Sensors and Materials, Vol. 34, No. 2 (2022) 819–833
MYU Tokyo

S & M 2855

Design of an Intelligent Grinding Parameter Selection Assistance System

Jyun-Yu Jhang[1,2] and Cheng-Jian Lin[1,3*]

[1]College of Intelligence, National Taichung University of Science and Technology, Taichung 404, Taiwan
[2]Department of Computer Science and Information Engineering,
National Taichung University of Science and Technology, Taichung 404, Taiwan
[3]Department of Computer Science and Information Engineering,
National Chin-Yi University of Technology, Taichung 411, Taiwan

(Received September 6, 2021; accepted December 20, 2021)

Keywords: grinding, convolutional neural network, surface roughness, Taguchi method, differential evolution

In this study, an intelligent grinding parameter selection assistance system (IGPSAS) that can be used by operators for grinding was designed. In the data collection stage, an ESG-1020 surface grinder and aluminum were used for grinding experiments. The proposed IGPSAS consists of two parts: a Taguchi-based convolutional neural network (TCNN) and a differential evolution algorithm. First, the proposed TCNN was used to establish a surface roughness prediction model. Then, the proposed differential evolution algorithm was used to determine the best processing parameters. To achieve better surface smoothness prediction capabilities in the CNN model, the Taguchi method was used to optimize the parameters of the network model architecture. The effect of each factor was analyzed, and a network with stable parameters was selected for machine processing. The performance of the proposed TCNN was verified experimentally. The mean average percentage error (MAPE) of the proposed TCNN's surface roughness prediction in the measurement of a NewView 8300 optical surface profile was 15.65%. In addition, the differential evolution algorithm was used to select the best processing parameters and perform actual processing. The MAPE of the surface roughness prediction of the proposed IGPSAS was experimentally determined to be 10.97%, demonstrating that the system effectively provides the user with the ability to operate the machine with the parameters set according to the desired processing quality.

1. Introduction

Driven by Industry 4.0, Internet of Things and artificial intelligence applications are being widely employed in industrial manufacturing, and intelligent processing has become vital.(1,2) Surface roughness, which can directly affect product life, is frequently used as a quality evaluation standard. Many scholars have conducted experiments on optimizing process parameters to improve surface roughness, and many manufacturers now provide a user-friendly

*Corresponding author: e-mail: cjlin@ncut.edu.tw
https://doi.org/10.18494/SAM3643

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

interface to assist operators. However, process parameter setting still relies on the experience of the operator. An unsuitable process parameter can result in low product quality and long postprocessing time. With the development of precision machining technology, establishing a stable and highly accurate model has become vital. A model for optimizing target parameters can be established through several approaches, such as mathematical models, evolutionary algorithms, artificial neural networks (ANNs), and convolutional neural networks (CNNs). Wei *et al.*(3) used the fractal root mean square deviation parameter to establish a mathematical model for surface roughness and proposed a relationship between this factor and fractal root mean square deviation. Kumar *et al.*(4) used regression modeling to analyze the correlation between process parameters and surface roughness to determine the optimal process parameters. Baskar *et al.*(5) applied multiobjective functions to establish mathematical models and adopted the ant colony algorithm to optimize model parameters. Knowledge of the machines being modeled is required when using mathematical models, and ANNs can be an alternative method for establishing such modeling.

Mitra and Ghivari(6) used an ANN model with various numbers of hidden-layer neurons and learning rates to develop an improved model architecture for grinding operations in a lead–zinc ore beneficiation plant. Sizemore *et al.*(7) established a surface roughness prediction model by adopting machine learning (ML) methods and ANN modeling.(7) The experimental results showed that the ANN model yielded smaller errors. Chen *et al.*(8) used an ANN model and linear regression models for their experiments. The ANN model was trained by inputs such as processing parameters, cutting force direction, and vibration force, and the output was surface roughness. The experimental results revealed that the ANN model fit the results better than linear regression models. The results of the aforementioned methods show that data-driven methods combined with ML can be employed to build models effectively. Moreover, many scholars(9-11) have proposed evolutionary algorithms to optimize their self-designed models, such as improved particle swarm optimization, the ant colony optimization algorithm, and the differential evolution (DE) algorithm. However, evolutionary algorithms require more iterations and longer computing times to adjust neural networks with many parameters. By contrast, backpropagation methods can adjust model parameters effectively and offer advantages of high learning accuracy, simple implementation, highly fitting nonlinear functions, and relatively stable training results.

