The International Journal of Advanced Manufacturing Technology (2022) 122:2701–2717
https://doi.org/10.1007/s00170-022-10061-1

ORIGINAL ARTICLE

Optimization of fixture locating layout design using comprehensive optimized machine learning

Mohammad Reza Chalak Qazani[1] · Hadi Parvaz[2] · Siamak Pedrammehr[3]

Received: 25 March 2022 / Accepted: 30 August 2022 / Published online: 8 September 2022

Abstract

Fixtures are commonly employed in production as work holding devices that keep the workpiece immobilized while machined. The workpiece's deformation, which affects machining precision, is greatly influenced by the positioning of fixture elements around the workpiece. By positioning the locators and clamps appropriately, the workpiece's deformation might be decreased. Therefore, it is required to model the fixture–workpiece system's complicated behavioral relationship. In this study, long short-term memory (LSTM), multilayer perception (MLP), and adaptive neuro-fuzzy inference system (ANFIS) are three machine-learning approaches employed to model the connection between locator and clamp positions and maximum workpiece deformation throughout end milling. The hyperparameters of the developed ANFIS, MLP, and LSTM are chosen using the evolutionary algorithms, including genetic algorithm (GA), particle swarm optimization (PSO), butterfly optimization algorithm (BOA), grey wolf optimization (GWO), and wolf optimization algorithm (WOA). Among developed methods, MLP optimized using BOA (BOA-MLP) reached the highest accuracy among all developed models in predicting the response surface. The developed model had a lower computational load than the final element model in calculating the response surface during the machining process. At the final step, the prementioned five evolutionary algorithms were implemented in the developed BOA-MLP to extract the optimal parameters of the fixture to decrease the deflection of the workpiece throughout the machining. The proposed method was modeled in MATLAB. The outcomes showed that the mentioned model was efficient enough compared with the previous method, such as optimized response surface methodology in the point view of 0.0441 μm lower workpiece deflection.

Keywords Fixture design · Finite-element method · Adaptive neuro-fuzzy inference system · Multilayer perceptron · Long short-term memory · Evolutionary algorithms

Nomenclature

Symbols

ANFIS	Adaptive neuro-fuzzy inference system
BOA	Butterfly optimization algorithm
BPTT	Backpropagation through time
CC	Correlation coefficient
CEC	Constant error carousel
FEM	Finite element method
FFNN	Feedforward neural network
FIS	Fuzzy inference system
GA	Genetic algorithm
GWO	Grey wolf optimization
LSTM	Long-short term memory
MLP	Multilayer perception
MSE	Mean square error
NRMSE	Normalized root mean square error
PSO	Particle swarm optimization
RMSE	Root mean square error
RSM	Response surface methodology
RTRL	Real-time recurrent learning
WOA	Wolf optimization algorithm

✉ Mohammad Reza Chalak Qazani
m.r.chalakqazani@gmail.com

Hadi Parvaz
h.parvaz@shahroodut.ac.ir

Siamak Pedrammehr
s.pedrammehr@gmail.com

[1] Institute for Intelligent Systems Research and Innovation, Deakin University, Geelong, VIC 3216, Australia

[2] Faculty of Mechanical and Mechatronics Engineering, Shahrood University of Technology, P.O. Box 3619995161, Shahrood, Iran

[3] Faculty of Mechanical Engineering, University of Tabriz, Tabriz, Iran

Acronyms

a_1, a_2, a_3, b_1, b_2, and b_3	Linear gains of fuzzy Rules 1 and 2
A_i and B_i	Fuzzy sets
c, k, a, and I	The sensory modality, the fragrance's perceived magnitude and observed power exponent, magnitude reliant on stimulus intensity, and modality in BOA
c_t^j	The memory cell of the j^{th} neuron at time t in LSTM
$\widetilde{c}_t^j$	The new memory value of the j^{th} neuron at time t in LSTM
c_t	The vector representations of c_t^j
C_1 and C_2	Clamp position
d	Targets of MLP
E	Young's modulus
f_t^j	The forget gate of the j^{th} neuron at time t in LSTM
F_{c1}	First clamp force
F_{c2}	Second clamp force
g	Centered logistic sigmoid function
$h_j(n)$	The j^{th} output node in the n^{th} data point of MLP
J and K	Coefficients of WOA
L	The distance between the prey and the best agent
m_t^j	The output of the j^{th} neuron in LSTM
m_{t-1}	The vector representations of m_{t-1}^j
$\mathrm{m} = \left(m_1, m_2, \cdots, m_T\right)$	The output sequence of LSTM
n_{x_i}	The i^{th} normalized input data
o_t^j	The output gate of LSTM
p_i and P^{best}	Position matrix for the i^{th} whale with the number of search agents and the best search agent's position vector
Q_i^j	The i^{th} node in the j^{th} ANFIS layer output
s_i	The i^{th} local induced field of MLP
t	Iteration number
T	The prediction period of the LSTM
v and u	Inputs of fuzzy rule
w_{i}	The i^{th} rule's firing strength
$\overline{w}_i$	Average of the i^{th} rule's firing strength
W_f, U_f, and V_f	The diagonal weight matrices of LSTM in forget gate
W_i, U_i, and V_i	The diagonal weight matrices of LSTM in input gate
W_o, U_o, and V_o	The diagonal weight matrices of LSTM in output gate
$\widetilde{x}$	Average deviation of the dataset
$\underline{x}$ and $\overline{x}$	Functions for extracting the minimum and maximum values of the data
x_i	The i^{th} raw dataset (combination of input and output)
$\mathrm{x} = \left(x_1, x_2, \cdots, x_T\right)$	Input sequence of LSTM
y	Actual outputs of MLP
y_j	The previous neuron's output
z_1 and z_2	Outputs of fuzzy Rules 1 and 2
$\Delta w_{ij}(n)$	The variation of weights in the i^{th} input and j^{th} output node in the n^{th} data point of MLP
ε	The weights of MLP
η	The learning rate of MLP
μ_x and μ_y	Membership degrees
Υ	Poisson's ratio
σ_x	Functions for extracting the standard
σ_{x_i}	The i^{th} standardized input data
σ	Standard logistic sigmoid function
$\varnothing'$	The derivative of the activation function

1 Introduction

Machining fixtures are an inescapable part of the work holding devices since they precisely locate and restrict the workpiece throughout the machining process [1]. Optimized machining fixture layout refers to properly selecting the position and orientation of locators and clamps around the workpiece to reduce its deformation throughout machining. Designing and optimizing fixture layouts have a distinctive role in production since they increase production rates and reach a higher quality standard [2]. However, factors including fixture component position, clamping force magnitude, and cutting force are inevitable in determining the optimum machining fixture layout. On the other hand, clamping and cutting forces must be implemented for immobilization and machining of the workpiece accordingly. They cannot be reduced any farther than a particular limit. As a result, workpiece deformation is unavoidable throughout machining, which may result in a defective workpiece. Identifying the proper location for the fixture elements, on the other hand, will reduce workpiece deformation, which would assist in machining accuracy.

While machining fixtures play a significant role in production, designers struggle to optimize the layout of the fixture [3]. Traditionally, fixture layout optimization has depended on the designer's knowledge and ability, which is expensive and laborious [4, 5]. The machining inaccuracy and the fixture arrangement do not have a direct analytical connection [6, 7]. The complex fixture–workpiece interactions were investigated using numerical approaches, including the finite-element method (FEM) [8, 9]. Edward [10] created a quick support layout optimization model using FEM to determine the stiffness of the workpiece and a set of tests to validate the model. Li and Melkote [11] suggested an optimization approach for fixture layout. The resultant nonlinear program was solved using Zoutendjik's technique of possible directions, and the ultimate enhanced solution depends on the initial viable solution. Hamedi [12] proposed the hybrid FFNN and GA to extract the optimal fixture layout. He used the dataset generated via FEM for training the FFNN. Padmanaban et al. [13] demonstrated an ant colony algorithm-based discrete and continuous technique for optimizing the layout of the fixture. Selvakumar et al. [14] used FFNN to estimate the deformation of the workpiece during milling process based on the regenerated data using FEM and Taguchi design experiment. Sundararaman et al. [15] modeled the relations between locator and clamp position to calculate the maximum deformation of the workpiece using response surface methodology (RSM). Afterwards, optimization techniques such as sequential approximation and LINGO were used. Xing et al. [16] optimized the fixture design for complex auto-body components using a non-domination sorting social radiation method. Lu and Zhao [17] used feed-forward neural network (FFNN) and GA to optimize the layout of the sheet metal workpiece fixture utilizing the 4–2-1 locating scheme. The FFNN is employed to estimate the deformation of sheet metal workpieces under various layouts of fixtures and their induced locating error. Lu utilizes a GA and Zhao [17] to identify the optimal location for the fourth fixture locator relying on the FFNN estimation model.

