S & M 3488

Spindle Thermal Error Prediction Based on Back Propagation Neural Network and Hybrid Taguchi Genetic Algorithm for a Computer Numerical Control Machine Tool

Chi-Yuan Lin,[1,2] Shu-Cing Wu,[2] Chi-Hsien Hu,[3] and Bo-Lin Jian[3*]

[1]Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung 411030, Taiwan
[2]Prospective Technology of Electrical Engineering and Computer Science, National Chin-Yi University of Technology, Taichung 411030, Taiwan
[3]Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung 411030, Taiwan

(Received July 6, 2023; accepted December 4, 2023)

Keywords: CNC milling machine, spindle thermal deformation, eddy current displacement meter, hybrid Taguchi genetic algorithm.

In this study, a search was made for the optimal training parameters for a back propagation neural network (BPNN) using the hybrid Taguchi genetic algorithm (HTGA). This was conducted to enhance the predictive accuracy of the model and solve the difficult problem of BPNN parameter adjustment. The 10-fold cross-validation method was used for verification and to assess the pros and cons of the model as well as optimize the training parameters. Practical spindle thermal deformation experiments were also conducted to verify the prediction results using a computer numerical control milling machine at different spindle speeds using contact thermal sensors and an eddy current sensor to measure deformation. The findings of this research demonstrate that the training parameters for the BPNN, when optimized using the HTGA, exhibit superior performance compared with those obtained through the conventional genetic algorithm methodology. The results of the experiment in thermal deformation and displacement indicate that the root-mean-square error of the predicted displacement and the actual displacement for the optimized BPNN training parameter model using HTGA were within 6 μm, and the results were better than those found by conventional methods.

1. Introduction

Methods for the enhancement of machining accuracy have attracted considerable attention as the demand for high-precision machining has risen. To achieve high precision, it is necessary to reduce errors.[(1)] Spindle thermal error accounts for 40 to 70% of the total thermal error[(2,3)] in precision machining. The reduction of spindle thermal expansion has been approached in several different ways.[(4)] The first approach uses cooling oil sprayed onto the spindle to lower the temperature. The second approach keeps the spindle at a constant temperature and uses a fixed value for error compensation. The third approach is to ensure an even distribution of temperature

[*]Corresponding author: e-mail: BoLin@ncut.edu.tw

https://doi.org/10.18494/SAM4674

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

all over the spindle to prevent irregular deformation. The fourth approach establishes a relationship between the spindle temperature and displacement. The amount of expansion is measured and a controller is used to compensate for the displacement.(5,6) Li *et al.*(7) proposed the use of improved particle swarm optimization (IPSO) and back propagation neural network (BPNN). IPSO is utilized to optimize parameters such as the initial weights and thresholds of BPNN, and it is compared with the genetic algorithm (GA)-BPNN prediction model. The results showed a spindle thermal error prediction accuracy of 93.1% for the GA-BPNN model. The IPSO-BPNN model was more effective and showed an accuracy of 96.5%. Wang(8) utilized the Grey Theory to reduce the number of temperature measurement points and adopted the hierarchy-GA (HGA) coupled with BPNN. The results indicated that the HGA-BPNN model can be applied to any type of computer-numerical-controlled (CNC) thermal error issue. Liang *et al.*(9) constructed a thermal error model by employing linear regression (LR), BPNN, and radial basis function (RBF) networks. Their results showed that the RBF thermal error model has a higher prediction accuracy than the LR and BPNN models. The use of the RBF model can enhance the machining accuracy by 65%. Li *et al.*(10) suggested that BPNN was suitable for the establishment of a thermal error model and utilized the bat algorithm (BA) to enhance the performance of BPNN. Integrating the BA with BPNN significantly improved the initially poor prediction accuracy of the latter. Comparative results showed that the BA-enhanced BPNN thermal error model was more stable, exhibiting fewer performance fluctuations, and more robust with higher prediction accuracy than the conventional BPNN model. Tan *et al.*(11) applied the least absolute shrinkage and selection operator to decrease the number of temperature measurement points to seven and developed a thermal error model using least squares support vector machine (LS-SVM) regression. This model was then evaluated against the grey and multiple linear regression models. The results showed that LS-SVM gave a better prediction than the grey model and multiple linear regression analysis, the enhancements being 74.6 and 54.3%, respectively. This verified the feasibility of creating a model with LS-SVM. Jian *et al.*(12) measured the increase in the temperature and displacement of a lathe spindle at spindle speeds of 1000, 2000, and 3000 rpm and created a thermal error model with multiple regression analysis and a general regression neural network (GRNN). The results showed that the GRNN model was better than the multiple regression analysis, and the prediction error of the GRNN model was less than 0.1 mm. Tseng and Chen(13) created a thermal error model using neural-fuzzy theory and multivariable linear regression analysis. The thermal error model with neural-fuzzy theory enhanced machining accuracy from 80 to 3 μm, and the thermal error model with multivariable linear regression analysis enhanced the machining accuracy by 10 μm. The results showed that the thermal error model with neural-fuzzy theory is better than the thermal error model with multivariable linear regression analysis, which enhanced the machining accuracy from 10 μm to 3 μm. Liu *et al.*(14) proposed a thermal error model with the improved grey wolf optimizer (IGWO) for the optimization of the smoothing parameter of the GRNN. The results showed the prediction accuracy of IGWO-GRNN to be 5.1% (or more) higher than that of the other algorithms.

Clearly, a robust model(15) is necessary for accurate prediction, and a suitable neural network based on the data type should be selected. Previous research showed that these thermal errors are nonlinear, and BPNN could be used for the efficient processing of nonlinear issues.(16) However, the initial parameters for training had to be determined before the neural network could be utilized; this was done by trial and error in the past. This is time-consuming and complicated because the optimal value for each parameter must be determined. The optimization method can help solve the above-mentioned problems. However, there are many optimization methods, such as GA, PSO, and simulated annealing algorithm (SAA). It is difficult to compare each optimization method because the results are varied by the convexity and concavity of the objective function and constraints.(17) Researchers who have used this method in the past found that GA is worthy of reference in terms of its ability to search for solutions without relying on GPU, and is also widely used in various fields.(18,19) As a result of the above discussion, in this study, we continue to investigate GA-based optimization methods. The results of the above discussion show that it is feasible to use various optimization methods to find the best settings for various model parameters. It is certain that this approach can overcome the issue of tuning the highly nonlinear-correlated parameters, in the case of not considering the computational complexity. Thus, the complexity problem and robustness are the limitations of the current optimization method. Therefore, in this study, we aim at finding a robust optimization method that outperforms the common optimizers such as GA and a suitable scenario for the application. Tsai *et al.*(20) proposed the hybrid Taguchi genetic algorithm (HTGA) method, which introduces Taguchi's method into the crossover step of GA and significantly improves the original sensitivity to mutation and crossover rates of GA. Therefore, in this study, HTGA was used. Its powerful global search capabilities can yield the optimal BPNN parameter settings for enhancing the prediction accuracy of BPNN in a reasonable time. Although the above optimization methods have been experimentally proven to be effective in many studies,(17–19) there are undeniably many excellent examples, such as the artificial bee colony (ABC) algorithm,(21) monarch butterfly optimization (MBO),(22) earthworm optimization algorithm (EWA),(23) elephant herding optimization (EHO),(24) moth search (MS) algorithm,(25) slime mold algorithm (SMA),(26) and Harris hawks optimization (HHO)(27) that can be referred to. The *K*-fold cross-validation (CV) was also utilized to verify the quality of the parameter model. A robust model with high prediction accuracy, which had optimal BPNN parameters, was finally created. Thus, the objective of this study is to develop a model for predicting the deformation of the machined spindle owing to thermal variation. The model can also be used to compensate for the deformation during actual machining in the future. For the speed and convenience of implementation, the model structure will be BPNN, and the model will be trained with numerous parameter settings, so HTGA will be used to search for the best parameters.