In industrial manufacturing, expert or decision-making systems(12) can provide suitable processing parameters and effectively improve processing quality and time. Lim and Chang(13) designed a predictive model that used an ML method to enable operators to evaluate the effects of processing parameters on surface roughness, thereby stabilizing product quality. However, these methods rely on high-precision predictive models; if the accuracy of the prediction model used is poor, the processing quality will be less than expected. In this study, to design a highly stable model structure, the design of experiment (DoE) method has been introduced.(14–18) Compared with relying on experience and trial and error for network architecture design, the DoE method provides improved stability and accuracy, and assists the operator to quickly and conveniently set the processing parameters.(19,20) In the present study, an intelligent grinding parameter assistance system (IGPSAS) was designed to provide suitable processing parameters according to quality requirements. The IGPSAS mainly consists of two parts: (1) a Taguchi-based CNN (TCNN) for surface roughness prediction and (2) a process parameter optimization system based on the DE algorithm. Through the IGPSAS, suitable processing parameters according to the surface roughness required by the operator can be attained.

The rest of this article is organized as follows. In Sect. 2, the experimental equipment and arrangement are discussed. In Sect. 3, the developed IGPSAS is presented. The experimental results and discussion are detailed in Sect. 4. Finally, conclusions are presented in Sect. 5.

2. Experimental Arrangement

The proposed IGPSAS was applied to a high-precision surface grinder, and a 3D optical profiler was used to measure surface roughness. The data acquired during grinding were used as the training data for the TCNN.

2.1 Surface grinder

An ESG-1020 surface grinder (EQUIPTOP, Taiwan), which comprises a Syntec MA11 controller, was used for the grinding experiments (Fig. 1 and Table 1). Details of the grinding wheel and cutting fluid are presented in Table 2.

2.2 Surface roughness measurement

After performing the grinding experiments, the surface roughness of the workpieces was measured using a 3D optical profiler (NewView 8300, Zygo, UK; Fig. 2). This optical profiler has a wide measurement range, enabling the accurate measurement of the surface roughness. The profiler specifications are presented in Table 3.

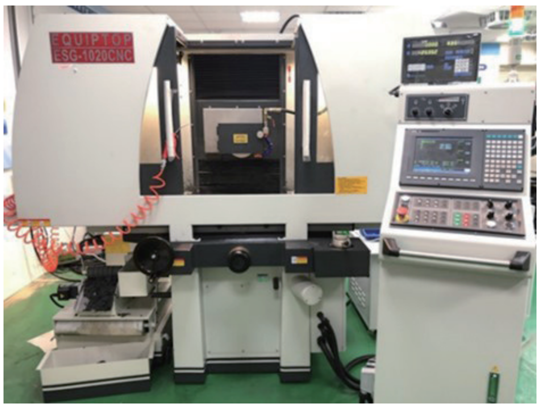


Fig. 1. (Color online) ESG-1020 surface grinder used in the study.

Table 1
ESG-1020 surface grinder specifications.

Controller	Syntec MA11
Workbench area (mm)	500 × 270
Workbench speed (M/min)	5 to 25
Highest speed (mm/min)	1100/550
Wheel size (mm^2)	250 × 31.75
Wheel speed (RPM)	12000

Table 2
Specifications of consumables used in processing.

Workpiece	Material	Medium-carbon steel plate (S45C)
	Size (mm^2)	235 × 60
	Manufacturing company	China Steel Corporation
Wheel	Material	Aluminum oxide
	Model	WA46, WA60, WA80, WA100, WA120
	Size (mm^3)	205 × 13 × 31.75
	Manufacturing company	KINIK
Cutting fluid	Model	B-Cool 9665
	pH	9.5
	Manufacturing company	Blaser Swisslube

Fig. 2. (Color online) 3D optical profiler used in the study.

Table 3
3D optical profiler specifications.

Model	NewView™ 8300
Travel X/Y (mm)	150/150
Measurement array (pixel)	1024 × 1024
Field of view (mm)	0.04 to 16
Optical zoom	1.0×
Environment temperature (°C)	15–30

2.3 Experimental data acquisition

To establish a high-precision predictive model, data corresponding to different grinding conditions were collected. Five cases, each corresponding to a different number of grinding wheels, were included for data collection. Eight hundred and eighty data were captured in total, including the processing parameters such as feed rate, grinding wheel speed, the cutting depth of rough machining, the cutting depth of finish machining, and cutting pitch. For model prediction, 70 and 30% of the data were used as training and testing data, respectively.

3. IGPSAS

The IGPSAS provides suitable process parameters according to the operator requirements. The IGPSAS process block diagram is shown in Fig. 3. The operator only needs to input information regarding the material, wheel type, and desired surface roughness (Ra), then the optimized processing parameters are generated by the IGPSAS. After grinding is completed, a 3D optical profiler is used to verify the surface roughness.