The usage of machine-learning methods and evolutionary algorithms to extract the optimal fixture layout has been focused by lots of researchers in recent years to extract the optimal fixture layout. Low et al. [18] used reinforcement learning method to design the fixture layout fully automatic. Higher computational time and lack of accuracy were the disadvantages of their proposed method. Wu et al. [19] used genetic algorithm to extract the optimal position of the clamping points in blade fixture. As they predict the deformation of the blade during the milling process via FEM, the higher computational load and limited time of evaluation were their proposed method disadvantages. Rezaei Aderiani et al. [20] extract the optimal fixture layout for compliant sheet metal assemblies using GA and FEM. The higher computational load and limited times of evaluation were their model limitations; same as Wu's model [19]. Wu et al. [21] used the hybrid model of FEM and GA to extract the optimal layout of the fixture in end-milling process of the fan blade. The same methodology has been followed by Butt et al. [22] to extract the optimal fixture layout in milling operation. Du et al. [23] used simulated annealing algorithm (SAA) as a global optimization method instead of evolutionary algorithm in order to decrease the number of trials in extracting the optimal solutions. However, they still used FEM for calculation of the workpiece deformation during the milling process, which is very time consuming. Vinosh et al. [24] followed the same methodology of previous studies, which is based on the combination of evolutionary algorithm and FEM. However, they [24] used Taguchi methods for designing of experiments in order to reduce the number of trials. Michael Thomas Rex et al. [25] used the FEM and mixed discrete-integer GA to decrease the deformation of the prismatic workpiece in the pocket milling operation. Alshameri et al. [26] used SAA; same as Du et al. [23] for calculation of the set of points for designing the robust fixture. Hu [27] used hidden Markov model for optimization of digital twin-driven reconfigurable fixturing in aviation industry. Yu and Wang [28] used machine-learning method to extract the model that can imitate the same process using RSM and mathematical optimization. Li et al. [29] used whale optimization algorithm to reduce the clamping deformation of the curved thin-walled parts. Feng et al. [30] proposed the recent machine learning–based prediction model of the workpiece deformation calculator using extreme gradient boosting (XGBoost) method.

Based on reviewed articles, the previous works can be divided into two groups. The first group of the work is established based on the combination of optimization method (evolutionary, global, mathematic) with FEM [10, 11, 13, 16, 19–27, 29]. These studies are accurate in calculation of the workpiece deformation during the machining operation. However, the FEM is computationally expensive, and each trial takes a couple of hours. Then, optimizer should reach the optimal results using the limited number of trials. On the other hand, the second group of studies are based on the combination of machine learning developed model and optimization technique [15, 17, 18, 28, 30]. Using these developed methods based on machine learning technique, the recalculation of the workpiece deformation can be achieved in couple of seconds. It should be noted that the idea of combining machine learning and optimization (second group) is limited in using some basic machine learning techniques including RSM, FFNN, and XGBoost. However, the comprehensive study with evaluations of many machine-learning methods and evolutionary algorithms is not investigated in this field.

The main contribution of this study is to fill this gap in the optimal designing of the optimal machining fixture. Cost-effective modeling must determine the connection between fixture components and the maximum workpiece deformation. Furthermore, the evolutionary technique's implementation will be streamlined to minimize the complexity and computing time of the machining fixture layout without compromising solution quality. As a result, multilayer perceptron (MLP), long short-term memory (LSTM), and adaptive neuro-fuzzy inference system (ANFIS) are employed to predict the deformation of the workpiece based on the different 3–2-1 locator's position of the machining fixture. The hyperparameters of the developed ANFIS, MLP, and LSTM models are chosen via the evolutionary algorithms, including particle swarm optimization (PSO), GA, grey wolf optimization (GWO), butterfly optimization algorithm (BOA), and whale optimization algorithm (WOA). The proposed optimized machine-learning methods are compared with each other, and the best candidate is selected as a proper representer for calculating the workpiece elastic deformation during the machining process. In addition, the selected model is employed in evolutionary algorithms to extract the optimal 3–2-1 locator's position of the machining fixture to achieve the minimum workpiece deformation.

In the next section, the design of the fixture layout with the addition of FEM is explained in detail. Section 3 explains the proposed methodology based on the regenerated data using the FEM model. It includes the explanation of ANFIS, MLP, LSTM, and evolutionary algorithms. Section 4 describes the implementation of the proposed method in MATLAB software to prove the efficiency of the proposed method compared to the previous method. The conclusion is remarked in Sect. 5.

2 Fixture layout and finite element simulation

The position of fixture elements has a substantial impact on workpiece deformation throughout machining and clamp actuation. As a result, it is critical to identify the proper position for the locators and clamps to reduce the workpiece's maximum deformation.

The model proposed by Sundararaman et al. [31] was used in the present study to evaluate the capabilities of the developed optimization methods in calculating the most proper positions of the locating and clamping agents in the fixture. In the mentioned research, a 2D simulation was performed using finite element analysis software to calculate the machining forces acting on a deformable part. The workpiece in the simulation was a 300 × 90 mm rectangular sheet with a thickness of 10 mm. The fixture in the present study is a 2D machining fixture with a 2–1 locating layout. This specific layout needs two clamping forces to push the workpiece toward the locators. The direction of the tool axis is perpendicular to the plane of the workpiece directed inward. In other words, the represented tool in Fig. 1 is the cross-section of the 3D endmill tool. It rotates clockwise to apply the two machining force components to the workpiece. The workpiece was steel with the modulus of elasticity and Poisson's ratio equal to 206 GPa and 0.3, respectively. At least two clamping forces must also act on the opposite edges of the locators so that the workpiece stays stable during machining. The chip removal effect is accounted for through the element death method [31]. The displacement of the clamping and locating points L_1, L_2, and C_2 along the y-axis is zero, whereas L_3 and C_1 along the x-axis are zero. The machining forces were calculated in two dimensions through performing simulations. The mentioned method

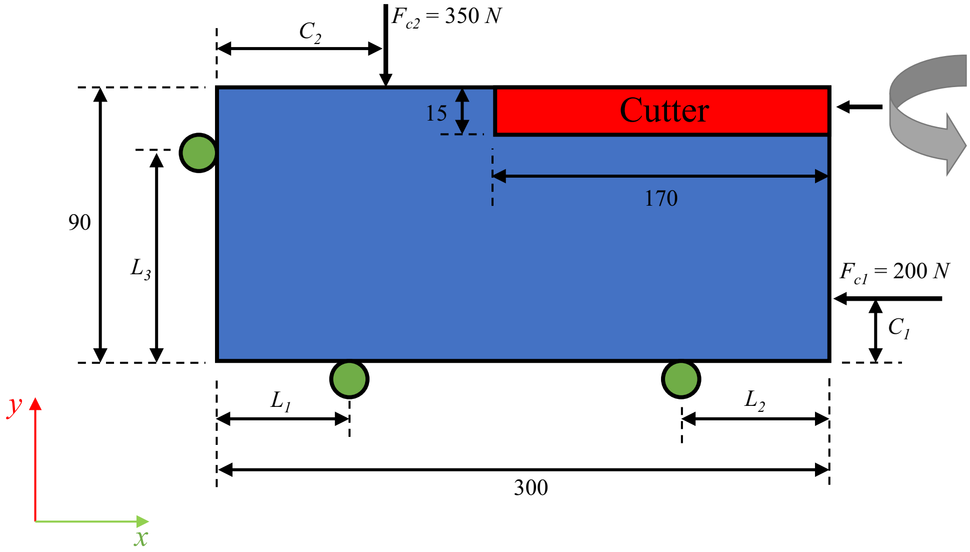


Fig. 1 The structure of the fixture layout in the end-milling process

suffers from several shortcomings. First, it does not utilize the experimentally validated machining force models to calculate the three-dimensional machining loads and torque. Second, even in the 2D simulation, the machining torque was not considered in the model, which strongly affects the locating layout. Finally, it would have been better if the machining model had been considered in three dimensions closer to the actual machining processes. The authors have identified these deficiencies, and comprehensive research is underway to address these deficiencies, which will be reported in future articles. The optimization parameters include the position of the locators and clamps' positions concerning the workpiece vertices. The objective function is to minimize the maximum amount of elastic deformation of the workpiece under the clamping and machining forces. The clamping forces were considered as $F_{c1} = 200\mathbf{N}$ and $F_{c2} = 350\mathbf{N}$ for the end-milling process. The machining forces were calculated as 100 N and 286 N in the x- and y-directions, respectively.

The clamp C_2 and locator L_1 are horizontally measured from the left top and left bottom corner of the workpiece. In contrast, clamp C_1 and locator L_3 are vertically measured from the workpiece's right bottom and left bottom corners. In addition, as illustrated in Fig. 1, the locator's position L_2 is determined by horizontally measuring from the right bottom corner. A horizontal step milling operation is conducted on the part. An endmill is used to perform this machining process. To convert 3D to 2D machining, it was assumed that the cutting flutes of the tool are parallel to the tool axis such that no axial force was applied to the workpiece. Therefore, only two cutting force components in radial and tangential directions were applied to the workpiece (Fig. 2). Depending on convergence analysis, the machining route is discredited into 11 load steps, and the maximum deformation for each load step is obtained. The required result for a single tool path at the final milling operation is the biggest eleven. The geometry of the 2D workpiece was being discretized into 42 quadrilateral plane elements, containing four nodes, each of which has two degrees of freedom. Figure 3 depicts the FEM model of locator, clamp, and machining forces at various load stages. Locators with positive reaction forces keep the workpiece connected with all the locators throughout machining, while those with opposing reaction forces induce the workpiece to be decoupled from the locators. External forces are imposed via clamps, and the limitation of positive reaction forces at the locators during machining is guaranteed.

