wear strength of FDM fabricated component. An improved approach makes use of statistical learning principle, structural risk minimisation, for imparting the generalisation ability to the model. Idea for introducing this principle came from SVR method, where this principle plays a key role in invoking the generalisation. Unlike the standard GP, each model participating in this approach is formulated from the set of combination of genes. Experiments were conducted to fabricate the FDM prototypes. The wear strength of the FDM fabricated prototypes is measured based on five input variables such as layer thickness, orientation, raster angle, raster width and air gap. Based on data obtained from the experiments, the proposed method is applied and its performance is compared to that of the other three potential models: standardised MGGP, SVR and ANN. Further, the sensitivity and parametric analysis is conducted to validate the robustness of the model by unveiling the dominant input process variables and hidden non-linear relationships.

2 FDM process

2.1 Experimental details and data collection

FDM by Stratasys Inc., USA has been widely used in the manufacturing industry from the 1990s as one of the layered manufacturing techniques. It has significant advantages in terms of cost effective, lesser build time, elimination of expensive tooling, flexibility and the possibility of producing very complex parts and shapes. Previous research mainly focused on dimensional accuracy of the built part [15], surface roughness improvement [8] and mechanical strength characterisation [30, 31]. These works demonstrated that properties of the built parts depend on input process variables and can be improved by their suitable selection. Wear strength is an important characteristic for the durability of part and very little work is done to understand the wear characteristic of the RP processed part [32]. To address this gap, the present work emphasised the sliding wear behaviour of FDM processed part and its relationship with process parameters.

The FDM process to be modelled is referred from an earlier study conducted on an investigation on sliding wear of FDM built parts [13]. The process input variables considered are layer thickness (x_1), orientation (x_2), raster angle (x_3), raster width (x_4) and air gap (x_5) and other factors such as part fill style, contour width and visible surface are fixed. For each of the input process variables, values at three levels (low, centre and high) are considered as per guidelines of machine manufacturer and industrial application. The output variable considered is wear strength (y), which is computed by calculating the ratio of wear volume and sliding distance. Wear volume (mm^3) is determined by multiplying the cross-sectional area with decrease in height, and sliding distance (m) is obtained by multiplying time with speed of rotation. For the wear testing, pin on disk apparatus (Ducom, TR- 20LE-M5) is used. The contact path diameter is set as 120 mm, and the disc is made to rotate with constant speed of 1 m/s. The disc is made of EN 31 hardened steel (hardness RC 62 and roughness (R_a) 0.32–0.35 μm).

Half factorial 2^5 unblocked design having 16 experimental run, 10 ($2K$, where $K=5$) axial run and 6 centre run have been used to create 32 set of data points. Nature of the data set collected is shown by its descriptive statistics in Table 1. Selection of training and testing data set affect the prediction ability of the model. In this work, Kennard and Stone algorithm is used to select the appropriate training and testing data set. The algorithm selects the training samples in such a way that the data is distributed uniformly throughout the domain. Twenty-six samples are chosen as the set of training data with the remaining as test samples. The training data is used for formulating the models, whereas the test data samples are used for testing the generalisation ability of the models.

3 Computational intelligence methods

3.1 Multi-gene genetic programming

In order to understand the concepts of MGGP method, the basics of GP are first discussed in brief. GP generates models automatically based on the given data using Darwinian principle of "Survival of the fittest" [33]. Working principle of GP is the same as GA but the only difference between them, is that, GA evolves solutions represented by strings (binary or

Table 1 Descriptive statistics of the input and output process variables used in FDM experiment

Parameter	Layer thickness (x_1)	Orientation (x_2)	Raster angle (x_3)	Raster width (x_4)	Air gap (x_5)	Wear strength (mm^3/m) (y)
Mean	0.18	15	30	0.4564	0.004	0.0266
Standard error	0.008	2.020	4.04	0.006	0.0005	0.0016
Median	0.178	15	30	0.4564	0.004	0.0269
Standard deviation	0.048	11.43	22.86	0.038	0.003	0.009
Variance	0.002	130.64	522.58	0.001	9.29E−6	8.86E−5
Kurtosis	−1.20	−1.22	−1.22	−1.22	−1.22	−0.50
Skewness	0.35	0	0	−9.5E−15	−2.02E−15	0.10
Minimum	0.127	0	0	0.4064	0	0.011
Maximum	0.254	30	60	0.5064	0.008	0.048

real number) of fixed length, whereas GP generates solutions represented by tree structures of varying sizes.

GP algorithm starts by generating the models randomly. The number of models generated is represented by the population size. The models are formed by combining the elements randomly from the functional and terminal set. A function set F usually comprises of elements such as basic arithmetic operations (+, −, ×, /, etc.), Boolean operators (AND, OR, etc.) or other operators as defined by the user. The terminal set T comprises elements such as numerical constants and input decision variables of the process. An example of a model constructed is shown in Fig. 1. The performance of initial population is evaluated on the training data based on the fitness function, namely, root mean square error (RMSE), given by

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N} |G_i - A_i|^2}{N}} \tag{1}$$

where G_i is the valued predicted of ith data sample by the MGGP model, G_i is the actual value of the ith data sample and N is the number of training samples.

Based on the performance on training data, algorithm selects models for genetic operations such as reproduction, mutation and crossover. The purpose of performing genetic operations is to form a new population that represent new generation. A subtree crossover operation is used, in which a branch of tree is selected from both the parents and is randomly exchanged. A subtree mutation operation is used, in which, the node (terminal or functional) is selected at random from the tree, and is replaced by branch/or the whole new generated random tree. This iterative phenomenon of generating new populations/generations continues as long as the termination criterion is not met. Termination criterion is the maximum number of generations and the threshold error of the model, whichever is achieved earlier.