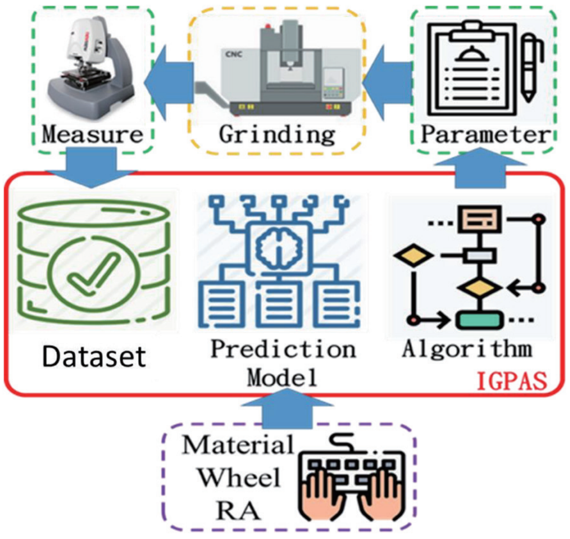


Fig. 3. (Color online) IGPSAS process block diagram.

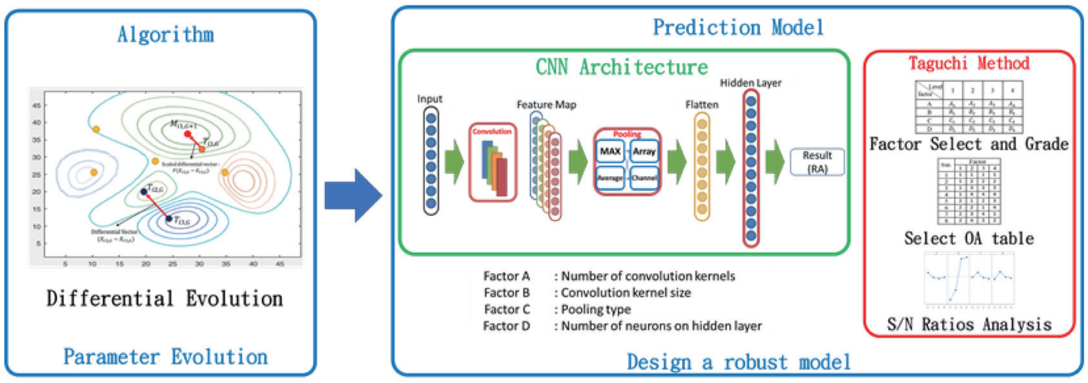


Fig. 4. (Color online) Architecture of IGPSAS.

The architecture of the IGPSAS can be divided into two parts (Fig. 4): (1) the TCNN prediction model proposed in this study and (2) the DE algorithm for parameter optimization. In the prediction model, the feature extraction ability of the convolution operation was introduced to further improve the accuracy of the model. The Taguchi optimization technique was then applied to adjust a variety of model architectures, such as the number of convolution kernels, pooling methods, and the number of hidden-layer neurons. The effect of each factor was analyzed to select the most stable factor and establish the most robust TCNN architecture. After the model was established, a DE algorithm was employed to determine the optimal process parameters. Each process parameter was evaluated using the trained TCNN during evolution. Generally, when the predicted RA value meets an operator's setting, the TCNN outputs the optimal process parameters. The TCNN prediction model and the DE algorithm are described in this section.

3.1 TCNN

CNNs[(20–23)] are widely used in various fields, such as image classification and sensor signal analysis. A CNN typically includes a convolutional layer, a pooling layer, and a fully connected layer. Feature extraction is performed in the convolutional layer, and feature extraction and parameter selection assistance are performed in the pooling layer. The fully connected layer comprises numerous neurons and simulates nonlinear functions. However, designing a good network architecture involves extensive experimentation and high time costs. By using the Taguchi experimental design method, the number of experiments can be reduced and the factors that affect the results of the experiment can be determined. The Taguchi method includes four steps: 1) defining the objective function, 2) selecting the orthogonal array based on the factors and levels, 3) conducting the experiments, and 4) analyzing the experimental results. The Taguchi method uses the signal-to-noise (S/N) ratio to determine the optimal parameters by observing the interactions between various factors. In this study, the convolution kernels, pooling methods, and the number of hidden-layer neurons were considered influential factors. Each factor was divided into four levels, as listed in Table 4. The $L_{16}(4^5)$ orthogonal array given in Table 5 was selected for setting the parameters. The Taguchi method reduced the number of experiments from 256 (4^4) to 16 (4^2) times.

3.2 Process parameter optimization using DE algorithm

Evolutionary algorithms emulate the survival and competition strategies of animals to solve scientific and engineering problems optimally. Among them, the DE algorithm offers advantages of a simple structure, relatively few parameters, and good performance in various applications.

Table 4
List of factors and levels used in the Taguchi method.

Factor	Level			
	1	2	3	4
A	A_1	A_2	A_3	A_4
B	B_1	B_2	B_3	B_4
C	C_1	C_2	C_3	C_4
D	D_1	D_2	D_3	D_4

Table 5
$L_{16}(4^5)$ orthogonal array.