All experiments in the study were conducted using ANSYS and MATLAB. The random location for the clamps and locators in the required range was created in a MATLAB m-file to conduct exploratory tests. The maximum deformation of the workpiece is computed while considering the limitations and saved as an output text file. The design variables and the maximum deformation of the workpiece are subsequently coded under MATLAB.

3 Methodology

This paper aims to estimate the workpiece's elastic deformation throughout machining inside the planer fixture with a 2–1 locating scheme. Figure 4 shows the schematic proposed method in the current research as the main contribution, which is the combination of machine-learning methods including ANFIS, MLP, and LSTM with evolutionary algorithms including GA, PSO, GWO, BOA, and WOA. According to Fig. 4, the process starts with selecting different locating and clamping points. Then, the FEM using ANSYS software is designed and simulated based on these predefined parameters. After simulation, the maximum deformation of the workpiece is captured in the FEM environment to be used in the proposed machine-learning methods. Then, three machine-learning methods, as the most common methods in this field (ANFIS, MLP and LSTM), are chosen to be trained based on the calculated data on FEM software. It should be noted that the evolutionary

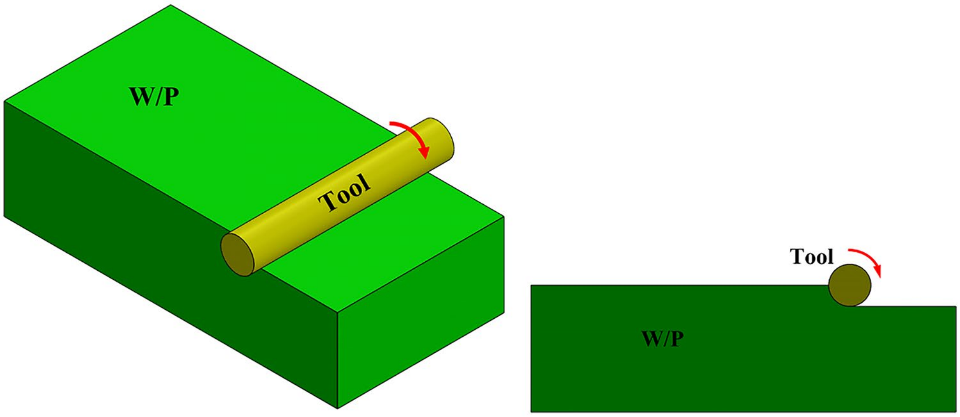


Fig. 2 The 3D and projected 2D end-milling machining process

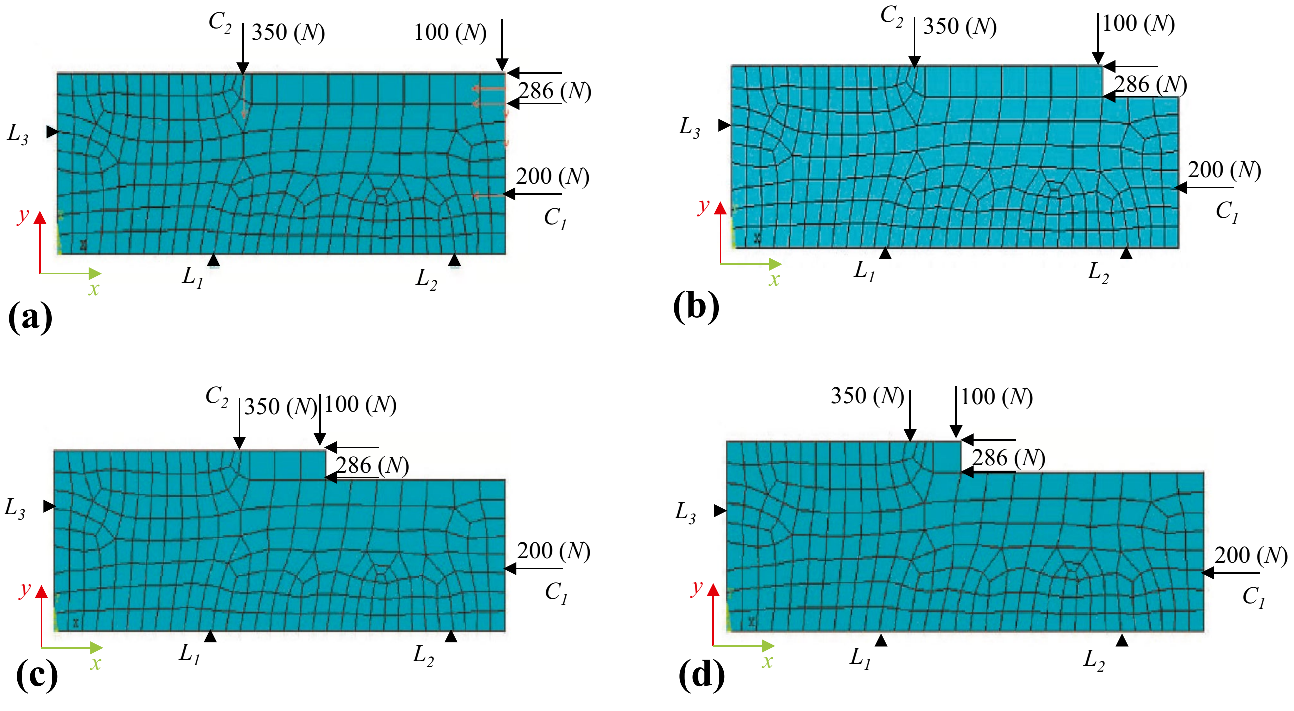


Fig. 3 Four different load steps' finite element mesh for end-milling process. **a** Load step 1, **b** load step 4, **c** load step 8, **d** load step 10 [31]

algorithms are used to extract the optimal hyperparameters of the developed machine-learning methods and extract the optimal fixture locating and clamping positions using developed machine-learning methods.

3.1 Data recording

The design parameter range should be as narrow as possible. It is due to the optimization point of the method to increase

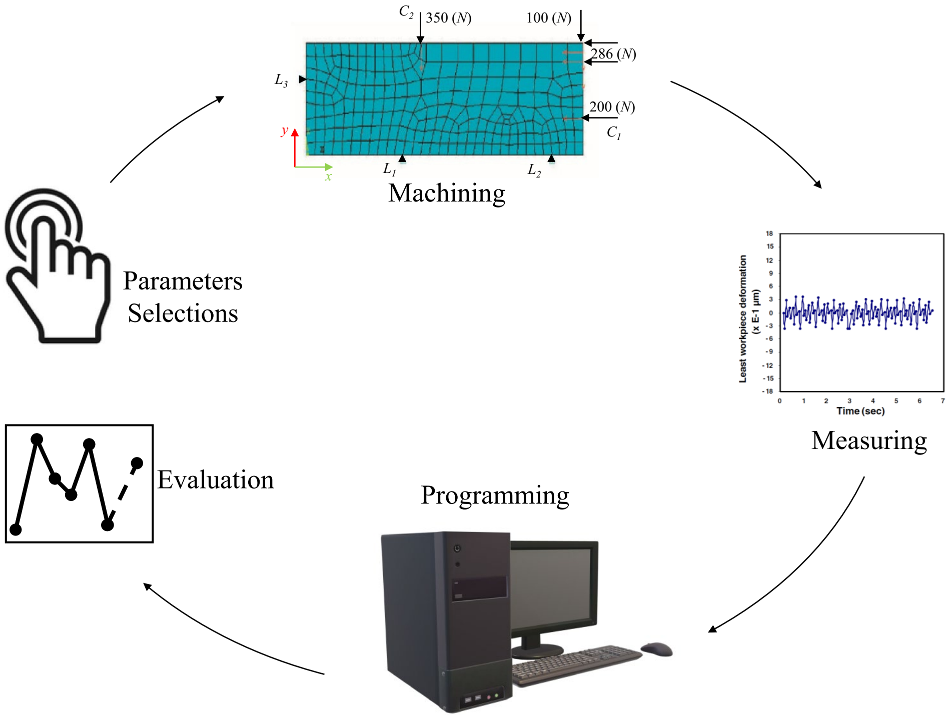


Fig. 4 Schematic structure of the proposed method using comprehensive machine-learning methods and evolutionary algorithms

the convergence speed of the evolutionary algorithm. In addition, narrowing the search area of evolutionary optimization algorithms can decrease the chance of local optimal point problem. The areas with a significant impact on the response of maximum deformation of the workpiece should be identified reasonably. The initial tests are designed and executed using a random and systematic search strategy to discover the best area with the least amount of workpiece deformation and determine the possible range for the design indices. Random trials consisting of ten sets with experiments of ten, twenty, thirty, and one hundred were prepared and performed in reference [31] for this configuration.