In summary, the main structure, machine model, and sensors used in this study are introduced in Sect. 2.1. In Sect. 2.2, the BPNN with *K*-fold CV and HTGA are introduced. The details of data collection are described in Sect. 3.1. In Sect. 3.2, the parameters adjusted by the optimization method and the corresponding experimental results are described and discussed. In Sect. 3.3, the use of optimal BPNN training parameters for model creation and the experimental results of using GA and HTGA are illustrated. The differences and improvements are also shown. The final contribution of the whole study is presented in Conclusions.

2. Methods

2.1 Equipment and main process flowchart

A Brother S500X1 CNC milling machine was used in this study. Temperature changes were measured using PT1000 contact-type temperature sensors with a measurement range from 0 to 200 °C. The accuracy was ±0.15% and the sampling frequency was 5 s. An EX-110V eddy current displacement sensor was also used. Calculations of displacement were made using variations in eddy current generated by magnetic induction on the surface of a moving metal plate. The measurement range was from 0 to 2 mm, the accuracy is the full scale ±0.3%, with a resolution of 0.4 and a sampling frequency of 5 s. The location of the sensors is shown in Fig. 1. As displacement was difficult to measure while the machine was running, the relationship between the increased temperature and displacement of the spindle during idle running was established to determine displacement based on spindle temperature. Spindle temperature and displacement data were collected, and HTGA and GA were used to find the optimal training parameters for BPNN. The optimal training parameters were found and an accurate prediction model was created. A diagram of the experimental structure is shown in Fig. 2. The computer used in this study ran the MATLAB Deep Learning Toolbox 13.0 in Windows 10. It had an Intel i7-9700 processor and 32 GB of memory. The model training data were collected at spindle speeds of 1000, 2500, 5000, 7500, and 9000 rpm; these were 10, 25, 50, 75, and 90% of the maximum machine spindle speed of 10000 rpm, respectively. In addition, the spindle speeds most frequently used in normal machining operations, namely, 2000, 3000, and 6000 rpm, were used for verification of the model data and to search for the optimal training parameters of BPNN based on HTGA and GA for model creation.

In the first part of the process, temperature and displacement signals is collected from the sensors [see Figs. 1(a) and 1(c)]. In the second part, HTGA is used to search for the optimal BPNN training parameters. In the third part, the optimal parameters are used to create the model.

2.2 Classifier and optimization methods

An important step in this study was the establishment of a relationship between spindle temperature and displacement. The temperature–displacement relationship is a complex combination of nonlinear equations, but it is not easy to find this nonlinear equation model. For example, Jian *et al.*(12) used COMSOL to simulate the thermal dynamics of the spindle, but eventually, a general regression neural network was adopted to model the temperature and displacement. From the literature, it is clear that machine learning to model temperature and displacement is the future trend. A previous study(16) revealed that BPNN can be used for nonlinear mapping and is suitable for the creation of a thermal error model, and it was therefore used in this study. Neural network is the term used for a mathematical model that simulates a neural system. The neural network has neurons or nodes connected to each other. The neurons acquire information from the outside world or from other neurons and send information to the

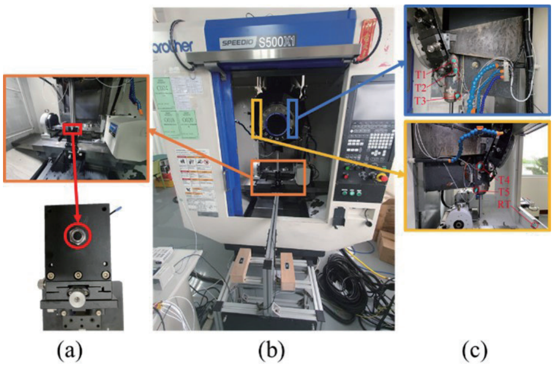
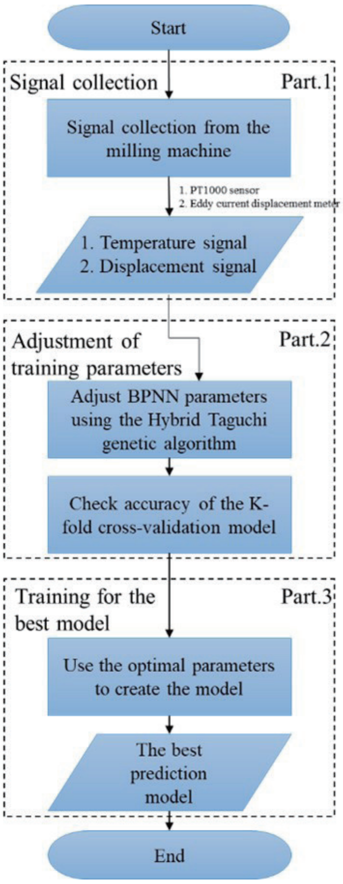


Fig. 1. (Color online) Machine used in experiments and locations of measuring devices. (a) Eddy current displacement meter, (b) S500X1 machine used in experiments, and (c) locations of temperature sensors.