Experiment	Factor			
	1	2	3	4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	1	4	4	4
5	2	1	2	3
6	2	2	1	4
7	2	3	4	1
8	2	4	3	2
9	3	1	3	4
10	3	2	4	3
11	3	3	1	2
12	3	4	2	1
13	4	1	4	2
14	4	2	3	1
15	4	3	2	4
16	4	4	1	3

Therefore, the DE algorithm was implemented in the IGPSAS to optimize the process parameters. The steps of the DE algorithm are detailed as follows.

- **Initialization**

The parameters of the DE algorithm were set and the target vector (*T*) of its solution space was initialized randomly. The equation is presented as

$$T_i = \left[T_{i,1}, T_{i,2}, \ldots, T_{i,D}\right],\ i = 1, 2, \ldots, NP, \tag{1}$$

where *NP* denotes the population size and *D* is the input dimension.

- **Mutation**

Three target vectors were randomly selected from the solution space, namely, $T_{r1}^{(G)}$, $T_{r2}^{(G)}$, and $T_{r3}^{(G)}$. Then, the mutation factor *F* was multiplied by the vector difference between $T_{r2}^{(G)}$ and $T_{r3}^{(G)}$. Next, a new mutated vector $M_1^{(G+1)}$ was generated. The equation is shown as

$$M_i^{(G+1)} = T_{r1}^{(G+1)} + F(T_{r2}^{(G)} - T_{r3}^{(G)})\,, \tag{2}$$

where $M_i^{(G+1)} = \left[M_{i,1}^{(G+1)}, M_{i,2}^{(G+1)}, \cdots, M_{i,D}^{(G+1)}\right]$ and *G* represents the generation. The mutation factor affects the convergence speed. Searching for possible solutions is easier when *F* is large, whereas a small *F* might cause the search to converge to local best solutions.

- **Recombination**

The crossover rate (*CR*) was set to enable the recombination of the target vectors *T* and *M* into a new vector $N_i^{(G+1)}$:

$$N_{i,D}^{(G+1)} = \begin{cases} M_{i,D}^{(G+1)}, & \text{if } rand(0,1) \le CR, \\ T_{i,D}^{(G)}, & \text{otherwise,} \end{cases} \tag{3}$$

where $rand\left(0,1\right)$ is a random variable in the range of 0–1. $N_i^{(G+1)} = \left[N_{i,1}^{(G+1)}, N_{i,2}^{(G+1)}, \cdots, N_{i,D}^{(G+1)}\right]$.

- **Selection**

The prediction model was evaluated using the fitness function. *T* and *N* were input into the model; if the fitness value of *N* was better than that of *T*, then *T* was replaced by *N*; otherwise, *T* was retained until the next iteration. The selection equation and fitness function are presented as

$$T_i^{(G+1)} = \begin{cases} N_i^{(G+1)}, & \text{if } Fit\left(N_i^{(G+1)}\right) < Fit\left(T_i^{(G)}\right), \\ T_i^{(G)}, & \text{otherwise,} \end{cases} \tag{4}$$

$$Fit(X) = \left|SetSR - Model(X)\right|, \tag{5}$$

where *SetSR* represents the target surface roughness predefined by the operator and *Model*(*X*) stands for the surface roughness predicted by the TCNN. The fitness function retains the minimal result of two differences.

4. Experimental Results and Discussion

4.1 TCNN architecture analysis of variance

Tables 6 and 7 respectively show the parameter setting and the results obtained upon inserting the influential factors of the CNN based on Tables 4 and 5. Different CNN architectures were trained; four evaluation functions, namely, mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and MAPE were applied in the study. These evaluation functions are defined in Eqs. (6)–(9). The experimental results are shown in Table 8. In addition, the analysis of variance was performed. The goal of the model is to predict the surface roughness closest to the operator's setting; therefore, the smaller the S/N ratio, the better the prediction. The S/N ratio equation is shown as Eq. (10).

Table 6
Model parameters and their levels.

Influential factor		Level			
		1	2	3	4
Convolution	Number of filters (A)	16	24	32	40
	Kernel size (B)	2	3	4	5
Pooling	Pooling type (C)	Average	Max	Average (Channel)	Max (Channel)
Hidden layer	Number of neurons (D)	20	25	30	35

Table 7
$L_{16}(4^5)$ orthogonal array.

Experiment	Factor			
	1	2	3	4
1	16	2	Average	20
2	16	3	Max	25
3	16	4	Average (Channel)	30
4	16	5	Max (Channel)	35
5	24	2	Max	30
6	24	3	Average	35
7	24	4	Max (Channel)	20
8	24	5	Average (Channel)	25
9	32	2	Average (Channel)	35
10	32	3	Max (Channel)	30
11	32	4	Average	25
12	32	5	Max	20
13	40	2	Max (Channel)	25
14	40	3	Average (Channel)	20
15	40	4	Max	35
16	40	5	Average	30

Table 8
Error and S/N ratios of TCNN.