The most significant workpiece deformation was calculated for each trial. Table 1 shows the maximum workpiece deformations as a function of the design parameters. The connection between fixture element position and the maximum deformation of the workpiece is simulated in the area with the most significant potential to discover the optimum solution. Table 1 indicates that the associated locator and clamp position values for the two successive minimums of maximum workpiece deformation offer the ideal potential range for the design parameters. In the design input parameter L_1, the range is set initially at 5 to 148 mm. This generates the optimum outcome and goes narrower because of the initial trials, ranging from 105.1 to 119.4 mm. Likewise, ranges are generated for additional design input parameters. The experiment plan used is a second-order rotatable design with 10 star points, 10 center points, and 32 factorial points (52 experimental situations). As mentioned before, the maximum deformation of the workpiece is determined for the appropriate number of tests using FEM simulation, and the related variables of design are saved in MATLAB. The recorded variables of design and the APDL input batch file are transmitted to ANSYS to estimate the maximum deformation of the workpiece.

3.2 Data pre-processing

Three tasks should be implemented in the data before developing the model and applying the dataset. Initially, the data out of range should be removed to increase the robustness of the system. The second step is related to the normalization or standardization of the system to decrease the data complexity for a system before training. In this work, both methods are employed to decrease the complexity of the network's input data and increase the system's accuracy. The following formula is used to calculate the data standardization:

$$\sigma_{x_i} = \frac{x_i - \widetilde{x}}{\sigma_x} \tag{1}$$

where x_i and σ_{x_i} are respectively the i^{th} raw and standardized dataset. Dataset is defined as the combination of input data (fixturing and clamping position) and output data (elastic deformation of the workpiece during the end-milling process). σ_x and $\widetilde{x}$ are also the functions for extracting the standard and average deviation of the data.

The normalized data can also be obtained as

$$n_{x_i} = \frac{x_i - \underline{x}}{\overline{x} - \underline{x}} \tag{2}$$

where n_{x_i} is the i^{th} normalized dataset. $\underline{x}$ and $\overline{x}$ are also the functions for extracting the minimum and maximum values of the data. At the final network pre-tuning process, the data is divided into 86.54% and 13.46% for the testing and training process of the network. The testing data is not shown to the system until the testing stage of the network in order to reach realistic results via the proposed models.

Three models, including ANFIS, MLP, and LSTM, were utilized in this paper to estimate the elastic workpiece deformation during machining based on the locating and clamping positions of the fixture.

3.3 Adaptive neuro-fuzzy inference system

Being a Takagi–Sugeno-type artificial intelligence model used to solve nonlinear and complicated problems, ANFIS is a hybrid system composed of an artificial NN and a fuzzy inference system (FIS) [32]. This implies that ANFIS is capable of self-learning and reasoning operation. ANFIS's fuzzy rules are as below:

Rule 1: if v is O_1, and u is P_1 then $z_1 = a_1 v + a_2 u + a_3$.

Rule 2: if v is O_2, and u is P_2 then $z_2 = b_1 v + b_2 u + b_3$.

In Rules 1 and 2, the variables v and u are inputs. The outputs z_1 and z_2 are specified by Rules 1 and 2. The parameters a_1, a_2, a_3, b_1, b_2, and b_3 are acquired during the learning process. As seen in Fig. 5, the ANFIS structure comprises five layers, each with one output and two inputs.

Each layer has the following meaning:

The first layer is a layer of fuzzification that employs the functions membership in computing fuzzy clusters according to the training input data. The indices a_i and b_i are utilized to establish the membership function and calculate the degree of membership as

$$Q_i^1 = \mu_{A_i}(u) = \frac{1}{1 + \left|\frac{u-c}{a}\right|^{2b}};\ i = 1,2 \tag{3}$$

$$Q_i^1 = \mu_{B_i}(u);\ i = 1,2 \tag{4}$$

in which membership degrees are μ_u and μ_y. Fuzzy sets are A_i and B_i, and x is the input, whereas Q_i^j is the i^{th} node in the j^{th} layer output.

Table 1 The maximum deformation of the workpiece and experimental design matrix is estimated by FEM [31]

Exp. no	L_1 (mm)	L_2 (mm)	L_3 (mm)	C_1 (mm)	C_2 (mm)	FEM simulation (μm)	Exp. no	L_1 (mm)	L_2 (mm)	L_3 (mm)	C_1 (mm)	C_2 (mm)	FEM simulation (μm)
1	119.40	33.6	53	23	113	25.5491	27	112.25	26.45	57	26	119	25.64277
2	112.25	26.45	57	18.86	119	25.43024	28	105.1	19.3	53	23	113	27.43631
3	119.4	33.6	53	29	125	25.95754	29	112.25	26.45	57	26	119	25.64277
4	112.25	43.46	57	26	119	30.28401	30	105.1	19.3	61	23	125	29.65783
5	112.25	26.45	57	26	119	25.64277	31	112.25	26.45	57	26	119	25.64277
6	105.1	33.6	53	29	125	26.65154	32	119.4	33.6	53	23	125	25.99496
7	112.25	26.45	47.49	26	119	24.74045	33	119.4	33.6	53	29	113	25.72552
8	119.4	33.6	61	29	125	25.96203	34	119.4	33.6	61	23	113	25.9717
9	105.1	19.3	53	29	113	27.78806	35	112.25	26.45	57	26	119	25.64277
10	119.4	19.3	61	23	125	26.07392	36	112.25	26.45	57	26	119	25.64277
11	105.1	33.6	53	23	125	26.35655	37	105.1	33.6	61	23	113	26.11597
12	105.1	19.3	61	29	125	29.9249	38	119.4	19.3	53	23	113	24.15352
13	119.4	19.3	61	29	113	25.30678	39	112.25	26.45	57	26	119	25.64277
14	129.26	26.45	57	26	119	24.80506	40	105.1	33.6	61	29	125	27.48692
15	112.25	26.45	57	33.14	119	26.02377	41	112.25	26.45	57	26	119	25.64277
16	112.25	9.44	57	26	119	28.96266	42	112.25	26.45	57	26	119	25.64277
17	119.4	19.3	61	23	113	25.03575	43	119.4	19.3	53	29	113	24.51811
18	119.4	19.3	53	23	125	25.15874	44	112.25	26.45	57	26	104.73	24.53737
19	112.25	26.45	57	26	119	25.64277	45	105.1	19.3	61	29	113	28.69449
20	95.24	26.45	57	26	119	30.34922	46	112.25	26.45	66.51	26	119	26.9164
21	105.1	19.3	53	29	125	28.97801	47	105.1	33.6	53	23	113	26.11466
22	119.4	33.6	61	23	125	26.01751	48	112.25	26.45	57	26	133.27	26.11466
23	119.4	19.3	53	29	125	25.51013	49	105.1	19.3	61	23	113	28.56149
24	105.1	33.6	53	29	113	25.78437	50	105.1	33.6	61	23	125	27.22389
25	105.1	19.3	53	23	125	28.58835	51	119.4	33.6	61	29	113	26.0596
26	105.1	33.6	61	29	113	26.46299	52	119.4	19.3	61	29	125	26.30803

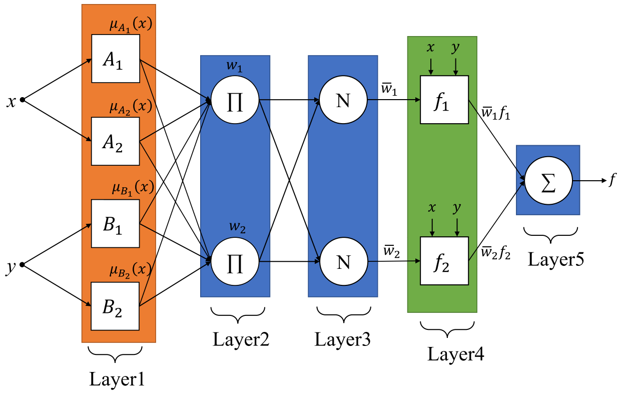


Fig. 5 The structure of ANFIS

The rule is layer 2, which is utilized to double the firing power. The output should be calculated as

$$Q_i^2 = w_i = \mu_{A_i}(u).\mu_{B_i}(y), i = 1, 2 \tag{5}$$

Layer 3 is for normalizing the last layer's firing intensity. The normalized values are generated by dividing the i^{th} rule's firing strength by the summation of all firing strengths:

$$Q_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2 + w_3 + w_4}, i = 1, \cdots, 4 \tag{6}$$

Layer 4 is the layer of defuzzification. The result is computed as the normalized firing strength and the parameter set dot product (p_i, q_i, and r_i) as

$$Q_i^4 = \overline{w}_i f_i = \overline{w}_i \left(p_i u + q_i y + r_i\right) \tag{7}$$

The final result of layer 5 is created by adding the defuzzification outputs of each rule:

$$Q_i^5 = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{8}$$

The error signals from the output to the input layers are computed using the gradient descent approach's backpropagation algorithm. The algorithm attempts to minimize the training error by adjusting the customizable parameters.

3.4 Multilayer perceptron

Originated by Rosenblatt [33], the MLP is an FFNN, and a single hidden layer MLP is known as vanilla [34]. The simplest MLP comprises three layers: output, hidden, and input. The neurons use nonlinear activation functions except the input nodes.

MLP can be used for classification and regression problems via supervised training. The inputs and outputs are used during the training process to adjust the network parameters, including weights and biases, to minimize the error between target and predicted outputs. The backpropagation algorithm is employed to derive the optimal weights and biases regarding the root mean square error (RMSE), mean square error (MSE), or other measures. The stochastic gradient descent technique is used in backpropagation through a backward pass of weights and biases through the MLP. During the training process, the j^{th} output node in the n^{th} data point can be represented as

$$h_j(n) = d_j(n) - y_j(n) \tag{9}$$

where d and y are the targets and actual outputs.