Unlike GP, the evolutionary phase of the MGGP algorithm evolves models, where every model is formed by combining set of trees/genes. The MGGP algorithm is outlined as follows:

```
BEGIN
Step 1: Formulate problem
Step 2: MGGP algorithm
Begin
2.a Set parameters such as terminal and function set, number of generations, population size, genetic operators rate and maximum number of genes
2.b Generate initial population of genes
2.c Models are formed by combining set of genes using least squares method
2.d Evaluate performance of models based on RMSE
2.e Apply genetic operations and form the new population
2.f Cross-check the models performance against the termination criterion, and if not satisfied, GOTO Step 2.e
End;
END;
```

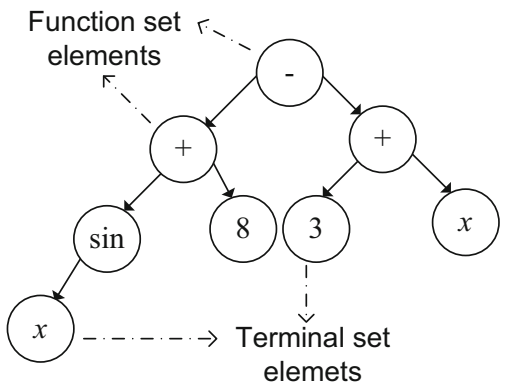

Fig. 1 Example of the GP model $8+\sin(x)-3+x$

3.1.1 Settings of parameter for implementation of MGGP

Parameter settings play a major role in implementing the MGGP algorithm in an effective manner. A trial and error method is used to select the parameter settings (Table 2). The function set consists of broader set of elements so as to evolve variety of non-linear forms of mathematical models. The values of population size and number of generations fairly depend on the complexity of the data. Based on previous applications of the algorithm by Garg et al. [34–39], the population size and number of generations should be fairly large for data of higher complexity, so as to find the models with minimum error. Maximum number of genes and maximum depth of the gene influences the size and the number of models to be searched in the global space. The maximum number of genes and maximum depth of gene is chosen at seven and six, respectively.

GPTIPS software [40, 41] is used for the implementation of MGGP algorithm. This software is a new "genetic programming and symbolic regression" code written based on MGGP [42] for the use with MATLAB. MGGP method is applied to the data set as collected in Section 2. The best MGGP model (see Eq. 2) is selected based on minimum RMSE on training data from all runs, and its performance is discussed in Section 4.

3.2 Improved multi-gene genetic programming

Since the MGGP method evolves models that are formed by combining the set of genes, it tends to produce large size models that may over-fit on the testing data. This indicates that the underlying relationships of the whole data were not learned, and instead a set of relationships existing only on training cases were learned, but these have no correspondence over the whole possible set of cases. The poor performance of models on testing data is undesirable and is likely to give falsify information about the process. Therefore, an improved approach of MGGP is proposed. In this approach, a well-established statistical criterion, SRM, is integrated in the paradigm of MGGP. SRM is a modified form of empirical risk minimisation principle and minimises the upper bound on the expected risk. The algorithm of the proposed approach is as follows:

```
{
BEGIN
Step 1: Formulate problem
Step 2: MGGP algorithm
Begin
2.a Set initial parameters such as function set, terminal set, population size, number of generations, genetic operators rate and maximum number of genes.
2.b Generate initial population of models
Step 3: Define SRM criterion
3.a Evaluation of models using SRM criterion
3.b If termination criterion is satisfied select the best model, else apply genetic operations
3.c Select the model based on minimum training error
End;
End;
End;
{
```

Each model participating in the evolutionary process is combination of several trees/genes, and their performance is evaluated using SRM criterion given by

$$\mathrm{SRM} = \frac{\mathrm{SSE}}{N}\left(1-\sqrt{\left(\left(\frac{b}{N}-\left(\frac{b}{N}\log\left(\frac{b}{N}\right)\right)+\left(\frac{\log\left(\frac{b}{N}\right)}{2N}\right)\right)\right)}\right)^{-1} \quad (3)$$

where b is the number of nodes of GP tree (size of model), SSE is the sum of square of error of GP model on the training data and N is the number of training samples. SRM add size term (b) to the empirical error, and thereby punishes the models of larger size, and thus prevents over-fitting.

3.2.1 Settings of parameter for implementation of Im-MGGP

For the fair comparison with the MGGP method, the parameter settings of the Im-MGGP method are kept the same as discussed in Section 3.1.1. The Im-MGGP method is applied on the data set as discussed in Section 2. The best model (Eq. 4) is selected based on minimum RMSE on training data, and its performance is discussed in Section 4.

Table 2 Parameter settings for MGGP and Im-MGGP

Parameters	Values assigned
Runs	10
Population size	400
Number of generations	100
Tournament size	2
Max depth of tree	6
Max genes	7
Functional set (F)	(Multiply, plus, minus, tan, tanh, sin, cos, plocg, exp)
Terminal set (T)	(x_1, x_2, x_3, x_4, x_5 [−10 10])
Crossover probability rate	0.85
Reproduction probability rate	0.10
Mutation probability rate	0.05

Table 3 Parameter settings for ANN

Parameters	Values assigned
Training data set	25
Testing data set	7
Number of hidden layer	1–4
Number of neurons in hidden layer	2–9
Activation function	Sigmoid
Number of epochs	1000
Learning rate	0.70
Architecture selection	Trial and error
Target goal mean square error	10^{-5}
Minimum performance gradient	10^{-5}

3.3 Support vector regression

SVR is a well-known and applied CI method in the field of the data modelling [43–45]. SVR method came from the support vector machine (SVMs), which has been applied for solving the classification problems. Unlike statistical regression methods, SVR is void of statistical assumptions and drives the data for generating non-linear models.

Since, SVR works on the principle of SRM, it is well known for invoking generalisation ability to the models. Input variables in the lower dimensional space are projected into a higher dimensional space H so as to convert the regression problem with non-linearity to the linear regression problem. To assist with such conversion, several hyperspace or transfer function can be used.

SVR model is model developed based on the training data. In this study, there are five input process variables and one output variable. The SVR model formulated is given by

$$y = z(x) = \sum_{i=1}^{N} w_i \rho_i(x) + b = w^H \rho(x) + b \quad (5)$$

where the function $\rho_i(x)$ is the transformed higher-dimensional space$w = [w_1 w_2 \ldots w_N]^H$ and $\rho = [\rho_1 \rho_2 \ldots \rho_N]^H$

Equation (5) represents a non-linear function, which projects the input variable space into a higher dimensional space H. The model given by $\rho(x)$ is linear in nature and is a converted form of original non-linear model in higher dimensional space. The kernel function learns from the data discussed in Section 2 and the regularised risk function (L_r) is minimised. By minimising this risk function L_r, the parameters w (weight) and b (bias) are estimated.