Experiment	Evaluation functions					
	MSE	RMSE	MAE	S/N ratio	Standard deviation	Mean
1	0.006111	0.078174	0.043420	25.722036	0.036039	0.042568
2	0.001197	0.034488	0.018890	32.874371	0.016657	0.018192
3	0.000518	0.022663	0.012621	36.490011	0.011088	0.011934
4	0.000500	0.022362	0.012279	36.634964	0.010942	0.011714
5	0.005500	0.074162	0.041771	26.152873	0.034349	0.040478
6	0.004756	0.068963	0.037079	26.880113	0.032104	0.036933
7	0.000539	0.023098	0.013114	36.284488	0.011304	0.012251
8	0.000421	0.020477	0.011406	37.370717	0.010043	0.010768
9	0.005897	0.076787	0.044329	25.797082	0.035487	0.042338
10	0.004739	0.068839	0.037599	26.864878	0.032053	0.037059
11	0.000483	0.021960	0.012419	36.731651	0.010761	0.011621
12	0.000552	0.023422	0.013272	36.167501	0.011459	0.012416
13	0.006097	0.078085	0.045798	25.616193	0.036057	0.043327
14	0.004719	0.068691	0.036207	26.952472	0.031987	0.036539
15	0.000501	0.022238	0.012205	36.683965	0.010879	0.011648
16	0.000382	0.019464	0.011266	37.730323	0.009573	0.010371

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2 \tag{6}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2} \tag{7}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right| \tag{8}$$

$$MAPE = \left(\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|\right)*100\% \tag{9}$$

$$S/N = -10\ \log\left(\frac{1}{n}\Sigma y_i^{\,2}\right) \tag{10}$$

As shown in Table 9 and Fig. 5, the most significant factor affecting the accuracy of the model was the size of the convolution kernel, followed by the pooling method, the number of hidden-layer neurons, and the number of convolution kernels. The best performance of the TCNN was achieved using 16 convolution kernels with a size of 5, 25 hidden layers and average

Table 9
Response table for S/N ratios.

Level	A	B	C	D
1	**32.93**	25.82	31.77	31.28
2	31.67	28.39	**32.97**	**33.15**
3	31.39	36.55	31.65	31.81
4	31.75	**36.98**	31.35	31.5
Delta	1.54	11.15	1.62	1.87
Rank	4	1	3	2

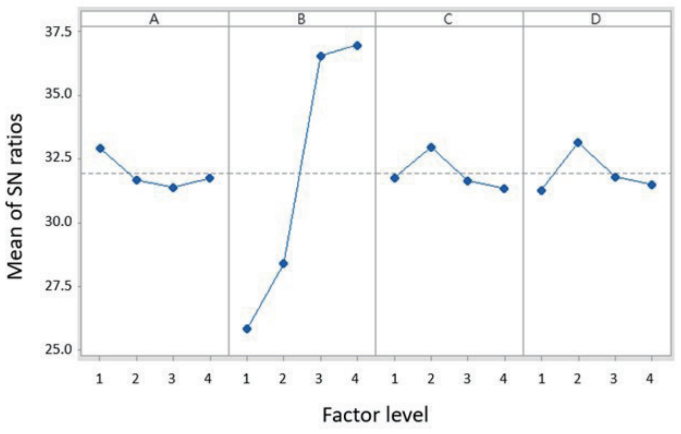
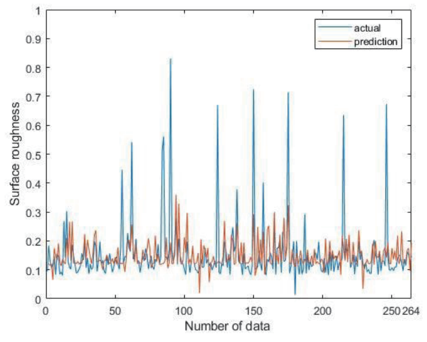


Fig. 5. (Color online) Plot of S/N ratios.

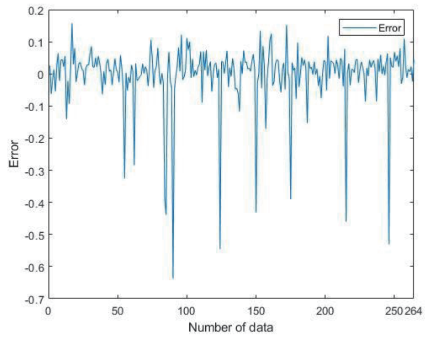
Table 10
Original and robust model error comparison.