The weights can be calculated by minimizing the error using the entire network output, as follows:

$$\varepsilon(n) = \frac{1}{2} \sum_j h_j^2(n) \tag{10}$$

The variation of weights using the gradient descent method is

$$\Delta s_{ij}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial s_i(n)} y_i(n) \tag{11}$$

where y_i and s_i are the last neuron's output and local induced field, respectively. In addition, η is the learning rate that affects the convergence of the MLP. The weight derivative concerning the local induced field can be simplified for an output node as

$$-\frac{\partial \varepsilon(n)}{\partial s_i(n)} = h_j(n) \varnothing' \left(s_j(n)\right) \tag{12}$$

where $\varnothing'$ is the derivative of the activation function, which is constant against weight and local induced field. The backpropagation of the activation function method is used to vary the output layer weights by the hidden layer weights [35].

3.5 Long short-term memory

Hochreiter et al. [36] had demonstrated the effectiveness of the LSTM to address these disadvantages of the RNN. The gradient information eliminates or destroys if a highly long sequence is employed for learning. As an improved version of the RNN, the LSTM model can handle a long sequence of data samples using its memory blocks. The output is generated by a fully connected layer and a regression layer. It consists of three gates: input, output, and forget gates. Each gate is composed of the dot product and sigmoid function to protect the gradient information against distortion or elimination and control the information flow. Figure 6 presents a memory block depicting f multiplicative gating units to determine the information flow and memory cells to connect and memories the secular state.

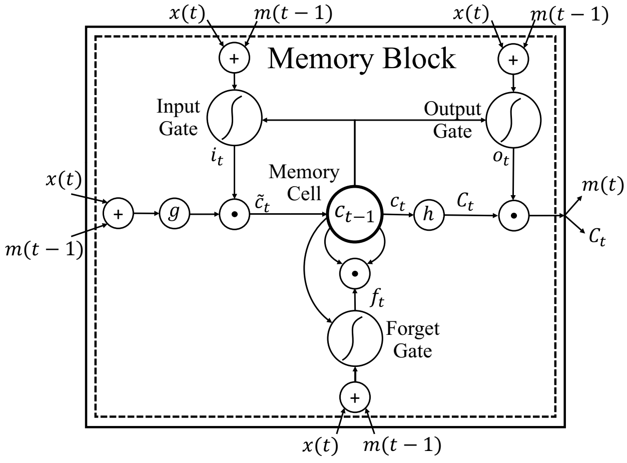


Fig. 6 A memory block depicting the data flow at time step t

The model input and output sequences are denoted by $\mathbf{x} = \left(x_1, x_2, \cdots, x_T\right)$ and $\mathbf{m} = \left(m_1, m_2, \cdots, m_T\right)$, respectively, where T is the prediction period. The memory cell of the j^{th} neuron at time t is denoted by c_t^j. The output of the j^{th} neuron, m_t^j, is

$$m_t^j = o_t^j \tanh\left(c_t^j\right) \tag{13}$$

where o_t^j is the output gate that decides the information to be propagated. The output gate is expressed as

$$o_t^j = \sigma\left(W_o x_t + U_o m_{t-1} + V_o c_t\right)^j \tag{14}$$

where m_{t-1} and c_t are the vector representations of m_{t-1}^j and c_t^j, respectively; while W_o, U_o, and V_o are the diagonal weight matrices that require online tuning with respect to minimizing a loss function. In addition, σ is a standard logistic sigmoid function defined as

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{15}$$

The focus of the memory cell is a recurrent constant error carousel (CEC) unit, which is activated to generate the cell state. The CEC benefits from eliminating the error caused by opening and closing the multiplicative gates in the LSTM model. The memory cell, c_t^j, has to be updated at each time step by elimination of the current memory cell and addition of the new memory value, $\tilde{c}_t^j$, as follows:

$$c_t^j = f_t^j c_{t-1}^j + i_t^j \tilde{c}_t^j \tag{16}$$

where the new memory value is

$$\tilde{c}_t^j = \tanh\left(W_c x_t + U_c m_{t-1}\right)^j \tag{17}$$

A forget gate prohibits the internal cell values from increasing without limit while continuing the time series mechanism (instead of segmenting). Then, the outdated information flow resets, and the CEC weight is substituted with the multiplicative forget gate activation. After updating the memory cell based on the new memory value, the forget gate, f_t^j, is computed as

$$f_t^j = \sigma\left(W_f x_t + U_f h_{t-1} + V_f c_{t-1}\right)^j \tag{18}$$

where W_f, U_f, and V_f are the diagonal weight matrices. The same methodology is adopted in the input gate, which determines the reserved new features as follows:

$$i_t^j = \sigma\left(W_i x_t + U_i h_{t-1} + V_i c_{t-1}\right)^j \tag{19}$$

where W_i, U_i, and V_i are the diagonal weight matrices. It should be noted that the value of the three gates is between 0 and 1. The LSTM output is formulated as

$$y = g\left(W_d h_t + b_d\right) \tag{20}$$

where g is a centered logistic sigmoid function within the range $[-2, 2]$, i.e.,

$$g(x) = \frac{4}{1 + e^{-x}} - 2 \tag{21}$$

Training of the LSTM is based on a modified real-time recurrent learning (RTRL) and a truncated backpropagation through time (BPTT) along with the gradient descent optimization method. The loss function is explained as the sum of square errors. The memory cell shortens the errors by exploiting the linear CEC of the memory cell. Inside the CEC, the error recedes and is discharged from the cell in a degraded exponential manner. This is the primary capability of the LSTM in dealing with a long prediction horizon compared with the RNN.

The hyperparameters of the three investigated machine-learning methods, namely, ANFIS, MLP, and LSTM, are chosen based on five evolutionary algorithms in the last five subsections of this section.

3.6 Genetic algorithm

The GA belongs to the group of evolutionary-based algorithms based on the evolutionary algorithms' natural selection. Recently, GA has been utilized to solve a wide range of optimization problems. It is a bio-inspired operator that uses mutation, crossover, and selection. Mitchell [37] devised this approach according to the natural selection process. Genetic algorithms often start with generating a randomly produced population of chromosomes. Following that, the created chromosomes are "assessed" using an objective function, and those that best approximate the "ideal solution" to the issue have a higher probability of reproducing.

Table 2 The selected hyperparameter of ANFIS uses four optimization methods

ITEM	Initial FIS	Epoch number	Initial step size	Step size decrease rate	Step size increase rate
GA	2	5	0.1000	1.0000	1.5000
PSO	3	7	0.0495	0.9796	1.1429
GWO	3	10	0.0604	0.6456	1.4587
BOA	2	7	0.0773	0.9472	1.3302
WOA	2	10	0.0874	0.7694	1.3513

GA parameters must be carefully chosen to maximize the accuracy and speed of the final result's convergence. The crossover and mutation parameters must be adjusted correctly to ensure the success of the GA and the acquired output. The high mutation rate raises the danger of missing a solution close to the present state. Setting the mutation parameter to a small value raises the likelihood of being trapped in the local optimum. In addition, crossover prevents kids from becoming identical replicas of the parents from the prior generations. The mutation parameter had to be set to a fair value and the crossover parameter to a significant value, as suggested by Mitchell [37]. In the present study, the population size is 80. The small population size imposes a constraint on the GA search capabilities. In addition, the large population increases optimization length without significantly improving the outcomes. The GA results will reveal that the mutation, crossover, generation, and population parameters have all been selected appropriately in this paper.

3.7 Particle swarm optimization

Kennedy et al. [38] developed PSO as a swarm-based algorithm category of the evolutionary algorithms. PSO is a stochastic population-based optimization method via the collaborative behavior of a different colony of species known as populations [39]. It is similar to evolutionary computation methods such as GA [37]. It starts by defining the initial particles to evaluate the objective function at every particle's location. It then decides a minor objective function as the best location. It calculates the new velocities according to the best position of the practices and their neighbors and the current velocity. The mentioned procedure continues iteratively until the algorithm fulfills the stopping criteria. PSO is capable of improving the candidate solution of an optimization problem iteratively.

3.8 Grey wolf optimization

In comparison to a GA [37] or PSO [38], Mirjalili et al. [40] proposed a GWO method that requires fewer parameters to be predefined. The adjustment of controller parameters is straightforward, with a balance between exploitation and exploration. GWO imitates grey wolves' social hunting behaviors. A grey wolf method is divided into four distinct groups: alphas (the search agent/solution of the highest rank α), betas (the second rank β), deltas (the third rank δ), and omegas (the fourth-ranked search agent ω). Omegas should lead all three ranking agents, while all other solutions are omegas. GWO's mathematical relationship is dependent on these four wolves' groups in prey hunting. A fitness function is necessary for GWO to estimate the optimal hyperparameters of the presented methods.