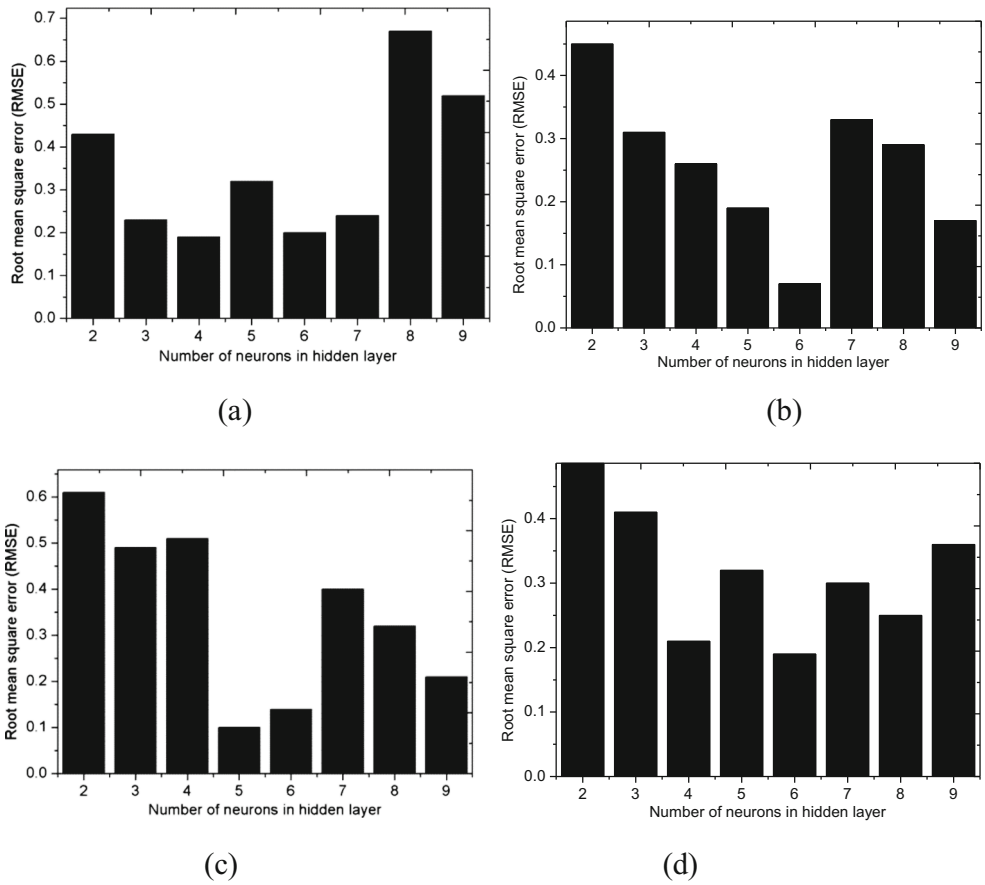

Fig. 2 Impact of number of neurons on performance of ANN along **a** number of hidden layer 1, **b** number of hidden layer 2, **c** number of hidden layer 3 and **d** number of hidden layer 4

$$L_r(w) = \frac{1}{2}w^T w + \lambda\sum_{i=1}^{N} |y_i - z(x)|_e \quad (6)$$

$$\text{where } |y_i - z(x)|_e = \begin{cases} 0, & \text{if } |y_i - z(x)| < \varepsilon \\ |y_i - z(x)| - \varepsilon, & \text{otherwise} \end{cases}$$

The weight vector norm and the approximation error trade-offs is regulated by the regularisation parameter (λ). With an increase in values of λ or weight vector norm, the approximation error can be decreased but deregulating the control of regularisation parameter (λ) may also results in over-fitting of the SVR model. Values of λ and ε, where ε is the tolerance level, are fixed by the practitioner. ε-insensitive loss function ($|y_i - z(x)|_e$) given by Eq. 6 is minimised so as to estimate the parameters. If the values predicted by the SVR model $z(x)$ lies within the defined tolerance level ε, the values of loss function is zero, and for the points outside ε, it is the magnitude of the difference between the values predicted by the SVR model and tolerance level ε. The points on the margin lines ($y = z(x) \pm \varepsilon$) are called support vectors, whereas those outside are known as error set.

3.3.1 Settings of parameter for implementation of SVR

Parameter settings of SVR such as selection of kernel function play a major role in learning and minimising the loss function efficiently. In this study, kernel function, namely radial basis function (RBF), is chosen based on its popularity among researchers for faster and efficient training of the SVR model. To implement SVR, LS-SVM tool box [46] built in MATLAB is applied on the data as discussed in Section 2. This toolbox has been used extensively by researchers [47–49] to solve symbolic regression problems of varying nature. RBF parameters, namely, λ and σ, are determined using coupled simulated annealing (CSA) and a grid search method. Firstly, the CSA determines the good initial values of λ and σ, and then, these are passed to the grid search method which uses cross-validation approach to fine tune the parameters. A SVR model with optimal parameters λ=102.06 and σ=0.0285 is found at the second iteration. The performance of the SVR model is discussed in Section 4.

3.4 Artificial neural network

ANN is one among the popular CI method based on the principle of neural networks found in nervous system of living organisms [50, 51]. ANN has also been applied successfully for machining problems in engineering [52, 53]. The architecture of ANN consists of three layers: input layer, hidden layer and the output layer. Each layer is characterised by the neuron/s. The number of neurons in the input and output layer is equal to number of input and output variables considered in the study. The hidden and output layer makes use of the activation function. The given layer is connected to pre- and after layer using links. Each link is characterised by a weight. Each weighted link is computed from the multiplication of input values and weights. The neurons in the hidden and output layer are offset by a threshold value.