Start
Signal collection
Part.1
Signal collection from the milling machine
1. PT1000 sensor
2. Eddy current displacement meter
1. Temperature signal
2. Displacement signal
Adjustment of training parameters
Part.2
Adjust BPNN parameters using the Hybrid Taguchi genetic algorithm
Check accuracy of the K-fold cross-validation model
Training for the best model
Part.3
Use the optimal parameters to create the model
The best prediction model
End

Fig. 2. (Color online) Experimental flowchart.

outside world or other neurons after simple computations. The purpose of the neural network is to simulate the biological neural system and to handle a large number of parallel computations, as well as distributed storage and processing. The nonlinear mapping and data-oriented characteristics of a neural network can solve many problems, and it is quite different from the model creation methods used in the past. The advantage of a neural network is that it is data-oriented instead of pattern-oriented, and the assumption of a relationship between input and output is not required for data orientation. The neural network has high data error tolerance and easily adapts to new data. Specifically, statistical regression is assumed to be a linear structure; however, since the sequential data of the heat displacement are formed randomly, it is nonlinear data. Therefore, statistical regression is not suitable to obtain the relationship between the input and output data. A simple neural network has an input layer, a hidden layer, and an output layer. The layers are formed by many nodes. The input layer receives data in the network structure and the output layer sends it out. There is usually only one layer for output and one for input. The hidden layer interacts with the processing unit of the neural network and uses a nonlinear transfer function. The number of hidden layers is usually 1 or 2 for good convergence. The more layers there are, the more complicated the problem will become. Too many layers will cause network convergence problems. The BPNN structure diagram is shown in Fig. 3.

Here, X_i is the input vector, Y_o is the output vector corresponding to the input vector, and θ is the initial value of the neuron, also known as bias. The value of the hidden layer is the value of weights through the connection input and hidden from the value of the input layer. The computation of the neuron value of the hidden layer is expressed as[28]

$$H_h = f\left(x\right) = \left(\sum_{i=1}^{N} W_{ih} X_i\right) - \theta_h, \tag{1}$$

where H_h is the hth neuron of the hidden layer, X_i is the ith neuron of the input layer, W_{ih} is the connection weight of the ith neuron of the input layer and the kth neuron of the hidden layer, and θ_h is the bias value of the hth neuron of the hidden layer. The dimensional numbers of both the input and output layers are N.

The BPNN includes forward and back propagations. The forward propagation is the output value of each neuron obtained from computation through the input layer, and the error between the output value and the actual value is also calculated. The error calculation method is expressed as[29]

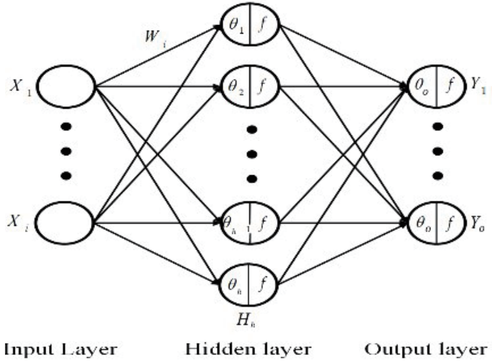


Fig. 3. Structure of neural network.

$$E = \frac{1}{2}\sum_{o}^{N}\left(\left(T_o - Y_o\right)^2\right), \tag{2}$$

where E is the error between the output value and the actual value, T_o is the oth actual value, Y_o is the oth output value, and N is the number of neurons in the output layer. The constant in the equation is presumed for a subsequent differential cancellation.

If the error is greater than the preset tolerance value after forward propagation, then the BPNN will be calculated for the corrected weights using the gradient descent method.(30) Its weight correction equation is (31)

$$\Delta W = -\eta \frac{\partial E}{\partial W}, \tag{3}$$

where W is the weight connection to each neuron, ∂E is the corrected amount of the weight connection to each neuron, and η is the learning rate that controls the rate of the corrected weight. Poor network convergence will result if the learning rate is too large or too small.

In this paper, after the BPNN structure has been described, the optimized BPNN training parameters used in this study will be introduced. To show the performances of the trained and untrained BPNN, the BPNN should have the same structure as that shown in Fig. 3. The input layer is five temperature points, using a single hidden layer with a sigmoid activation function, and the output layer is also a single layer with a linear activation function. In addition, the optimization method will tune the following parameters. First, if there are too few neurons in the hidden layer, the relationship between output and input will result in ineffective mapping. Too many neurons will lengthen the computation time. The second adjusted parameter is the training function. A suitable training function will enhance the convergence speed and prediction accuracy of the network. The third adjustment parameter is the learning rate. This parameter has an impact on network convergence; if the value of the parameter is too high, the network will converge to a local minimum, and if the value is too small, the network calculation time will increase. In this study 10-fold verification was used to ensure the objectivity of the BPNN training parameters for the model, and it was also used as a basis for the adjustment of model parameters. Suitable training data are very important for neural network training. Factors such as the size of the training sample and the correctness of the data in the sample have a significant impact on the training of a neural network. The best training can be achieved by extensive, even, and differential data distribution. Cross-validation is a verification standard for model quality, and the proper selection of training data is very important. The model should not be based on only one training result because this may not be a valid selection. The 10-fold method is a better and more commonly used cross-validation method. The data is divided into 10 sets; one is selected as test data and the remaining sets are used for training. This is continued until all sets have been used as test data.(32,33) The average of the 10 implementations is used for the model. A schematic of the 10-fold method is shown in Fig. 4.

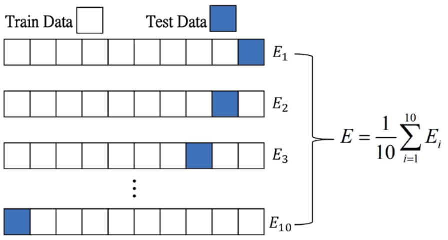


Fig. 4. (Color online) Tenfold schematic diagram.

The test data set is blue and the training data is white. E_i represents the error of the ith training and E is the total average error, which is the final error of the model. The quality of the model can be judged from the final average; a large value is an indication of instability.

HTGA was proposed by Tsai *et al.*(20) in 2004, and in this study, it was used to search for the optimal BPNN training parameters to solve the complicated and time-consuming problem of BPNN parameter adjustment. HTGA is a combination of the conventional GA and the Taguchi method. The development of GA was based on Darwin's theory of evolution and the concept of survival of the fittest and elimination, and it includes selection, crossover, and mutation. Selection produces individuals with higher fitness for crossover, and the fittest have the highest probability of selection. Roulette wheel selection is often used for selecting individuals with higher fitness than the total fitness value of a group. Because they occupy a greater area, there is more chance of them being selected. The crossover is performed after good genes have been selected. The crossover rate must first be defined and will only be performed for two sets of chromosomes with a random value greater than the crossover rate. The crossover may be single- or two-point. In a single-point crossover, a random point is selected, and the two chromosomes are cut and recombined into a new chromosome. In two-point crossover, the chromosomes are cut in two places and the three bits are recombined. Mutation prevents too early a convergence because this would prevent an optimal global value being reached. The method provides a variation rate, and if the randomly generated number is greater than this rate, variation will take place.