3.9 Butterfly optimization algorithm

Arora and Singh [41] were the first to suggest BOA, which relies on butterfly behavior in nature. Compared to well-established evolutionary algorithms, including GA, which retains qualifying solutions, BOA does not exclude any solutions from the search space, implying that every solution seems to have an equal opportunity to improve with the novel solution. Another notable distinction between BOA and typical evolutionary algorithms is that BOA computes the value of the fitness function for creating original solutions. In BOA, the butterflies act as search agents, moving randomly toward each other and generating further novel solutions. BOA is divided into three broad steps: initialization, iteration, and last phases. The approach defines the solution space, cost function, and boundary variables of the issue during the initialization stage by generating a population. In the second stage, BOA uses an iteration technique to identify the position of butterflies at random, computing and storing fitness values. Ultimately, BOA reaches the stop in the final stage, discovering the optimal solution to

Table 3 The selected hyperparameter of MLP uses five optimization methods

Method	No. of layers	No. of neurons	Mutation
GA	1	1	0.0099
PSO	1	10	0.0100
GWO	8	3	0.0012
BOA	1	4	0.0006
WOA	2	5	0.0004

Table 4 The selected hyperparameter of LSTM uses four optimization methods

Method	No. of layers	No. of units	Learning rate
GA	2	80	0.5000
PSO	1	244	0.4549
GWO	2	115	0.3857
BOA	2	219	0.6660
WOA	3	27	0.2691

the optimization issue. From a mathematical standpoint, the butterflies act as search agents for the BOA and generate fragrance at their positions utilizing $k = cI^a$, where c, k, a, and I represent the sensory modality, the fragrance's perceived magnitude and observed power exponent, magnitude reliant on stimulus intensity, and modality, respectively. The algorithm comprises two critical stages: local and universe searches.

Butterflies use techniques mentioned in the local and universe phases to discover activities, including food and partners. The necessities for food and partner have a noticeable fraction chance of p in such activities due to the physical proximity and several other elements, including rain and windy weather. The probability is used in this algorithm to find common approaches between the local and universe investigations in the problem's search space. BOA determines the best fitness employing the most outstanding solution discovered in the issue by satisfying the stopping criteria. It is worth noting that BOA is launched with the following parameters: $c = 0.01$, $a = [0.1,\ 0.3]$, and $p = 0.5$.

3.10 Whale optimization algorithm

Mirjalili et al. [36] suggested WOA based on the behavior of whales. The three operators in this algorithm are bubble-net foraging behaviors, prey search, and encircling prey. Whales detect the location of their prey and encircle it. The best hunting location is not shown right at the start of the hunt. The remaining agent whales will then gravitate toward the most effective search agent whale. The whales' reported behavior was mathematically modeled as follows:

$$L = \left| K \cdot P^{best}(t) - Kp_i(t) \right| \tag{22}$$

$$p_i(t+1) = P^{best}(t) - J \cdot L \tag{23}$$

where J and K denote coefficients, t is the current iteration, and p_i and P^{best} represent the position matrix for the i^{th} whale with the number of search agents and the best search agent's position vector, respectively. Moreover, L denotes the distance between the prey and the best agent. P^{best} should be replaced with a better option in each iteration. The coefficients J and K may also be computed using the following formula:

$$J = 2a \cdot r - a \tag{24}$$

$$K = 2 \cdot r \tag{25}$$

where $0 < r < 1$ is a random number that can be acquired using Eq. (26).

$$a = 2 - t \times \frac{2}{Max_{iter}} \tag{26}$$

Table 5 The three investigated machine-learning techniques, including ANFIS, MLP, and LSTM, using five optimization methods, including GA, PSO, GWO, BOA, and WOA

ML method	EA method	MSE (μm)	RMSE (μm)	NRMSE (μm)	CC	R^2	Mean (μm)	Std (μm)
ANFIS	GA	4.628	2.1513	2.1513	**0.80322**	0.30714	–0.17601	2.165
	PSO	4.8786	2.2087	0.083487	0.79179	0.29041	**–0.16377**	2.2242
	GWO	3.6713	1.9161	0.072424	0.95409	0.43146	–0.35242	1.9018
	BOA	**3.5210**	**1.8764**	**0.070926**	0.95153	**0.44205**	–0.33914	**1.8635**
	WOA	3.6957	1.9224	0.072665	0.95200	0.4307	–0.35194	1.9084
MLP	GA	0.61266	0.78273	0.029586	0.71162	0.64765	0.0251	0.78996
	PSO	0.033099	0.18193	0.0068767	0.97279	0.98607	0.037672	0.17973
	GWO	0.096533	0.31070	0.011744	**0.98465**	0.9615	–0.069621	0.30575
	BOA	**0.022988**	**0.15162**	**0.0057309**	0.97428	**0.99004**	**0.014916**	**0.15235**
	WOA	0.060377	0.24572	0.0092877	0.96316	0.97549	0.022393	0.24708
LSTM	GA	1.05410	1.02670	0.038808	0.67794	–0.61143	–0.143770	1.02650
	PSO	1.18240	1.08740	0.041102	0.68187	0.40542	–0.445580	1.00160
	GWO	1.16430	1.07900	0.040786	0.66048	0.51406	0.13704	1.08080
	BOA	0.94224	0.97069	0.036690	0.73049	0.066185	**–0.048618**	0.97893
	WOA	**0.58792**	**0.76676**	**0.028982**	**0.84293**	**0.65371**	0.076406	**0.77039**

ML machine learning, *EA* evolutionary algorithm, *MSE* mean square error, *RMSE* root mean square error, *NRMSE* normalized root mean square error, *CC* correlation coefficient

Fig. 7 **a** Actual and estimated workpiece elastic deformation during the machining based on the different locating and clamping positions using BOA-MLP during the testing stage. **b** Error between the actual and estimated workpiece elastic deformation during the machining based on the different locating and clamping positions using BOA-MLP during the testing stage. **c** Error histogram of the BOA-MLP for calculation of the workpiece elastic deformation during the machining based on the different locating and clamping positions during the testing stage. **d** Regression between the actual and estimated workpiece elastic deformation during the machining using BOA-MLP based on the different locating and clamping positions during the testing stage

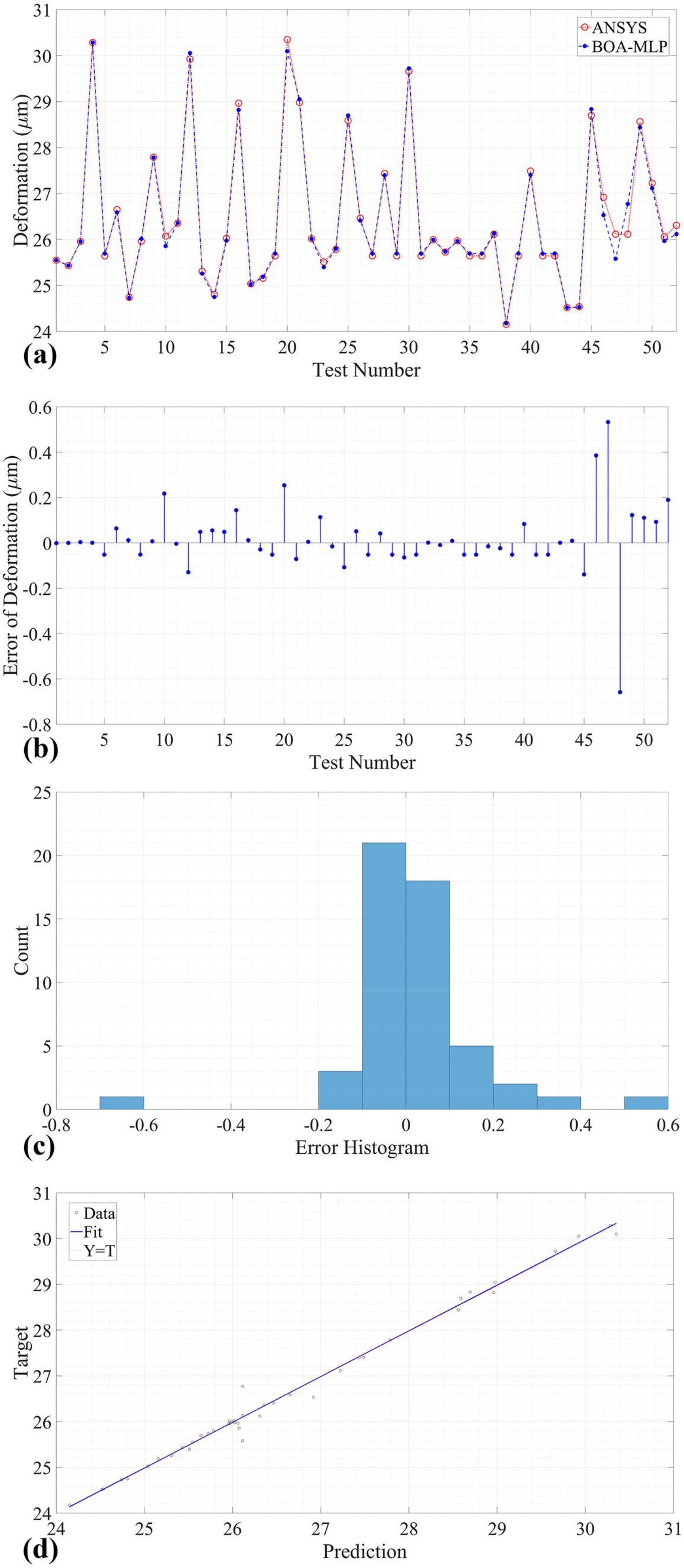
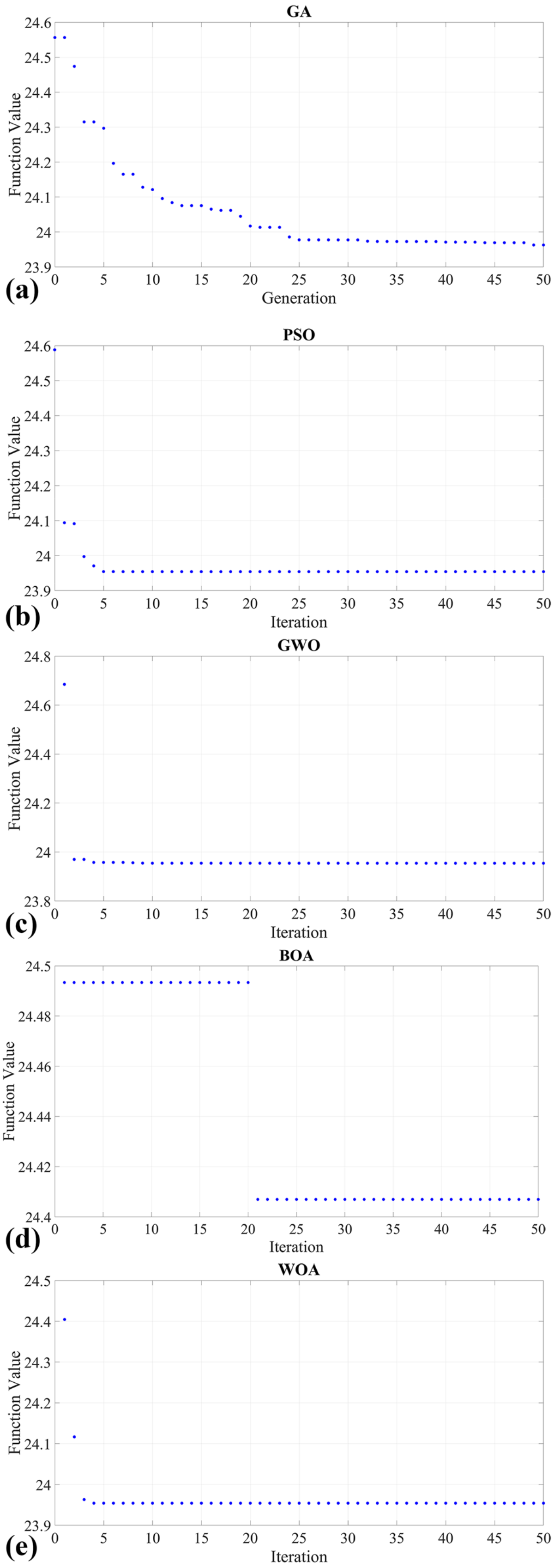