Each weighted link outputs a value that is passed to each in neuron in the hidden layer. The summation of the weighted inputs is done in hidden layer and is further input to activation function (A) which produces an output O.

$$O = A\left(\sum_{i=0}^{n-1} w_i x_i - \beta\right) \quad (7)$$

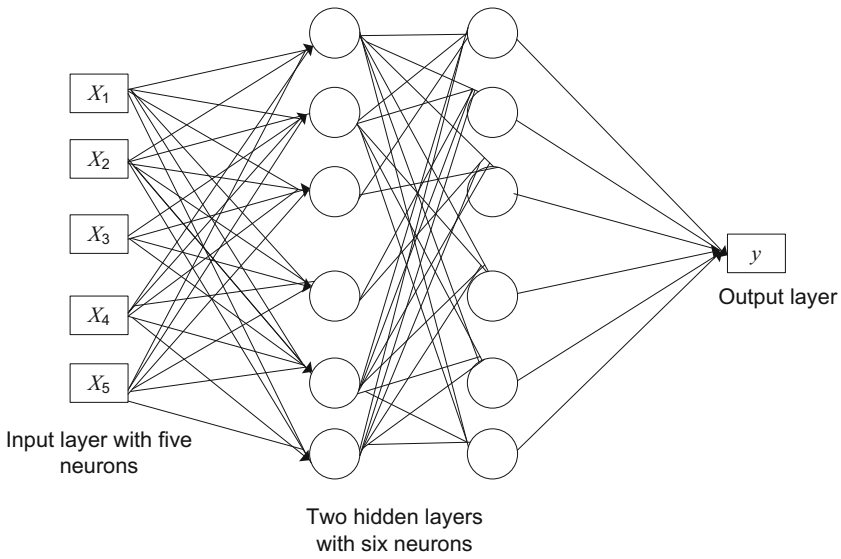

Fig. 3 Architecture of ANN determined based on a trial and error approach

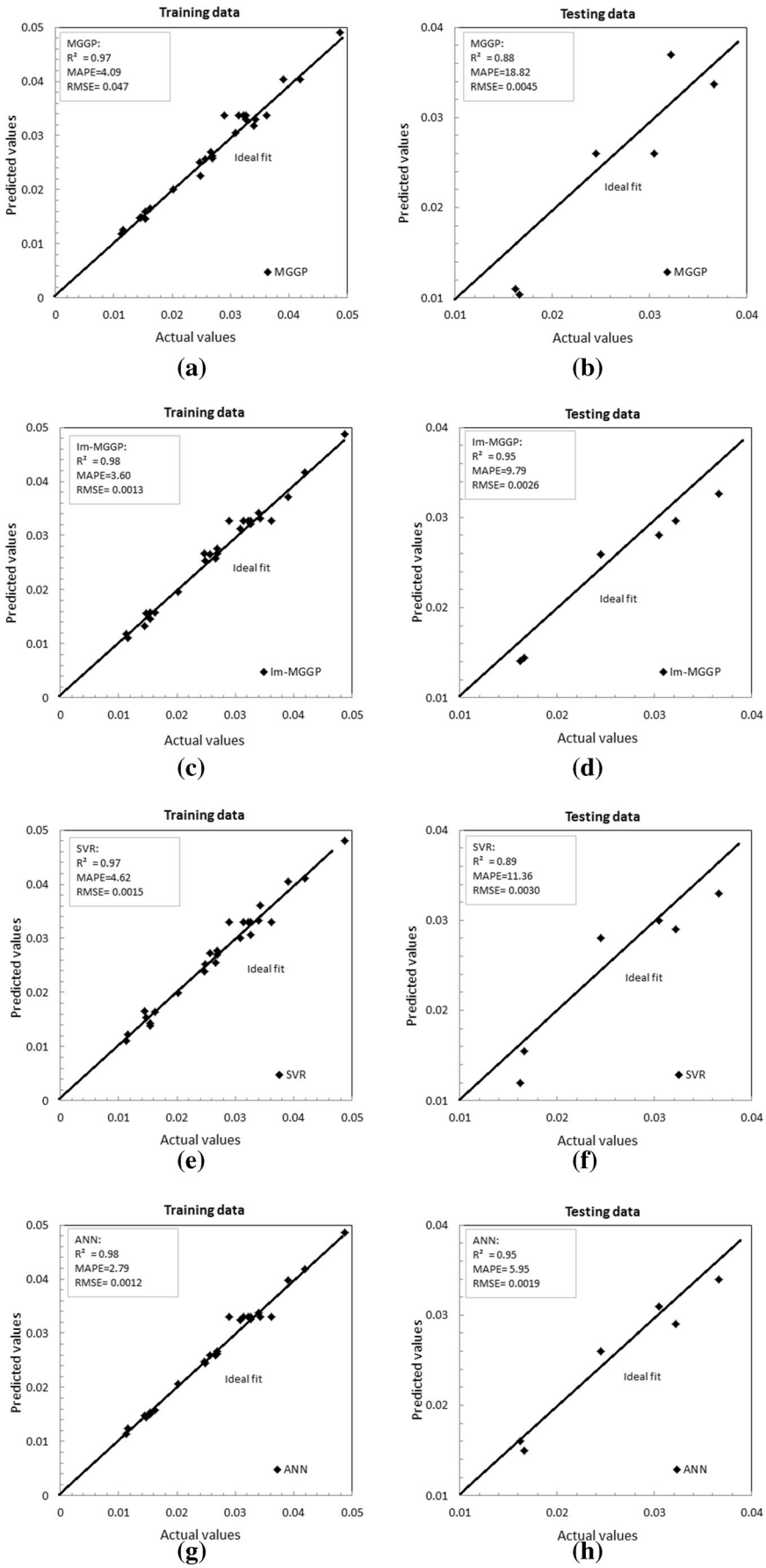

Fig. 4 Fit of the computational intelligence formulated models on training and testing data **a**, **b** MGGP; **c**, **d** Im-MGGP; **e**, **f** SVR; and **g**, **h** ANN

where w_i the weight on the links, x_i is the ith input variables and β is the threshold or offset of the neuron. In the present study, the sigmoid-logistic function used as activation function is given by

$$A(x) = \frac{1}{1+e^{-x}} \tag{8}$$

The error function used to estimate the deviation of the predicted value of the network from the experimental sample i is given by

$$E_{\text{sample}} = \frac{1}{2}\sum_{n=1}^{N}(A_i - M_i)^2 \tag{9}$$

where A_i and M_i are actual and predicted values for ith sample, respectively, and M is the number of neurons in the output layer. The average error for the whole network is given by

$$\text{Average error} = \frac{1}{2}\sum_{n=1}^{N}\sum_{n=1}^{N}(A_i - M_i)^2 \tag{10}$$

where N is the total number of samples. Back propagation algorithm, namely, Levenberg-Marquardt algorithm [54] that works on the principle of the second derivative is used to optimise the average error. The simpler form of Hessian matrix is used, and the algorithm iterates weights using formulae

$$x_{k+1} = x_k - \left[J^T j + \mu I\right]^{-1} J^T e \tag{11}$$

where J is the Jacobian matrix that consists of the first derivatives of the network errors, e is a vector of network errors, μ is the learning rate and I is the identity matrix. Weights in each link are estimated and updated until the threshold error is achieved. Threshold error fixed by the user is the maximum number of epochs or the minimum error of the model whichever is achieved earlier.