The Taguchi method(34) was initially used in agricultural production and biological analysis. It is also called quality engineering as it made a considerable contribution to industrial production. Product quality can be considerably improved using the simple orthogonal Taguchi arrays for experiments and variance analysis. The Taguchi method allows an experimental result that is close to the total divisor to be obtained with the fewest number of experiments. The method has the following characteristics: quality characteristics based on loss functions, the selection and definition of experimental factors, signal-to-noise (*S*/*N*) ratio, and an orthogonal table. The Taguchi methods used in HTGA include orthogonal arrays, the characteristic of the larger the better and smaller the better, the *S*/*N* ratio, and so forth.(35) The orthogonal arrays are often used to simplify experiments and have the following advantages: their use greatly reduces the number of experiments needed and allows simple data analysis. The orthogonal arrays originated from the Latin Square and are usually represented by L. GA is usually based on the crossover of two sets of chromosomes, as shown in the following example. Here, the number 8 after the symbol L represents the total number of experiments; there are two levels for each factor, and the number of control factors is seven. Optimization was used to search for the maximum or minimum values in this study. The expected value of the larger the better and its loss function is as shown in Eq. (4). The expected value of the smaller the better is 0 and its loss function is as shown in Eq. (5).(36)

$$S = \frac{1}{n}\sum_{t=1}^{n} y_t^2 \tag{4}$$

$$S = \frac{1}{n}\sum_{t=1}^{n} \frac{1}{y_t^2} \tag{5}$$

y is the objective function, *n* is the number of objection functions, and Eqs. (4) and (5) represent the mean-square deviation with the target. Taguchi utilized the *S*/*N* ratio to list the average value as a useful signal, and the loss caused by variation is included in the loss signal. The higher the ratio, the better the quality. They are changed in Eqs. (6) and (7).

$$\eta_{(SLB)} = -10\log\left(\frac{1}{n}\sum_{t=1}^{n} y_t^2\right) \tag{6}$$

$$\eta_{(LTB)} = -10\log\left(\frac{1}{n}\sum_{t=1}^{n} \frac{1}{y_t^2}\right) \tag{7}$$

3. Results

3.1 Data collection and analysis

In this study, spindle temperature and displacement data were collected with the machine running at spindle speeds of 1000, 2000, 2500, 3000, 5000, 6000, 7500, and 9000 rpm. The results for spindle speeds of 3000 and 7500 rpm are shown in Figs. 5 and 6, respectively. Figure 1 shows the location of the sensors.

Figures 5 and 6 show that the temperature at a measurement point does not change immediately with a change in ambient temperature; there is a delay. The displacement varies with a change in temperature at different spindle speeds. The changes shown in Figs. 5 and 6 confirm that it is feasible to use spindle temperature for the prediction of displacement. The ambient temperature and temperature measurement points in Fig. 5 do not show a significant change. In Figs. 5 and 6, the displacement reached a steady state after about 2 h of operation, and

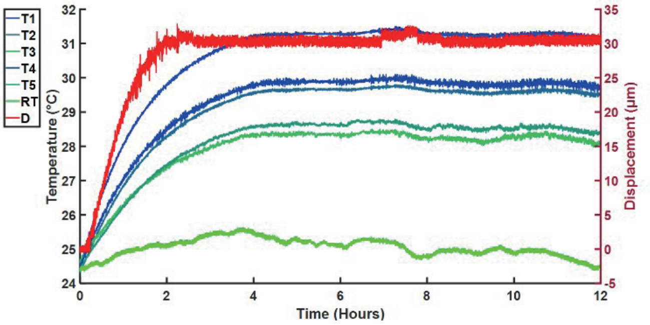


Fig. 5. (Color online) Temperature at six locations and spindle displacement at a spindle speed of 3000 rpm.

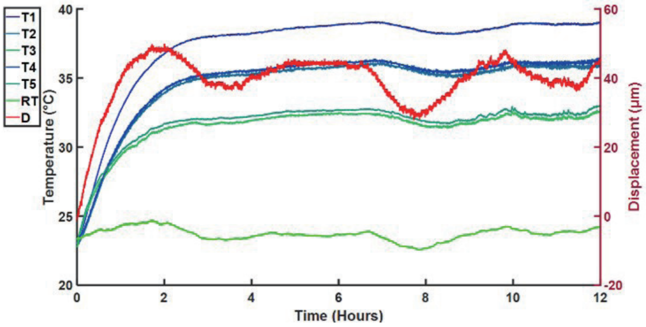


Fig. 6. (Color online) Temperature at six locations and spindle displacement at a spindle speed of 7500 rpm.

the change in displacement is based on the changes measured after 2 to 12 h of operation. In Fig. 6, the ambient temperature changes more significantly between 8 and 12 h, and the temperature measurement point is also affected more significantly, so we can see that the environmental temperature affects the temperature measurement. In addition, the displacement in Fig. 6 also increases with the increase in temperature over 8 to 12 h.

3.2 Adjustment of BPNN training parameters

The first BPNN training parameter adjusted was the relationship between output and input that cannot be effectively mapped owing to the very few neurons in the hidden layer. However, too many neurons will increase the computation time. The second parameter adjusted was the training function. A suitable training function enhances the convergence speed and prediction accuracy of the network. The training function used in this study was set using the training function application in the MATLAB Deep Learning Toolbox 13.0. The function and algorithm name are shown in Table 1. The third parameter to be adjusted was the learning rate. A learning rate that is too large will make it difficult for the network to converge to a local minimum. If the

Table 1
Training functions and algorithms.