Fig. 8 The convergence plot of the evolutionary-based optimization ▶ algorithm for calculation of the best locating and clamping positions of the fixture layout to reach the lowest workpiece deformation during the machining via **a** GA, **b** PSO, **c** GWO, **d** BOA, and **e** WOA



The two basic mechanisms for bubble-net behavior used by humpback whales are the shrinking encircling mechanism and spiral updating position.

4 Results and discussions

Three models optimized by five evolutionary methods have been investigated in this study to estimate the elastic deformation of the workpiece throughout machining based on the different positions of the locating and clamping points. Moreover, the mentioned method is compared and validated with the previous work published in this topic using RSM [1] with the combined GA and PSO methods. The investigated methods are shown by GA, PSO, GWO, BOA, and WOA for optimized method and ANFIS, MLP, and LSTM for machine-learning methods. For instance, PSO-MLP refers to the MLP machine-learning method, which is optimized by the PSO algorithm. The models are developed in MATLAB software to be evaluated and prove the efficiency of the proposed methods. This section consists of two subsections regarding the verification and validation of this study.

4.1 Verification

The considered models in Sect. 3 with five optimization methods are developed in MATLAB. ANFIS is proposed according to the described model in Sect. 3.3. MLP is developed according to the described model in Sect. 3.4. LSTM is designed according to the described model in Sect. 3.5. Then, the models are optimized based on the five optimization methods in Sect. 3.6–3.10.

The ANFIS function of MATLAB is used to design the ANFIS model. Also, the feedforwardnet and trainNetwork function of MATLAB is used to design the MLP and LSTM model. The proposed algorithms in Sect. 3 are developed under the MATLAB coding environment, and all the results are extracted using the plot function. In addition, the GA and PSO optimization methods are developed using the ga and particleswarm functions of MATLAB. Also, GWO, BOA, and WOA are employed via the provided code in FileExchange [31–33]. Five optimization methods are used to extract the optimal hyperparameters of the proposed ANFIS, MLP, and LSTM. Initially, the evolutionary algorithms are implemented in ANFIS, then the extracted hyperparameters using GA, PSO, GWO, BOA, and WOA are shown in Table 2. In the second step of the research, the investigated evolutionary-based optimization algorithms are implemented in the MLP model, and the extracted optimal hyperparameters are reported in Table 3 using GA, PSO, GWO, BOA, and WOA, respectively. Lastly, five investigated evolutionary-based optimization algorithms are implemented in the LSTM model. The extracted three optimal hyperparameters, including the number of layers, number of units, and learning rate, are represented in Table 4. In the next step, the extracted ANFIS, MLP, and LSTM using the extracted optimal hyperparameters are trained and tested to find the most appropriate model in estimating the workpiece deformation during the machining process.

4.2 Validation

After extracting the hyperparameters shown in Tables 2, 3 and 4, the models are trained using 86.54% of the represented dataset in Table 2. Then, the models are tested using 13.46% of the dataset, and the results are reported in Table 5. Table 5 reports the results regarding MSE, RMSE, normalized root mean square error (NRMSE), mean, and Std between the calculated workpiece deformation during end milling via FEM and proposed methods. In addition, it represents the correlation coefficient (CC) and R^2 between the calculated workpiece deformation during end milling via FEM and proposed methods.

Based on the represented data in Table 5, among the ANFIS models, BOA is the best evolutionary-based optimization method in the extraction of the optimal hyperparameters with the lowest deformation of the workpiece during the machining process. In addition, BOA is the best evolutionary-based optimization method among MLP models in extracting the optimal hyperparameters with the lowest deformation of the workpiece during the machining process. Lastly, WOA is the best evolutionary-based optimization method in calculating the workpiece deformation during the machining process. After evaluating the models, the BOA-MLP is selected as the most suitable method for calculating the workpiece deformation inside the 3–2-1 located workpiece during the machining.

Figure 7a–b represents the results of the extracted BOA-MLP during the testing process as the best-extracted model for calculation of the workpiece elastic deformation during the machining based on the different locating and clamping positions. Figure 7a shows the actual and estimated workpiece elastic deformation during the machining based on the different locating and clamping positions using BOA-MLP during the testing stage. Figure 7b shows the error between the actual and estimated workpiece elastic deformation during the machining based on the different locating and clamping positions using BOA-MLP during the testing stage. Figure 7c presents the error histogram of the BOA-MLP for calculating the workpiece elastic deformation during the machining based on the different locating and clamping positions during the testing stage. Also, Fig. 7d presents the regression between the actual and estimated workpiece elastic deformation during the machining using BOA-MLP based on the different locating and clamping positions during the testing stage.

The extracted BOA-MLP is a candidate as the best model for representing the workpiece elastic deformation during the machining based on the different locating and clamping positions. Then, the five explained evolutionary-based optimization methods used this extracted model to calculate the best locating and clamping position of the fixture layout to reach the lowest elastic deformation of the workpiece. In addition, the previously proposed method using the GA- and PSO-RSM model [1] is investigated and compared with our recently proposed model to prove the efficiency of our proposed method.

The convergence plot of the five investigated evolutionary algorithms, including GA, PSO, GWO, BOA, and WOA, is shown in Fig. 8a-e. Based on the represented results, PSO, GWO, and WOA can reach the results more accurately and faster than GA and BOA. Also, the BOA has the slowest convergence speed between all investigated evolutionary-based optimization methods. GWO and WOA can extract the optimal solution after three iterations. Table 6 shows the mathematical representation of the results in Fig. 8a-e with the addition of the previous method results using the GA- and PSO-RSM method [1]. Based on the represented results

Table 6 The results of five investigated evolutionary algorithms via the BOA-MLP model and the previously proposed model using RSM [31]

Method	L_1 (mm)	L_2 (mm)	L_3 (mm)	C_1 (mm)	C_2 (mm)	Elastic deformation (μm)
OPT-RSM	118.73	24.71	53.09	23.63	113.42	23.9983
GA	119.39	24.46	53.01	23.04	113.01	23.9628
PSO	**119.40**	**24.45**	**53**	**23**	**113**	**23.9542**
GWO	**119.40**	**24.45**	**53**	**23**	**113**	**23.9542**
BOA	116.18	23.13	53	23.12	113	24.4069
WOA	**119.40**	**24.45**	**53**	**23**	**113**	**23.9542**

in Table 6, the elastic deformation of the workpiece during the machine of the workpiece using the extracted results of our proposed method is 0.0441 µm lower than the previous GA- and PSO-RSM method.