3.4.1 Settings of parameter for implementation of ANN

Three layer feed-forward neural networks is implemented in MATLAB R2010b. The settings chosen for ANN are shown in Table 3. Number of neurons in the hidden layer and number of hidden layers influence the generalisation ability of the ANN model, and hence, it is important to select them properly. In the present work, a trial and error approach is used to select the optimum number of neurons and number of hidden layers. As shown by bar graph in Fig. 2, for number of neurons, six, and hidden layers, two, the RMSE is minimum, and therefore the ANN model with two hidden layers with six neurons is selected. Architecture of the selected ANN network is shown in Fig. 3. The performance of the ANN model on training and testing data is discussed in Section 4.

4 Computation of models performance

The results obtained from the four models are illustrated in Fig. 4 on the training and testing data, respectively. Performance of the proposed models is evaluated using the five metrics: the coefficient of determination (R^2), the mean absolute percentage error (MAPE), the RMSE, the relative error (%) and multi-objective error function (MO) given by

$$R^2 = \left(\frac{\sum_{i=1}^{n}\left(A_i - \overline{A_i}\right)\left(M_i - \overline{M_i}\right)}{\sqrt{\sum_{i=1}^{n}\left(A_i - \overline{A_i}\right)^2 \sum_{i=1}^{n}\left(M_i - \overline{M_i}\right)^2}}\right)^2 \tag{12}$$

$$\text{MAPE}(\%) = \frac{1}{n}\sum_{i}\left|\frac{A_i - M_i}{A_i}\right| \times 100 \tag{13}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}|M_i - A_i|^2}{N}} \tag{14}$$

$$\text{Relative error}\,(\%) = \frac{|M_i - A_i|}{A_i} \times 100 \tag{15}$$

$$\text{Multi-objective error} = \frac{\text{MAPE} + \text{RMSE}}{R^2} \tag{16}$$

where M_i and A_i are the predicted and actual values, respectively, $\overline{M_i}$ and $\overline{A_i}$ are the average values of the predicted and actual, respectively, and n is the number of training samples. Since, the values of R^2 do not change by changing the models values equally and the functions MAPE and RMSE and the relative error only shows the error and no correlation. Therefore, a MO error function that is a combination of these metrics is also used.

Table 4 Multi-objective error of the four models

Models	Training data	Testing data
MGGP	4.26	21.43
Im-MGGP	3.67	10.30
SVR	4.76	12.76
ANN	2.84	6.26

Table 5 Descriptive statistics of the relative error (%) of the four models

Models	Count	Mean	LCI 95%	UCI 95%	Std dev	SE mean	Median	Maximum	Minimum
MGGP	32	6.85	3.80	9.90	8.45	1.49	4.00	37.25	0.39
Im-MGGP	32	4.76	3.34	6.19	3.95	0.70	3.24	13.34	0.17
SVR	32	5.89	3.91	7.86	5.47	0.96	4.27	25.97	0.33
ANN	32	3.38	2.13	4.62	3.44	0.60	2.29	13.99	0.009

The result of training phase shown in Fig. 4a, c, e and g indicates that all the four models have impressively learned the non-linear relationship between the input variables and wear strength with high correlation values and relatively low error values. The result of the testing phase shown in Fig. 4b, d, f and h indicates that the predictions obtained from the proposed Im-MGGP and ANN models are in good agreement with the experimental data with values of R^2 achieved is as high as 0.95. Between MGGP and SVR, SVR has shown better performance.

MO values of the four models are computed on the training and testing data as shown in Table 4. The descriptive statistics of the errors are shown in Table 5, which illustrates error mean, standard deviation (Std dev), standard error of mean (SE mean), lower confidence interval (LCI) of mean at 95 %, upper confidence interval (UCI) of mean at 95 %, median, maximum and minimum. The lower MO error values on the testing data and the lower values of range (UCI-LCI) of the confidence intervals of the proposed Im-MGGP and ANN models indicate that they are able to generalise the wear strength values satisfactory based on the variations input experimental conditions.

The goodness of fit of the four models is evaluated based on the hypothesis tests (Table 6). These are *t*-tests to determine the mean and *f*-tests for variance. For the *t*-tests and the *f*-tests, the *p* values of all four models is >0.05, so there is not enough evidence to conclude that the actual values and predicted values from these models differ. Therefore, all models have statistically satisfactory goodness of fit from the modelling point of view.

Thus, from the statistical comparison presented, it can be concluded that the proposed Im-MGGP model is able to capture the dynamics of the interactive effect of layer thickness, orientation, raster angle, raster width and air gap on the wear strength satisfactory. Its performance was found to be on par with ANN. Between MGGP and SVR, SVR has shown better performance.

Table 6 *p* values to evaluate goodness of fit of the four models

95 % CI	MGGP	Im-MGGP	SVR	ANN
Mean paired *t*-test	0.0614	0.3584	0.2612	0.4167
Variance *f*-test	0.4926	0.8653	0.5826	0.9316

5 Sensitivity and parametric analysis of the proposed model

Sensitivity and parametric analysis about the mean is conducted for the validation of the proposed Im-MGGP model. The sensitivity analysis (SA) percentage of the outputs to each input parameter is determined using the following formulas:

$$L_i = f_{\max}(x_i) - f_{\min}(x_i) \tag{17}$$

$$\mathrm{SA}_i = \frac{L_i}{\sum_{j=1}^{n} L_j} \times 100 \tag{18}$$

where $f_{\max}(x_i)$ and $f_{\min}(x_i)$ are the maximum and minimum of the predicted output over the *i*th input domain, respectively, where the other variables are equal to their mean values.