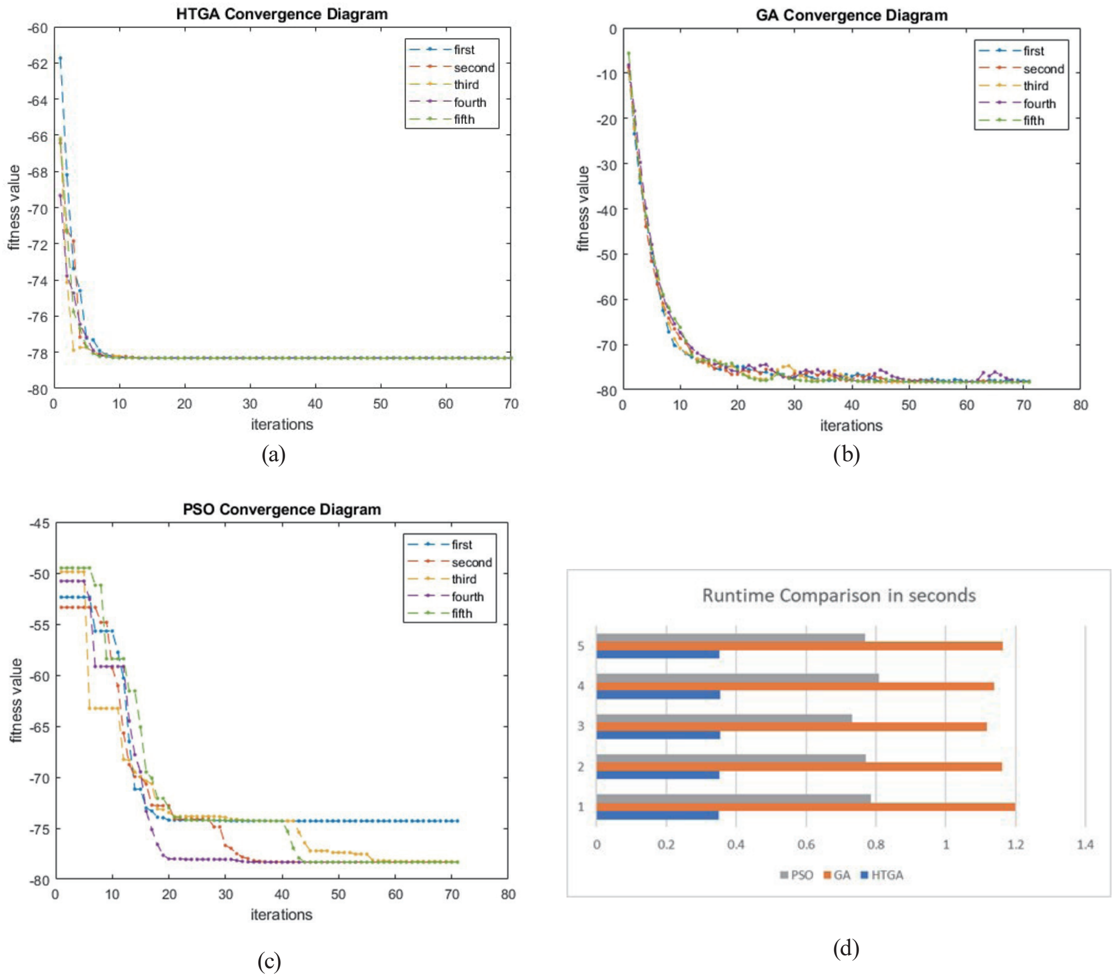
Abbreviation	Name of algorithm
LM	Levenberg–Marquardt[37]
BR	Bayesian Regularization[38]
BFG	BFGS Quasi-Newton[39]
RP	Resilient Backpropagation[40]
SCG	Scaled Conjugate Gradient[41]
CGB	Conjugate Gradient with Powell/Beale Restarts[42]
CGF	Fletcher–Powell Conjugate Gradient[43]
CGP	Polak–Ribiére Conjugate Gradient[43]
OSS	One Step Secant[44]
GDX	Variable Learning Rate Gradient Descent[45]
GDM	Gradient Descent with Momentum[45]
GD	Gradient Descent[45]

parameter is too small, the time taken for calculation will increase, and it will take longer to reach a minimum. All these parameters are determined by data type, and different data have different parameters. In this study, HTGA was used to search for the optimal BPNN training parameters, and they were compared with the optimal BPNN training parameters using GA and PSO, the two common optimization methods widely used in engineering. The parameter settings for HTGA and GA were 50 for the number of chromosomes, 70 for iteration, 0.7 for the crossover rate, and 0.2 for the variation rate. BPNN uses one-layer hidden layers. The range of BPNN parameters used for the number of neurons in the hidden layer was from 10 to 100. The training function learning rate was from 0.0001 to 1. The details of the training function and names of the algorithm are listed in Table 1. The convergence results of the adjusted BPNN parameters using HTGA, GA, and PSO are shown in Figs. 7(a)–7(c), respectively, and the runtime comparison is presented for evaluating the computational complexity.

In Fig. 7, HTGA not only demonstrates a more rapid convergence speed but also exhibits greater robustness in comparison with the GA and PSO. This is evident as HTGA's fitness values stabilize at a constant level, whereas the fitness values for GA and PSO may fluctuate over time. Also, HTGA helps reduce the computational burden for the computer when having the shortest time to converge with the same settings. When HTGA is used to find the optimal BPNN training parameters, if the number of neurons in the hidden layer is 15 and the training function used is the Levenberg–Marquardt algorithm, the learning rate will be 0.5. Comparison with the results found in a previous study[46] confirms that the optimization of BPNN training parameters using HTGA can eliminate the tedious and time-consuming adjustment of the parameters by trial and error. According to the above results, the proposed solution (HTGA) outperforms the GA and can expedite the optimization process, thus resolving the challenge of selecting BPNN training parameters more efficiently.

3.3 Use of optimal BPNN training parameters for model creation

In the previous section, the optimal BPNN parameters were found using HTGA. In this section, those optimal parameters were used for creating a thermal error neural network model.



(a)

(b)

(c)

(d)

Fig. 7. (Color online) Convergence diagrams and runtime comparison for the proposed HTGA, GA, and PSO. (a) Convergence diagram of HTGA. (b) Convergence diagram of GA. (c) Convergence diagram of PSO. (d) Runtime comparison.

The training data used spindle speeds of 1000, 2500, 5000, 7500, and 9000 rpm, which represent 10, 25, 50, 75, and 90%, respectively, of the maximum spindle speed (10000 rpm) of the machine used in the experiments. The spindle speeds most commonly used for actual machining, namely, 2000, 3000, and 6000 rpm, were used for model data verification and to search for the optimal training parameters of BPNN using HTGA (BP-HTGA) as well as GA (BP-GA) for model creation. The training and verification data of the optimal training parameters found using HTGA for model creation and the results of the verification data compared with the optimal training parameters of BPNN using GA are shown in Figs. 8–10. The search for the optimal training parameters of BPNN using HTGA and GA for model creation and the error between the prediction and actual value of the model are shown in Table 2.

Figures 8–10 confirm that the results obtained in the training model created using BP-HTGA at various speeds are good. The predicted displacement values are very close to the actual values. A reference to the error values of training data in Table 2 shows that the BP-HTGA errors are within 2 μm, and the training errors at various spindle speeds are all smaller than those of

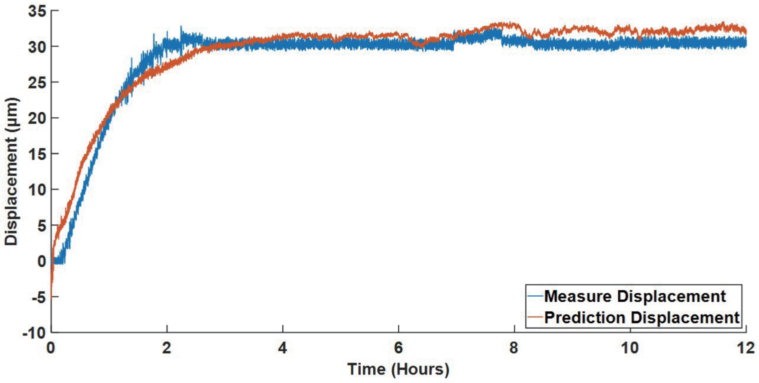


Fig. 8. (Color online) Prediction and actual displacement of spindle at 7500 rpm (BP-HTGA).