Model architecture	MSE	RMSE	MAE	MAPE (%)
ANN(7)	0.009723	0.098609	0.052348	33.02
DNN(8)	0.008232	0.090732	0.045764	28.28
1DCNN	0.001213	0.034825	0.025672	18.97
TCNN	0.000590	0.024298	0.018698	15.65

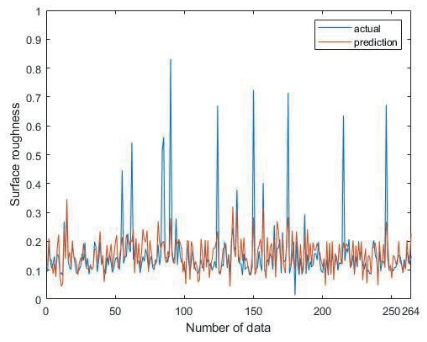
pooling (Fig. 5). The performance of the model is shown in Table 10 and Fig. 6, and the details of the TCNN architecture are presented in Table 11. As shown in Table 10, the experimental results of the ANN and DNN methods without a convolution operation have an MAPE higher than 25%, which means that the accuracy of the established prediction model is insufficient. The MAPE of 1DCNN with the convolution operation is 18.97%. This means that the convolution operation can effectively perform feature extraction and improve the accuracy of the model. On the other hand, the proposed TCNN using the Taguchi method can effectively decrease the MAPE from 18.97 to 15.65%, showing that the Taguchi method optimized the model architecture.



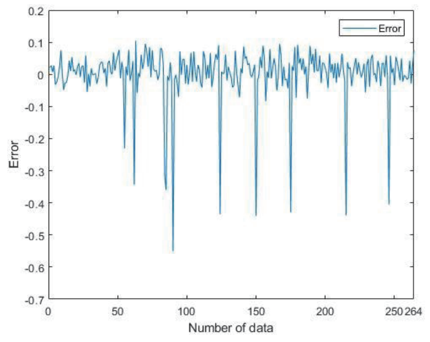
(a)



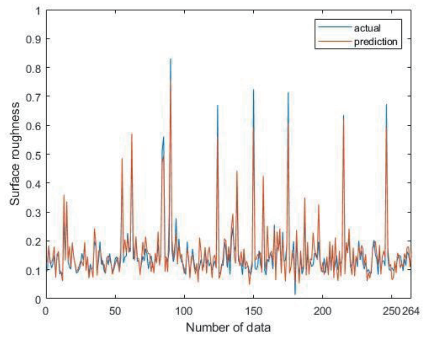
(e)



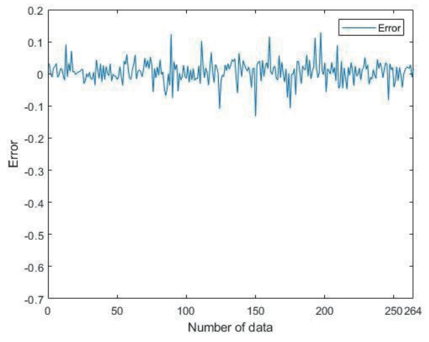
(b)



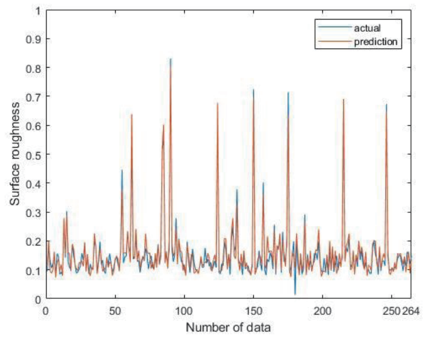
(f)



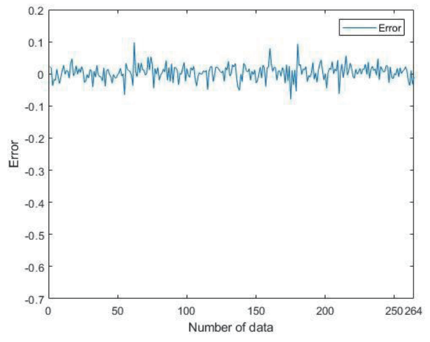
(c)



(g)



(d)



(h)

Fig. 6. (Color online) Performance of testing data. (a) ANN, (b) DNN, (c) 1DCNN, (d) TCNN, (e) error of ANN, (f) error of DNN, (g) error of 1DCNN, and (h) error of TCNN.

Table 11
Final TCNN architecture.

Layer	Size
Input	1 × 6
Conv1 (Size, Channel)	1 × 5, 24
Conv2 (Size, Channel)	1 × 5, 24
Average pooling	1 × 2
Fully connected	1 × 25
Output	1

4.2 Process parameter optimization

In this study, the DE algorithm was used to select the appropriate processing parameters for the operators. Table 12 shows the parameters of the DE algorithm, namely, population size (*NP*), mutation factor (*F*), crossover rate (*CR*), and the number of generations. The ranges of the process parameters, which are based on the physical limitations of the grinding machine and material properties, are listed in Table 13. The DE algorithm evolved within the ranges shown in Table 13 to generate the optimal process parameters.