Table 7 shows the sensitivity results of input variables in the prediction of wear strength of FDM fabricated component. From Table 7, it is clear that the input variable, air gap has the highest impact on wear strength of component followed by raster width, orientation and layer thickness. The influence of raster angle on the wear strength of component was found to be minimum. This reveals that by regulating the air gap, a greatest variation in wear strength can be achieved. The parametric analysis provides a measure of the relative importance among the inputs of the Im-MGGP model and illustrates how the wear strength varies in response to the variation in input variables. On the formulated Im-MGGP model, the first input is varied between its mean±definite number of standard deviations and the wear strength is computed, while the other inputs are fixed at its mean value. This analysis is then repeated for the other inputs. Figure 5 displays the plots generated for each input variable and the wear strength. These

Table 7 Amount of impact of each input variable to the wear strength

Input variables	Relative contribution (%) to wear strength
Layer thickness	9.30
Orientation	18.6
Raster angle	2.32
Raster width	30.23
Air gap	39.53

plots reveal that, for example, the wear strength decreases with an increase in layer thickness and raster width, and increases with increase in air gap. Wear strength follows a parabolic non-linear relation with the orientation. Analysis complies well with the study conducted by Sood et al. [13], which validates the robustness of the model.

From Table 7 and Fig. 5, we can then select the optimal values of the input variables, which optimise the wear strength. In this way, our proposed model can be used to reveals insights on the phenomenon of impact of input process parameters on the wear strength of FDM fabricated component.

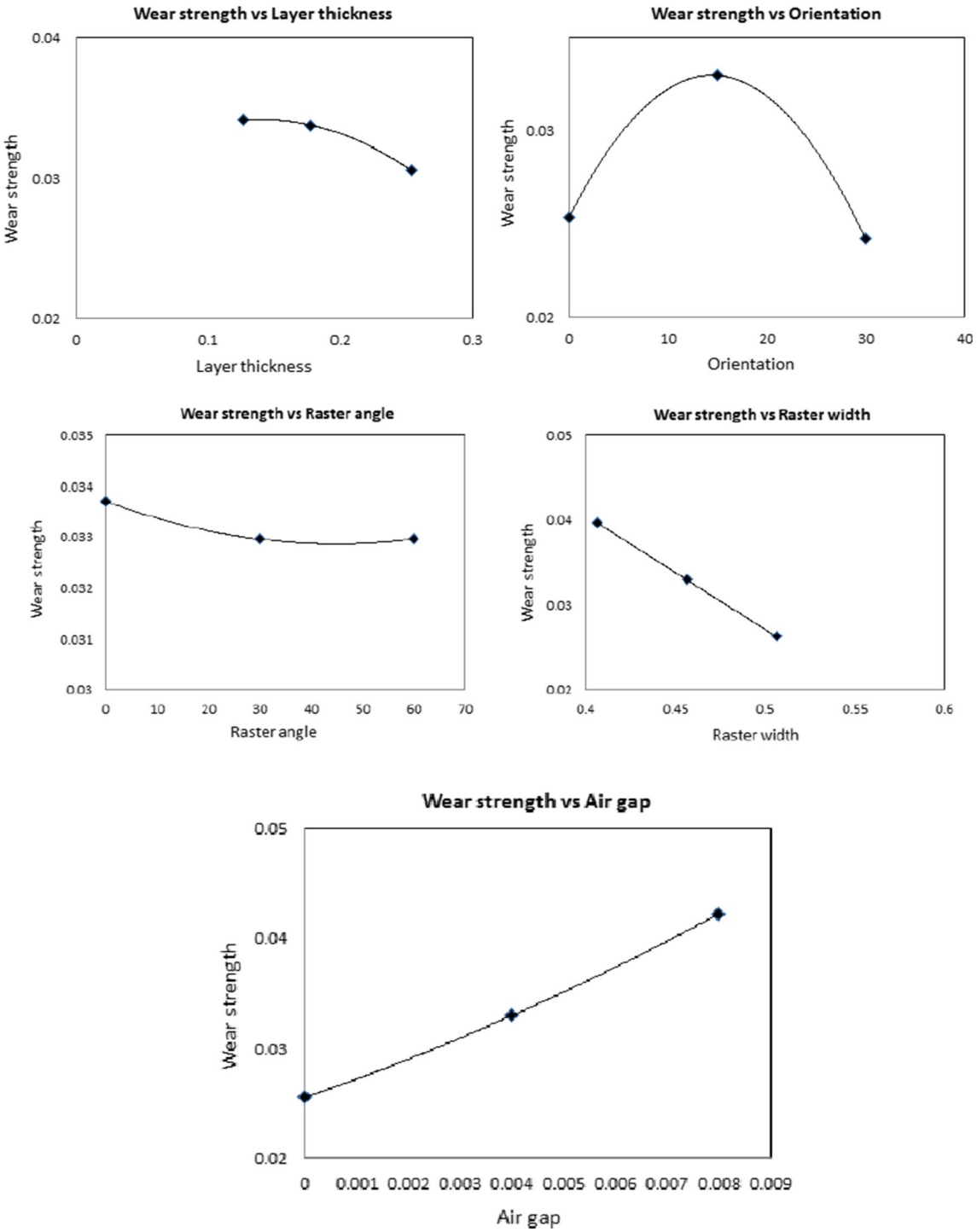

Fig. 5 Variation of wear strength with respect to each input variable

6 Conclusions

The present work establishes the motive behind formulating the functional relationships for studying the properties of FDM fabricated component. In addition, the issue of poor generalisation of MGGP approach is addressed. An improved approach of MGGP for the evaluation of wear strength of the FDM fabricated component is proposed. The proposed model outperforms the standardised MGGP and SVR models, and its performance is found to be at par with ANN model. The parametric and sensitivity analysis conducted validates the robustness of model by unveiling dominant input parameters and hidden non-linear relationships. It was found that the wear strength decreases with an increase in layer thickness and raster width and increases with increase in air gap. The high generalisation ability of the Im-MGGP model is beneficial for RP experts, who are currently looking for high-fidelity models that predict the wear strength in uncertain input process conditions. The model provides an explicit functional relationship between wear strength and the input process parameters, and thus can be used offline in shop floor for prediction. This model can also be further optimised and the optimal input process variables settings can be estimated to maximise the wear strength. Future work for authors is to investigate the environmental impacts of the 3-D printing FDM components using the improved evolutionary computational approach [55].