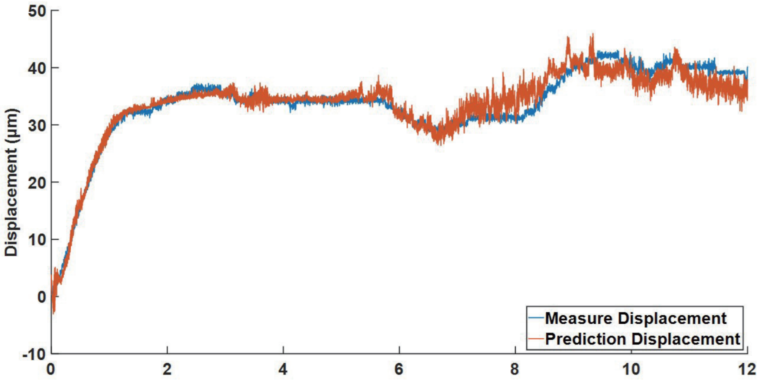


Fig. 9. (Color online) Prediction and actual displacement of spindle at 3000 rpm (BP-HTGA).

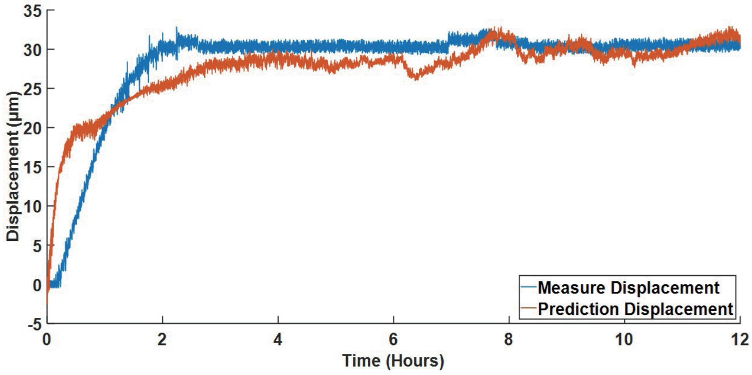


Fig. 10. (Color online) Prediction and actual displacement of spindle at 2000 rpm (BP-HTGA).

the BP-GA training data. The BP-HTGA model is clearly better than the BP-GA model for training. The test data were then used to verify the predictive ability of the model. Figures 9 and 10 show that BP-HTGA had better prediction at a spindle speed of 3000 rpm. Furthermore, according to the test data in Table 2, all the BP-HTGA prediction errors are less than those seen in BP-GA at different spindle speeds, and all the root-mean-square errors of BP-HTGA in the

Table 2
Root-mean-square error and mean-square error of model prediction and actual displacement.

	Spindle speed (rpm)	BP-HTGA		BP-GA	
		Mean-square error (μm)	Root-mean-square error (μm)	Mean-square error (μ)	Root-mean-square error (μm)
Training data	1000	1.5177	1.2319	2.1188	1.4556
	2500	0.6934	0.8327	0.9619	0.9808
	5000	0.5691	0.7544	0.6684	0.8176
	7500	3.5189	1.8759	3.8791	1.9695
	9000	1.8702	1.3675	2.0900	1.4457
Test data	2000	25.0911	5.0091	34.6703	5.8881
	3000	3.3098	1.8193	9.6334	3.1038
	6000	11.3408	3.3676	92.9659	9.6419

test data are under 6 μm, which indicates that the model created using HTGA has very good predictive ability. A comparison of these results with those of a previous study(46) showed that the model created using HTGA was better than that made using trial and error for parameter adjustment. It is clear that the optimization of BPNN training parameters using HTGA can effectively improve the adjustment of parameters as well as enhance the predictive ability of the model.

4. Discussion

To mitigate the expenses and complexities associated with adapting a thermal error model to diverse CNC machines, Tseng and Ho(47) recommended the application of multivariable linear regression and nonlinear regression techniques for effective thermal error compensation. An increasing amount of research supports a similar approach for addressing the persistent challenge of thermal error in these machines, as indicated in other studies.(48,49) The advantage of the methods previously mentioned lies in their straightforward implementation when integrated with a digital signal processor (DSP). This integration facilitates user-friendly operation and efficient deployment in real-world applications, thereby reducing the technical barriers to adoption and application in the field. However, the downside is that the values of the error statistical indexes are just acceptable, or, in some cases, error statistical indexes are not included in the research for performance evaluation. An intuitive comparison is shown in Table 3. In this table, we can discover that the linear regression method is easy to implement with an acceptable compensation performance, but according to the results of the above-mentioned research, the prediction will start losing its accuracy when predicting error change in a short period of time.

Owing to the unsatisfactory results of error change in a short period of time, the neural network is introduced in this paper to tackle the issue. Under the circumstances of not using multiple hidden layers or deep learning, this method can still achieve easy implementation and high accuracy for short-term predictions. The neural network relies on the initial weighting and training method, which requires experience and time for trial-and-error tuning. Therefore, we not only focus on finding the suitable method for thermal error prediction but also the tuning and training method of the neural network. For example, Li *et al.*(7) proposed IPSO to search initial weighting and BPNN parameters. Table 4 is a review of the studies using different models on the

Table 3
Review and comparison of linear regression methods in thermal error prediction research.

Linear regression type	Advantages	Prediction results (only *z*-axis is considered)
Multivariable linear regression[47] Nonlinear regression[47] General linear model[48]	Because of the simplicity of the models, it is easier to implement it into DSP.	Around 6 μm Around 3 μm Rise time was within 19 μm Stable time was within 15 μm.
Nonlinear regression[49]	The errors derived from the nonlinear regression approach are tenfold lower than those obtained through multiple linear regression analysis[49]	1 to 2 μm
Be advised that since the experiment equipment and the signal gathering conditions are different, the prediction results are for reference only.		

Table 4
Prediction performances of different methods on same CNC machine.

Method	For real-time and low-cost implementation Y for yes and N for no (Baseline is Arduino Due)	Prediction results (only the *z*-axis is considered)
Multivariate regression analysis[12]	Y	6.8 μm
General regression neural network[12]	N (highly complex to compute, and all testing samples are required to train with all train samples)	0.1 μm
General linear model[48]	Y	Rise time was within 19 μm. Stable time was within 15 μm.
Fine Gaussian SVM[50]	N (requires enough computational power for real-time application)	0.1 μm
This study (BP-GA)	Y	40 μm
This study (BP-HTGA)	Y	6 μm

same CNC machine. Although the results show that the general regression neural network has the best accuracy, this model is only suitable for training with a small number of samples, and it is difficult to realize it in real time. Besides, fine Gaussian SVM also rarely attains real-time performance on low-cost computing boards. The BP-HTGA in this paper shows its worth in its high accuracy and easy implementation characteristic.