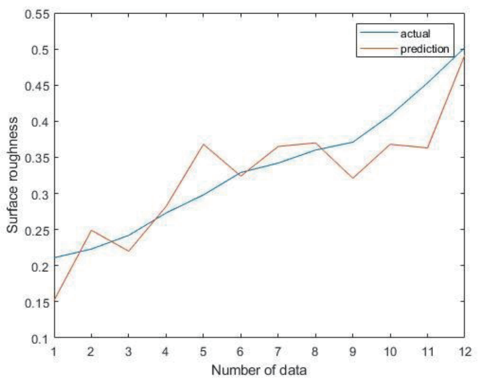
Twelve surface roughness values between 0.2 and 0.5 μm were set by the operator for model verification. The IGPSAS generated the suggested processing parameters and a grinding machine was used to conduct the experiments. The experimental results are shown in Fig. 7 and Table 14; the MSE, RMSE, MAE, and MAPE were used to evaluate the performance of the model. The experimental results also confirmed that the IGPSAS can effectively provide the correct processing parameters to operators, and the MAPE was 10.97%, indicating that the TCNN proposed in this study achieves high accuracy in surface roughness prediction.

Table 12
Initial parameters of DE.

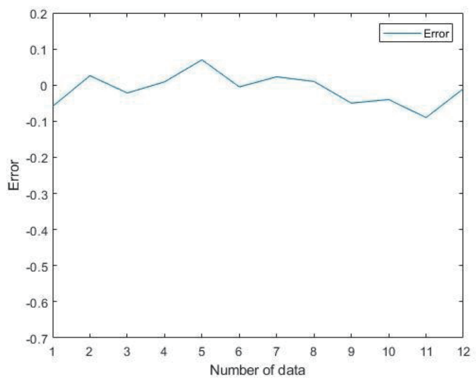
Population size (*NP*)	Mutation factor (*F*)	Crossover rate (*CR*)	Number of generations
20	0.5	0.5	300

Table 13
Processing parameter limits.

Suggested limits	Lower	Upper
Rotation speed (RPM)	2000	3500
Feed rate (M/min)	2	18
Roughing process depth (μm)	10	20
Finishing process depth (μm)	1	10
Cutting pitch (μm)	1	4



(a)



(b)

Fig. 7. (Color online) Experimental results of processing: (a) prediction and actual (b) error.

Table 14
Prediction results of proposed IGPSAS.

No.	Target Ra	Actual Ra	MSE	RMSE	MAE	MAPE (%)
1	0.211	0.152	0.001881	0.043374	0.0345	10.97
2	0.223	0.249				
3	0.242	0.220				
4	0.273	0.282				
5	0.298	0.368				
6	0.329	0.324				
7	0.342	0.365				
8	0.360	0.370				
9	0.371	0.321				
10	0.408	0.368				
11	0.453	0.363				
12	0.502	0.492				

5. Conclusions

In this study, we found that the processing parameters affect the product quality during grinding. The conventional processing parameter setting method relies on either the operator's experience or the machine manufacturer's default settings and may result in low-quality products. Therefore, the IGPSAS described herein was proposed to improve the product quality and achieve operational flexibility, thereby decreasing the operational difficulties associated with grinding. The IGPSAS, which provides the process parameters as per the needs of the operator, employed the TCNN to establish the corresponding model of processing parameters and surface smoothness. The TCNN was optimized using the Taguchi method through experiments to establish a better model architecture. Moreover, a DE algorithm was introduced to determine the best process parameters to provide operators with well-defined process parameters. To verify the performance of the proposed IGPSAS, a practical grinding process was performed and validation data were acquired using an optical profiler. The experimental results achieved a 10.97% MAPE for surface roughness, thus confirming that the proposed system can effectively assist operators in obtaining process parameters according to the desired surface roughness.

Only 880 data were captured in this study, which is insufficient. Therefore, to establish a high-precision predictive model, we will collect more data corresponding to different grinding conditions in future work. In addition, to apply the proposed IGPSAS to different machines without spending too much training time for processing modeling, we will apply the transfer learning technology to the proposed IGPSAS in future work.

Acknowledgments

The authors would like to thank the Ministry of Science and Technology of the Republic of China, Taiwan for financially supporting this research under Contract No. MOST 110-2218-E-005-016.