7 Appendix

$$\begin{aligned}\text{Wear strength}_{\text{MGGP}} = {} & 0.10772 + (0.10603) * \Big(\tan\Big(\big(((x4)-(x1))-(\tanh(x2)))* \\ & (\tan(\tanh(x1)))\Big)-((\exp(\exp(x1))) * ((\text{plog}(x3)) * (x5)))\Big)\Big) + (-0.0070485)* \\ & \Big(\tanh\Big(\big(((\text{plog}(x3))-(\tanh(x2))) * (\tan(\tanh(x1))))-\Big(\big(((-1.584267)) + (x4))* \\ & ((\text{plog}(x3)) * (\tan(x4)))\Big)\Big)\Big) + (0.00066149) * \Big(\tan\Big(\big(((x2)-(x1))-(\tanh(x2)))* \\ & (\tan(\tanh(x1)))\Big)-((\exp(\exp(x1))) * ((\text{plog}(x3)) * (\text{plog}(x3))))\Big)\Big) + (0.00044601)* \\ & \Big((\text{plog}((\exp(\exp(x1))) * (\tan((x1) * (x5))))) + \Big(\big(((x4)-(x1))-(\tanh(x2)))* \\ & (\tan(\tanh(x3)))\Big)-((\exp(\exp(x1))) * ((\tan(x2)) * (\tan(x4))))\Big)\Big) + (0.0031232)* \\ & ((\tan(\cos(\text{plog}(x5))))-((((6.831607)) * (x4)) * (\exp((2.068041))))) + (0.0077906)* \\ & (\text{plog}((\tanh(\tan(\tanh(x1)))) + (\tan((x2) + (x2))))) + (-1.1529)* \\ & \Big(((\tan(\cos(\text{plog}(x3))))-((((6.831607)) * (x4)) * (\exp(x1)))) * \Big((\text{plog}(x3))* \\ & (\tan((\tanh(x1)) * (x5)))\Big)\end{aligned} \tag{2}$$

$$\begin{aligned}\text{Wear strength}_{\text{Im-MGGP}} = {} & \\ & -0.021015 + (-0.022799) * \Big(((\exp(\tan(x2))) * (\cos(\cos(x1)))) + \Big((\exp(\exp(x4)))* \\ & (\tanh(\cos(x5)))\Big)\Big) + (-3.234\text{e}{-}006) * \Big(((\cos((x5) + (x1))) + (\tan(x5))) * \Big(\big((x2)* \\ & (x3)\Big)-((\tanh((x2) * (x3))) + ((6.898554)))\Big)\Big) + (0.16744) * \Big(\Big(\Big(\Big(((x1) + (x1))+ \\ & (x1)\Big) * (x5)\Big) + (\tan((x5)-(x5)))\Big)-((\sin(\cos(x2))) * (\tanh((x2) * (x5))))\Big)+ \\ & (0.066058) * \Big(\big(((\tan(x1)) * (((6.898554))-(\tan(x1)))) + (\tan(\exp((x5)-(x5)))))- \\ & (((\exp(\tan(x2))) * (\cos(\cos(x1)))) * (\tanh((x2) * (x5))))\Big) + (-0.0018666)* \\ & \Big(((\sin(\exp(\cos(x1)))) + (\cos(\cos((x1) + (x1))))) * \Big((\sin(\exp(\cos(x5)))) + \Big((x1)* \\ &)* (x2)\Big)\Big)\Big) + (-1.4071) * \Big(\Big(((\cos((x1) + (x1))) + (\exp((x5) * (x3))))-\Big(\sin\Big(\exp\Big((x1)+ \\ & (x1)\Big)\Big)\Big)\Big)\Big) * (\text{plog}((\cos((x5) + (x1))) + (\tan(x5))))\Big) + (0.6816) * \Big(\Big(\Big(\Big(\cos\Big((x1)+ \\ & (x1)\Big)\Big) + (\cos(x4))\Big)-(\sin(\exp(\cos(x1))))\Big) * (\text{plog}((\cos((x1) + (x1))) + (\tan(x5))))\Big)\end{aligned} \tag{4}$$

References

1. Mansour S, Hague R (2003) Impact of rapid manufacturing on design for manufacture for injection moulding. Proc Inst Mech Eng B J Eng Manuf 217(4):453–461
2. Hopkinson N, Hague R, Dickens P (2006) Rapid manufacturing, 217th edn. Wiley Online Library, Sussex, pp 453–461
3. Anitha R, Arunachalam S, Radhakrishnan P (2001) Critical parameters influencing the quality of prototypes in fused deposition modelling. J Mater Process Technol 118(1–3):385–388