The results in Sect. 3.1 show the temperature variation and thermal displacement of the spindle at 3000 and 7500 rpm. Figure 7 shows that HTGA has a clear advantage with better convergence and is successful in this scenario. Additionally, the results also show that the optimal number of neurons to find the hidden layer of BPNN is 15, the optimal training function is Levenberg–Marquardt, and the optimal learning rate is set to 0.5 in this scenario. The results in Sect. 3.3, which include the mean-square error and root-mean-square error metrics for both training and testing datasets at different spindle speeds, indicates that the BP-HTGA outperforms the model optimized by the standard GA. From the results of the experiments, it can be verified that the test data of this spindle at all speeds have relatively small error, and the optimized solution targeted in this study is superior to these of common GA methods. Many researchers have conducted similar experiments with various optimization methods, and the proposed method has been proven to yield the best or superior experimental results.[50–52] This study proved that the HTGA, GA, and PSO methods are suitable for this experimental scenario, and HTGA, in particular, gave better results. For a more objective verification of the results, a 10-fold cross-validation was used to build the model, thus proving that this process can be applied to build the thermal compensation model in this application scenario if more researchers develop better optimization methods in the future.

5. Conclusions

The methods used to search for the optimal training parameters of BPNN using HTGA were investigated. This was conducted to enhance the predictive accuracy of the model and solve the difficult problem of BPNN parameter adjustment. The 10-fold cross-validation used for the verification of the model can objectively assess the impact of training parameters and create an optimal thermal error model of the spindle. The experiments showed that the proposed application of HTGA can reduce the time needed for BPNN training parameter adjustment and optimize the BPNN spindle thermal deformation model. The HTGA model was better at finding the optimal training parameters of BPNN than the model created using GA. The BP-HTGA model prediction error was 6 μm compared with 10 μm for the BP-GA model. BP-HTGA can optimize parameters faster and give better prediction than BP-GA. The BP-HTGA model can reduce errors of 40 to 6 μm to enhance prediction accuracy as well as save time in parameter adjustment. Therefore, we proposed to use the BP-HTGA approach to find the best training parameters to build a better model to solve the temperature-induced lift-length compensation problem. Our experimental results confirm the advantage of using the optimization method to search the parameters of BPNN. Aside from our method, other well-known optimization methods can also be used, such as MBO, EWA, EHO, MS algorithm, SMA, and HHO.

References

1 Y. Li, W. H. Zhao, S. H. Lan, J. Ni, W. W. Wu, and B. H. Lu: Int. J. Mach. Tools Manuf. **95** (2015) 20. https://doi.org/10.1016/j.ijmachtools.2015.04.008

2 J. Bryan: CIRP Ann. **39** (1990) 645. https://doi.org/10.1016/s0007-8506(07)63001-7

3 W. Xian, G. Feng, Z. Jingdong, and Y. Wang: Proc. IEEE Int. Conf. Cloud Computing and Big Data Analysis (IEEE, 2016) 241. https://doi.org/10.1109/ICCCBDA.2016.7529594.

4 A. M. Abdulshahed, A. P. Longstaff, and S. Fletcher: Appl. Soft Comput. **27** (2015) 158. https://doi.org/10.1016/j.asoc.2014.11.012

5 T. C. Chen, C. J. Chang, J. P. Hung, R. M. Lee, and C. C. Wang: Appl. Sci-Basel. **6** (2016) 101. https://doi.org/ARTN10110.3390/app6040101

6 Y. Ling, C. Jihong, L. Haizhou, T. Hongliang, and M. Xinyong: Proc. Int. Conf. Computational and Information Sciences Conf. (IEEE, 2010) 861–864.

7 B. Li, X. T. Tian, and M. Zhang: Int. J. Adv. Manuf. Technol. **105** (2019) 1497. https://doi.org/10.1007/s00170-019-04375-w

8 K. C. Wang: J. Grey Syst. **22** (2010) 353.

9 R. J. Liang, W. H. Ye, H. Y. H. Zhang, and Q. F. Yang: Int. J. Adv. Manuf. Technol. **63** (2012) 1167. https://doi.org/10.1007/s00170-012-3978-6

10 Y. Li, J. Zhao, and S. J. Ji: Int. J. Adv. Manuf. Technol. **97** (2018) 2575. https://doi.org/10.1007/s00170-018-1978-x

11 F. Tan, M. Yin, L. Wang, and G. F. Yin: Int. J. Adv. Manuf. Technol. **94** (2018) 2861. https://doi.org/10.1007/s00170-017-1096-1

12 B. L. Jian, C. C. Wang, C. T. Hsieh, Y. P. Kuo, M. C. Houng, and H. T. Yau: Int. J. Adv. Manuf. Technol. **104** (2019) 466. https://doi.org/10.1007/s00170-019-04261-5

13 P. C. Tseng and S. L. Chen: JSME Int. J. Ser. C **45** (2002) 470. https://doi.org/10.1299/jsmec.45.470

14 Z. H. Liu, B. Yang, C. Ma, S. L. Wang, and Y. F. Yang: Int. J. Adv. Manuf. Technol. **106** (2020) 5001. https://doi.org/10.1007/s00170-020-04957-z

15 W. H. Ye, Y. X. Guo, H. F. Zhou, R. J. Liang, and W. F. Chen: Adv. Manuf. **8** (2020) 119. https://doi.org/10.1007/s40436-020-00293-3

16 F. Liang, J. M. Gao, and L. Xu: Int. Commun. Heat Mass Transfer. **112** (2020) 104502. https://doi.org/10.1016/j.icheatmasstransfer.2020.104502

17 L. Y. Chuang, C. S. Yang, H. S. Yang, and C. H. Yang: JMIR Med. Inform. **8** (2020) e16886. https://doi.org/10.2196/16886

18 H. R. Moetamedzadeh, E. Khanmirza, and R. Madoliat, Int. J. Nonlin. Sci. Num. **21** (2019) 51. https://doi.org/10.1515/ijnsns-2018-0093

19 W. L. Chu, M. J. Xie, L. W. Wu, Y. S. Guo, and H. T. Yau: IEEE Access **8** (2020) 169576. https://doi.org/10.1109/Access.2020.3022648

20 J. T. Tsai, T. K. Liu, and J. H. Chou: IEEE Trans. Evol. Comput. **8** (2004) 365. https://doi.org/10.1109/Tevc.2004.826895

21 An Idea Based on Honey Bee Swarm for Numerical Optimization: http://abc.erciyes.edu.tr/pub/tr06_2005.pdf (accessed November 2023)

22 Y. H. Feng, S. Deb, G. G. Wang, and A. H. Alavi: Expert Syst. Appl. **168** (2021) 114418. https://doi.org/10.1016/j.eswa.2020.114418