5 Conclusion

Fixtures are used in machining operation in order to locate the workpiece in precious location. Extracting the optimal fixture layout is not an easy task in order to reach the highest possible accuracy during the machining of workpieces. Previously, researchers used optimization methods, machine learning, and FEM models to extract the optimal fixture layout. However, there is a lack of comprehensive studies in evaluation of the fixture layout design using machine-learning and optimization methods. This study presented a comprehensive evaluation for optimal designing of the machining fixture using three standard machine-learning techniques, including ANFIS, MLP, and LSTM, which were optimized using five evolutionary algorithms including GA, PSO, GWO, BOA, and WOA. Also, the outcomes of the previously presented model using RSM, GA, and PSO were mentioned in this study for validation purposes. The investigated models were trained and tested using 86.54% and 13.46% of the FEM simulation software dataset. The prediction accuracy of the mentioned models was investigated using various parameters, including CC, MSE, RMSE, NRMSE, and R^2. Among the investigated methods, BOA-MLP reached the most accurate result based on the regenerated dataset in ANSYS software for the workpiece elastic deformation desiring the machining inside the fixture. The extracted accurate model was selected as the best accurate model to be employed inside the evolutionary algorithms to extract the optimal locating and clamping position of the 3–2-1 fixture during the machining. The results showed the 0.18% lower elastic deformation of the workpiece inside the 3–2-1 fixture during the machining using our proposed method compared to the previous GA- and PSO-RSM method.

Author contribution MRCQ: conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft, writing—review and editing, visualization, supervision, project administration. HP: conceptualization, resources, writing—review and editing, visualization. SP: writing—review and editing.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions. The authors declare that there is no funding to report regarding the present study.

Data availability All data used in this work have been properly cited within the article.

Code availability The authors declare that there is no code to report regarding the present study.

Declarations

Consent to participate The authors declare that all authors have read and approved to submit this manuscript to IJAMT.

Consent for publication The authors declare that all authors agree to sign the transfer of copyright for the publisher to publish this article upon on acceptance.

Competing interests The authors declare no competing interests.

References

1. Hoffman E (2012) Jig and fixture design. Cengage Learn
2. Kumar AS, Nee AY, Prombanpong S (1992) Expert fixture-design system for an automated manufacturing environment. Comput Aided Des 24(6):316–326
3. Wang Y, Chen X, Liu Q, Gindy N (2006) Optimisation of machining fixture layout under multi-constraints. Int J Mach Tools Manuf 46(12–13):1291–1300
4. Kang J, Chunzheng D, Jinxing K, Yi C, Yuwen S, Shanglin W (2020) Prediction of clamping deformation in vacuum fixture–workpiece system for low-rigidity thin-walled precision parts using finite element method. Int J Adv Manuf Technol 109(7):1895–1916
5. Wang H, Zhang K, Wu D, Yu T, Yu J, Liao Y (2021) Analysis and optimization of the machining fixture system stiffness for near-net-shaped aero-engine blade. Int J Adv Manuf Technol 113(11):3509–3523
6. Krishnakumar K, Melkote SN (2000) Machining fixture layout optimization using the genetic algorithm. Int J Mach Tools Manuf 40(4):579–598
7. Wang L et al (2021) Optimization of clamping for thin-walled rotational conical aluminum alloy with large diameter: modal simulation and experimental verification. Int J Adv Manuf Technol 114(7):2387–2396
8. Lee J, Haynes L (1987) Finite-element analysis of flexible fixturing system
9. Amaral N, Rencis JJ, Rong YK (2005) Development of a finite element analysis tool for fixture design integrity verification and optimisation. Int J Adv Manuf Technol 25(5):409–419
10. Edward C (1998) Fast support layout optimization. Int J Mach Tools Manuf 38(10–11):1221–1239

11. Li B, Melkote SN (1999) Improved workpiece location accuracy through fixture layout optimization. Int J Mach Tools Manuf 39(6):871–883
12. Hamedi M (2005) Intelligent fixture design through a hybrid system of artificial neural network and genetic algorithm. Artif Intell Rev 23(3):295–311
13. Padmanaban K, Arulshri K, Prabhakaran G (2009) Machining fixture layout design using ant colony algorithm based continuous optimization method. Int J Adv Manuf Technol 45(9):922–934
14. Selvakumar S, Arulshri K, Padmanaban K, Sasikumar K (2013) Design and optimization of machining fixture layout using ANN and DOE. Int J Adv Manuf Technol 65(9):1573–1586
15. Sundararaman K, Guharaja S, Padmanaban K, Sabareeswaran M (2014) Design and optimization of machining fixture layout for end-milling operation. Int J Adv Manuf Technol 73(5):669–679
16. Xing Y, Hu M, Zeng H, Wang Y (2015) Fixture layout optimisation based on a non-domination sorting social radiation algorithm for auto-body parts. Int J Prod Res 53(11):3475–3490
17. Lu C, Zhao H-W (2015) Fixture layout optimization for deformable sheet metal workpiece. Int J Adv Manuf Technol 78(1):85–98
18. Low DWW, Neo DWK, Kumar AS (2020) A study on automatic fixture design using reinforcement learning. Int J Adv Manuf Technol 107(5):2303–2311
19. Wu D, Zhao B, Wang H, Zhang K, Yu J (2020) Investigate on computer-aided fixture design and evaluation algorithm for near-net-shaped jet engine blade. J Manuf Process 54:393–412
20. Rezaei Aderiani A, Wärmefjord K, Söderberg R, Lindkvist L, Lindau B (2020) Optimal design of fixture layouts for compliant sheet metal assemblies. Int J Adv Manuf Technol 110(7):2181–2201
21. Wu B, Zheng Z, Wang J, Zhang Z, Zhang Y (2021) Layout optimization of auxiliary support for deflection errors suppression in end milling of flexible blade. Int J Adv Manuf Technol 115(5):1889–1905
22. Butt SU, Arshad M, Baqai AA, Saeed HA, Din NA, Khan RA (2021) Locator placement optimization for minimum part positioning error during machining operation using genetic algorithm. Int J Precis Eng Manuf 22(5):813–829
23. Du J, Liu C, Liu J, Zhang Y, Shi J (2021) Optimal design of fixture layout for compliant part with application in ship curved panel assembly. J Manuf Sci Eng 143(6)
24. Vinosh M, Raj TN, Prasath M (2021) Optimization of sheet metal resistance spot welding process fixture design. Materials Today: Proceedings 45:1696–1700
25. Michael Thomas Rex F, Hariharasakthisudhan P, Andrews A, Prince Abraham B (2022) Optimization of flexible fixture layout to improve form quality using parametric finite element model and mixed discrete-integer genetic algorithm. Proc Inst Mech Eng Part C: J Mech Eng Sci 236:16–29
26. Alshameri T, Dong Y, ALqadhi A (2022) Declining neighborhood simulated annealing algorithm to robust fixture synthesis in a point-set domain. Int J Adv Manufact Technol 119(11):8003–8023
27. Hu F (2022) Digital twin-driven reconfigurable fixturing optimization for trimming operation of aircraft skins. Aerospace 9(3):154
28. Yu K, Wang X (2022) Modeling and optimization of welding fixtures for a high-speed train aluminum alloy sidewall based on the response surface method. Int J Adv Manuf Technol 119(1):315–327
29. Li C, Wang Z, Tong H, Tian S, Yang L (2022) Optimization of the number and positions of fixture locators for curved thin-walled parts by whale optimization algorithm in J Phys: Conf Series 2174(1): IOP Publishing, p. 012013
30. Feng Q, Maier W, Stehle T, Möhring H-C (2022) Optimization of a clamping concept based on machine learning. Prod Eng Res Devel 16(1):9–22
31. Sundararaman K, Padmanaban K, Sabareeswaran M (2016) Optimization of machining fixture layout using integrated response surface methodology and evolutionary techniques. Proc Inst Mech Eng C J Mech Eng Sci 230(13):2245–2259
32. Walia N, Singh H, Sharma A (2015) ANFIS: adaptive neuro-fuzzy inference system—a survey. Int J Com App 13:123
33. Rosenblatt F (1957) The perceptron, a perceiving and recognizing automaton Project Para. Cornell Aeronautic Lab
34. Hastie T, Tibshirani R, Friedman J (2009) The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media, 2009
35. Haykin S, Network N (2004) A comprehensive foundation. Neural Netw 2(2004):41
36. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780
37. Mitchell M (1998) An introduction to genetic algorithms. MIT press
38. Kennedy J, Eberhart R (1995) Particle swarm optimization (PSO), in Proc. IEEE International Conference on Neural Networks, Perth, Australia, pp. 1942–1948
39. Rini DP, Shamsuddin SM, Yuhaniz SS (2011) Particle swarm optimization: technique, system and challenges. Int J Com App 14(1):19–26
40. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
41. Arora S, Singh S (2019) Butterfly optimization algorithm: a novel approach for global optimization. Soft Comput 23(3):715–734