4. Pandey PM, Reddy NV, Dhande SG (2003) Improvement of surface finish by staircase machining in fused deposition modeling. J Mater Process Technol 132(1–3):323–331
5. Masood SH, Song WQ (2004) Development of new metal/polymer materials for rapid tooling using fused deposition modelling. Mater Des 25(7):587–594
6. Stampfl J, Liska R (2005) New materials for rapid prototyping applications. Macromol Chem Phys 206(13):1253–1256
7. Zhong W et al (2001) Short fiber reinforced composites for fused deposition modeling. Mater Sci Eng A 301(2):125–130
8. Ahn D et al (2009) Representation of surface roughness in fused deposition modeling. J Mater Process Technol 209(15–16):5593–5600
9. Byun HS, Lee KH (2006) Determination of the optimal build direction for different rapid prototyping processes using multi-criterion decision making. Robot Comput Integr Manuf 22(1):69–80
10. Chang DY, Huang BH (2011) Studies on profile error and extruding aperture for the RP parts using the fused deposition modeling process. Int J Adv Manuf Technol 53(9–12):1027–1037
11. Li, A., et al. (2010) Optimization method to fabrication orientation of parts in fused deposition modeling rapid prototyping. in 2010 International Conference on Mechanic Automation and Control Engineering, MACE2010
12. Garg A, Rachmawati L, Tai K (2013) Classification-driven model selection approach of genetic programming in modelling of turning process. Int J Adv Manuf Technol 69(5–8):1137–1151
13. Sood AK et al (2012) An investigation on sliding wear of FDM built parts. CIRP J Manuf Sci Technol 5(1):48–54
14. Sood, A.K., R.K. Ohdar, and S.S. Mahapatra (2010) A hybrid ANN-BFOA approach for optimization of FDM process parameters, in lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) p. 396–403
15. Sood AK, Ohdar RK, Mahapatra SS (2010) Parametric appraisal of fused deposition modelling process using the grey Taguchi method. Proc Inst Mech Eng B J Eng Manuf 224(1):135–145
16. Vosniakos GC, Maroulis T, Pantelis D (2007) A method for optimizing process parameters in layer-based rapid prototyping. Proc Inst Mech Eng B J Eng Manuf 221(8):1329–1340
17. Cevik A, Guzelbey IH (2007) A soft computing based approach for the prediction of ultimate strength of metal plates in compression. Eng Struct 29(3):383–394
18. Cevik A, Sonebi M (2008) Modelling the performance of self-compacting SIFCON of cement slurries using genetic programming technique. Comput Concr 5(5):475–490
19. Gandomi AH et al (2010) Genetic programming and orthogonal least squares: a hybrid approach to modeling the compressive strength of CFRP-confined concrete cylinders. J Mech Mater Struct 5(5):735–753
20. Garg A, Tai K (2013) Comparison of statistical and machine learning methods in modelling of data with multicollinearity. Int J Model Identif Control 18(4):295–312
21. Gandomi AH, Alavi AH (2011) Multi-stage genetic programming: a new strategy to nonlinear system modeling. Inf Sci 181(23):5227–5239
22. Garg A, Tai K (2011) A hybrid genetic programming-artificial neural network approach for modeling of vibratory finishing process. Int Proceed Comput Sci Inform Technol (IPCSIT) 18:14–19
23. Garg, A. and K. Tai. (2012) Comparison of regression analysis, artificial neural network and genetic programming in handling the multicollinearity problem. In Proceedings of 2012 International Conference on Modelling, Identification and Control, ICMIC 2012
24. Garg, A., et al. (2014) Mathematical modelling of burr height of the drilling process using a statistical-based multi-gene genetic programming approach. The International Journal of Advanced Manufacturing Technology p. 1–14
25. Garg A et al (2014) An embedded simulation approach for modeling the thermal conductivity of 2D nanoscale material. Simul Model Pract Theory 44:1–13
26. Garg A et al (2014) Combined CI-MD approach in formulation of engineering moduli of single layer graphene sheet. Simul Model Pract Theory 48:93–111
27. Vijayaraghavan V. et al. (2014) An integrated computational approach for determining the elastic properties of boron nitride nanotubes. International Journal of Mechanics and Materials in Design p. 1–14
28. Chan KY et al (2011) Reducing overfitting in manufacturing process modeling using a backward elimination based genetic programming. Appl Soft Comput J 11(2):1648–1656
29. Gonçalves, I. et al. (2012) Random sampling technique for overfitting control in genetic programming, in lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) p. 218–229
30. Lee BH, Abdullah J, Khan ZA (2005) Optimization of rapid prototyping parameters for production of flexible ABS object. J Mater Process Technol 169(1):54–61
31. Sood AK, Ohdar RK, Mahapatra SS (2010) Parametric appraisal of mechanical property of fused deposition modelling processed parts. Mater Des 31(1):287–295
32. Kumar S, Kruth JP (2008) Wear performance of SLS/SLM materials. Adv Eng Mater 10(8):750–753
33. Koza, J.R. (1992) Genetic programming : on the programming of computers by means of natural selection. Vol. 1. MIT Press
34. Garg, A. and K. Tai. (2012) Review of genetic programming in modeling of machining processes. In Proceedings of 2012 international conference on modelling, identification and control, ICMIC 2012
35. Garg A et al (2014) A computational intelligence-based genetic programming approach for the simulation of soil water retention curves. Transp Porous Media 103(3):497–513
36. Garg A et al (2014) An integrated SRM-multi-gene genetic programming approach for prediction of factor of safety of 3-D soil nailed slopes. Eng Appl Artif Intell 30:30–40
37. Garg, A. and K. Tai (2013) Genetic programming for modeling vibratory finishing process: role of experimental designs and fitness functions, In Swarm, Evolutionary, and Memetic Computing. Springer international publishing. p. 23–31
38. Garg A, Tai K, Gupta A (2014) A modified multi-gene genetic programming approach for modelling true stress of dynamic strain aging regime of austenitic stainless steel 304. Meccanica 49(5):1193–1209
39. Garg, A., K. Tai, and M. Savalani (2014) Formulation of bead width model of an SLM prototype using modified multi-gene genetic programming approach. Int J Advanced Manufact Technol p. 1–14
40. Searson, D.P., D.E. Leahy, and M.J. Willis. (2010) GPTIPS: an open source genetic programming toolbox for multigene symbolic regression. In Proceedings of the International Multiconference of Engineers and Computer Scientists Citeseer
41. Searson D, Willis M, Montague G (2007) Co-evolution of non-linear PLS model components. J Chemom 21(12):592–603
42. Hinchliffe MP, Willis MJ (2003) Dynamic systems modelling using genetic programming. Comput Chem Eng 27(12):1841–1854
43. Byvatov E, Schneider G (2003) Support vector machine applications in bioinformatics. Appl Bioinform 2(2):67–77
44. Hearst MA et al (1998) Support vector machines. Intell Syst Applic IEEE 13(4):18–28
45. Kecman V (2001) Learning and soft computing: support vector machines, neural networks, and fuzzy logic models. MIT Press, Cambridge
46. Pelckmans K., et al. LS-SVMlab: a MATLAB/C toolbox for least squares support vector machines
47. Salgado DR, Alonso FJ (2007) An approach based on current and sound signals for in-process tool wear monitoring. Int J Mach Tools Manuf 47(14):2140–2152
48. Salgado DR et al (2009) In-process surface roughness prediction system using cutting vibrations in turning. Int J Adv Manuf Technol 43(1–2):40–51

49. Saptoro A., M.O. Tadé, and H. Vuthaluru (2012) A modified Kennard-Stone algorithm for optimal division of data for developing artificial neural network models. Chemical Product and Process Modeling, 7(1)
50. Li HY et al (2012) Artificial neural network and constitutive equations to predict the hot deformation behavior of modified 2.25Cr-1Mo steel. Mater Des 42:192–197
51. Ramanan TR et al (2011) An artificial neural network based heuristic for flow shop scheduling problems. J Intell Manuf 22(2):279–288
52. Zhao D et al (2014) Synchronized control with neuro-agents for leader-follower based multiple robotic manipulators. Neurocomputing 124:149–161
53. Zhao D, Ni W, Zhu Q (2014) A framework of neural networks based consensus control for multiple robotic manipulators. Neurocomputing 140:8–18
54. Cherkassky V, Ma Y (2003) Comparison of model selection for regression. Neural Comput 15(7):1691–1714
55. Lee CKM, Lam JSL (2012) Managing reverse logistics to enhance sustainability of industrial marketing, Ind Mark Manag 41(4):589–598. doi:10.1016/j.indmarman.2012.04.006