References

1 E. Manavalan and K. Jayakrishna: Comput. Ind. Eng. **127** (2019) 925. https://doi.org/10.1016/j.cie.2018.11.030
2 J. Chen, P. Hu, H. Zhou, J. Yang, J. Xie, Y. Jiang, Z. Gao, and C. Zhang: Engineering **5** (2019) 679. https://doi.org/10.1016/j.eng.2019.07.018
3 S. Wei, H. Zhao, and J. Jing: Appl. Surf. Sci. **357** (2015) 139. https://doi.org/10.1016/j.apsusc.2015.08.230
4 A. Kumar, S. K. Pradhan, and V. Jain: Mater. Today Proc. **27** (2020) 2724. https://doi.org/10.1016/j.matpr.2019.12.191
5 N. Baskar, R. Saravanan, P. Asokan, and G. Prabhaharan: Int. J. Adv. Manuf. Technol. **23** (2004) 311. http://dx.doi.org/10.1007/s00170-002-1533-6
6 K. Mitra and M. Ghivari: Comput. Chem. Eng. **30** (2006) 508. https://doi.org/10.1016/j.compchemeng.2005.10.007
7 N. E. Sizemore, M. L. Nogueira, N. P. Greis, and M. A. Davies: Procedia Manuf. **48** (2020) 1029. https://doi.org/10.1016/j.promfg.2020.05.142
8 Y. Chen, R. Sun, Y. Gao, and J. Leopold: Measurement **98** (2017) 25. http://dx.doi.org/10.1016/j.measurement.2016.11.027
9 S. Khalilpourazari and S. Khalilpourazary: Swarm Evol. Comput. **38** (2018) 173. http://dx.doi.org/10.1016/j.swevo.2017.07.008
10 A. M. Hemeida, S. A. Hassan, A. A. Mohamed, S. Alkhalaf, M. M. Mahmoud, T. Senjyu, and A. B. El-Din: Ain Shams Eng. J. **11** (2020) 659. https://doi.org/10.1016/j.asej.2020.01.007
11 C. Ciancio, G. Ambrogio, F. Gagliardi, and R. Musmanno: Neural Comput. Appl. **27** (2016) 2001. https://doi.org/10.1007/s00521-015-1994-9
12 N. V. V. S. Sudheer and K. K. Pavan: Procedia Mater. Sci. **6** (2014) 840. https://doi.org/10.1016/j.mspro.2014.07.101
13 T. G. Lim and W. R. Chang: Development of Human-Machine Interface for Automatic Control of Electro-Slag Remelting Process (IEEE, Pittsburgh, 2002) p. 501. https://doi.org/10.1109/IAS.2002.1044132
14 S. Shaji and V. Radhakrishnan: J. Mater. Process. Technol. **141** (2003) 51. https://doi.org/10.1016/S0924-0136(02)01112-3
15 E. S. Lee and S. Y. Baek: Int. J. Mach. Tools Manuf. **47** (2007) 509. https://doi.org/10.1016/j.ijmachtools.2006.06.007
16 S. Chatterjee, R. Rudrapati, P. Kumar pal, and G. Nandi: Mater. Today Proc. **5** (2018) 5272. https://doi.org/10.1016/j.matpr.2017.12.110
17 A. Saravanakumar, S. Dhanabal, E. Jayanand, and P. Logeshwaran: Today Proc. **5** (2018) 8131. https://doi.org/10.1016/j.pisc.2016.04.077
18 P. Mulimani and R. Bhat: Mater. Today:. Proc. **28** (2020) 2084. https://doi.org/10.1016/j.matpr.2020.03.230
19 C. J. Lin and Y. C. Li: Electronics **9** (2020) 1066. https://doi.org/10.3390/electronics9071066
20 F. Chou, Y. Tsai, Y. Chen, J. Tsai, and C. Kuo: IEEE Access **7** (2019) 68316. https://doi.org/10.1109/ACCESS.2019.2918563

21 A. P. Rifai, H. Aoyama, N. H. Tho, S. Z. Md Dawal, and N. A. Masruroh: Measurement **161** (2020) 107860. https://doi.org/10.1016/j.measurement.2020.107860
22 S. Tsuji and T. Kohama: Sens. Actuators, A **291** (2019) 7. https://doi.org/10.1016/j.sna.2019.02.032
23 C. Kantzos, J. Lao, and A. Rollett: Mater. Charact. **158** (2019) 109961. https://doi.org/10.1016/j.matchar.2019.109961

About the Authors

Jyun-Yu Jhang received his B.S. and M.S. degrees from the Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, Taichung, Taiwan, in 2015 and his Ph.D. degree from the Institute of Electrical and Control Engineering in 2021. Currently, he is an assistant professor of the College of Intelligence, National Taichung University of Science and Technology, Taichung, Taiwan. His current research interests include fuzzy logic theory, type-2 neural fuzzy systems, evolutionary computation, machine learning, and computer vision and applications. (jyjhang@nutc.edu.tw)

Cheng-Jian Lin received his B.S. degree in electrical engineering from Ta Tung Institute of Technology, Taipei, Taiwan, R.O.C., in 1986 and his M.S. and Ph.D. degrees in electrical and control engineering from National Chiao Tung University, Taiwan, R.O.C., in 1991 and 1996, respectively. Currently, he is a chair professor of the Computer Science and Information Engineering Department, National Chin-Yi University of Technology, Taichung, Taiwan, R.O.C., and the dean of Intelligence College, National Taichung University of Science and Technology, Taichung, Taiwan, R.O.C. His current research interests are in machine learning, pattern recognition, intelligent control, image processing, intelligent manufacturing, and evolutionary robots.
(cjlin@ncut.edu.tw)