23 G. G. Wang, D. Suash, and L. D. S. Coelho: Int. J. Bio-Inspired Comput. **12** (2018) 1. https://doi.org/10.1504/ijbic.2015.10004283

24 G. G. Wang, S. Deb, and L. D. S. Coelho: Proc. 2015 3rd Int. Symp. Computational and Business Intelligence Conf. (ISCBI, 2015). https://doi.org/10.1109/ISCBI.2015.8

25 G. G. Wang: Memet. Comput. **10** (2018) 151. https://doi.org/10.1007/s12293-016-0212-3

26 S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili: Future Gener. Comput. Syst. **111** (2020) 300. https://doi.org/10.1016/j.future.2020.03.055

27 A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen: Future Gener. Comput. Syst. **97** (2019) 849. https://doi.org/10.1016/j.future.2019.02.028

28 Y. Luo, B. Lin, and C. B. Wen: Digital Chin. Med. **1** (2018) 84. https://doi.org/10.1016/s2589-3777(19)30010-2

29 X. Yang, Q. Zhou, J. Wang, L. Han, R. Zhou, Y. He, and K. C. Li: Ann. Nucl. Energy **132** (2019) 576. https://doi.org/10.1016/j.anucene.2019.06.034

30 C. Jin, S. W. Jin, and L. N. Qin: Appl. Soft Comput. **12** (2012) 2147. https://doi.org/10.1016/j.asoc.2012.03.015

31 B. Nassih, A. Amine, M. Ngadi, and N. Hmina: Procedia Comput. Sci. **148** (2019) 116. https://doi.org/10.1016/j.procs.2019.01.015

32 T. Fushiki: Stat. Comput. **21** (2009) 137. https://doi.org/10.1007/s11222-009-9153-8

33 T. T. Wong, and N. Y. Yang: IEEE Trans. Knowl. Data Eng. **29** (2017) 2417. https://doi.org/10.1109/Tkde.2017.2740926

34 G. Taguchi: Commun. Stat. Theory Methods **14** (1985) 2785. https://doi.org/10.1080/03610928508829076

35 J. A. Ghani, I. A. Choudhury, and H. H. Hassan: J. Mater. Process Tech. **145** (2004) 84. https://doi.org/10.1016/S0924-0136(03)00865-3

36 T. Thankachan, K. S. Prakash, R. Malini, S. Ramu, P. Sundararaj, S. Rajandran, D. Rammasamy, and S. Jothi: Appl. Surf. Sci. **472** (2019) 22. https://doi.org/10.1016/j.apsusc.2018.06.117

37 J. Ashburner: Neuroimage **38** (2007) 95. https://doi.org/10.1016/j.neuroimage.2007.07.007

38 M. E. Tipping: J. Mach. Learning Researh **1** (2001) 211. https://doi.org/10.1162/15324430152748236

39 P. E. Gill, W. Murray, and M. H. Wright: Practical Optimization (Society for Industrial and Applied Mathematics, Philadelphia, 2019) Chap. 4.

40 M. Riedmiller, and H. Braun: Proc. IEEE Int. Conf. Neural Networks (IEEE, 1993).

41 M. F. Møller: Neural Networks **6** (1993) 525. https://doi.org/10.1016/s0893-6080(05)80056-5

42 M. J. D. Powell: Math. Program. **12** (1977) 241. https://doi.org/10.1007/bf01593790

43 L. E. Scales: Introduction to Non-Linear Optimization (Springer-Verlag, 1985) Chap. 3.

44 R. Battiti: Neural Comput. **4** (1992) 141. https://doi.org/10.1162/neco.1992.4.2.141

45 Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner: Proc. IEEE **86** (1998) 2278. https://doi.org/10.1109/5.726791

46 B. L. Jian, Y. S. Guo, C. H. Hu, L. W. Wu, and H. T. Yau: Sens. Mater. **32** (2020) 431. https://doi.org/10.18494/Sam.2020.2606

47 P. C. Tseng, and J. L. Ho: Int. J. Adv. Manuf. Technol. **19** (2002) 850. https://doi.org/10.1007/s001700200096

48 C. J. Lin, X. Y. Su, C. H. Hu, B. L. Jian, L. W. Wu, and H. T. Ya: Energies **13** (2020) 949. https://doi.org/10.3390/en13040949

49 N. Wessel, J. Konvicka, F. Weidermann, S. Nestmann, R. Neugebauer, U. Schwarz, A. Wessel, and J. Kurths: Chaos **14** (2004). 23. https://doi.org/10.1063/1.1622351

50 G. Li, X. Tang, Z. Li, K. Xu, and C. Li: Precis. Eng. **73** (2022) 140. https://doi.org/10.1016/j.precisioneng.2021.08.021

51 Z. Li, Q. Wang, B. Zhu, B. Wang, W. Zhu, and Y. Dai: Case Stud. Therm. Eng. **39** (2022) 102432. https://doi.org/10.1016/j.csite.2022.102432

52 Z. Li, B. Wang, B. Zhu, Q. Wang, and W. Zhu: Case Stud. Therm. Eng. **38** (2022) 102326. https://doi.org/10.1016/j.csite.2022.102326

About the Authors

Chi-Yuan Lin received his B.S. degree in electronic engineering from National Taiwan University of Science and Technology, New Taipei City, Taiwan, in 1988, M.S. degree in electronic and information engineering from National Yun Lin University of Science and Technology, Douliu, Taiwan, in 1998, and Ph.D. degree in electrical engineering from National Cheng Kung University, Tainan City, Taiwan, in 2004. He is currently a professor at the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung City, Taiwan. His research interests include neural networks, multimedia coding, data hiding, IoT smart applications, and master control unit creative applications.
(chiyuan@ncut.edu.tw)

Shu-Cing Wu received his M.S. degree from the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung City, Taiwan, in 2019, where he is currently pursuing his Ph.D. degree under the Prospective Technology of Electrical Engineering and Computer Science Program. He serves as a Technical Consultant with Kou-Yi Electronic, Taichung City. His current research interests include embedded system development; integration of communication applications, such as RFID, Bluetooth, and Wi-Fi; personal computer (PC)-based human–machine interface programming; software development; peripheral system circuit; PCB design; deep learning; and machine learning model training and its applications.

Chi-Hsien Hu received his M.S. degree from the Department of Electrical Engineering, National Chin-Yi University of Technology in 2019.

Bo-Lin Jian received his B.S. degree from the Department of Electrical Engineering, National Formosa University, in 2009 and M.S. degree in materials science and engineering from National Taiwan University of Science and Technology in 2011. His Ph.D. degree was awarded by the Department of Aeronautics and Astronautics, National Cheng Kung University in 2017. He is an associate professor in the Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung, Taiwan. His current research interests include signal and image processing, machine learning, and control systems. (BoLin@ncut.edu.tw)