Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 5740362, 10 pages
https://doi.org/10.1155/2020/5740362

Research Article

A Novel Performance Prediction Model for the Machining Process Based on the Interval Type-2 Fuzzy Neural Network

Wenwen Tian[1,2,3], **Fei Zhao**[1,2,3], **Zheng Sun**[1,2,3], **Suiyan Shang**[1,2,3], **Xuesong Mei**[1,2,3] and **Guangde Chen**[4]

[1]*State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong Univeristy, Xi'an 710049, Shaanxi, China*
[2]*Shaanxi Key Laboratory of Intelligent Robots, Xi'an Jiaotong Univeristy, Xi'an 710049, Shaanxi, China*
[3]*School of Mechanical Engineering, Xi'an Jiaotong Univeristy, Xi'an 710049, Shaanxi, China*
[4]*School of Science, Xi'an Jiaotong Univeristy, Xi'an 710049, Shaanxi, China*

Correspondence should be addressed to Fei Zhao; ztzhao@xjtu.edu.cn

Received 4 May 2020; Revised 1 July 2020; Accepted 13 July 2020; Published 24 September 2020

Academic Editor: Kai Zhang

The prediction model is the most important part of the virtual metrology system. Predicting the performance of the machining process has been widely applied in manufacturing, which can reduce costs and improve efficiency compared with the manual operation. In this paper, a novel performance prediction model for the machining process is proposed based on the interval type-2 fuzzy neural network. The interval type-2 fuzzy logic system with a complete rule base, type-reduction, and defuzzified output is simplified by the BMM method to meet the requirements of the prediction. The proposed prediction model is trained using a gradient-based optimization algorithm. To evaluate the performance of the proposed approach, it is applied to wire electrical discharge turning process for predicting material removal rate and surface roughness with a published dataset. The results show that the proposed method is an effective scheme in the studied cases.

1. Introduction

The main objective in the machining process is the improvement of the product quality and productivity at the same time. In order to ensure that product quality meets the requirements, the machining quality inspection of multiple key parts is an essential link in the production process. However, it is almost impossible to perform the total inspection for all parts in real manufacturing processes because such an inspection takes remarkable time and imposes an undeniable delay in the quality control. Although the quality in this method is guaranteed, the continuity of the production is adversely affected. In the context of intelligent manufacturing, finding novel intelligent detection methods is highly required. Reviewing the literature indicates that several schemes have been proposed in this area so far. More specifically, the virtual metrology technology is an effective scheme to perform the quality process monitoring. Investigation and implementation of the virtual metrology have increasingly attracted attentions from both academia and industry. Studies show that this method can effectively reduce the detection time and achieve the total inspection of the product quality. The virtual metrology has been widely applied in diverse high-tech industries, including the semiconductor industry [1], TFT-LCD industry [2], and the solar-cell industry [3]. Moreover, the virtual metrology technology has been applied for processing the CNC machine tool and reasonable results have been achieved accordingly [4–7].

In recent years, different modelling techniques in prediction of machining performances have been proposed for various processes. Because of the causal relationship between machining process and performances, machining process parameters and performances are often used as input and output for modelling, respectively. Due to complexity and uncertainty of the machining processes, soft computing techniques are being preferred to physics-based models for predicting the performance of the machining processes and

optimizing them. A review of application of soft computing techniques in machining performance prediction and optimization has been presented [8]. Machining is the core process in the manufacturing industry, which intends to achieve either a minimum production cost, maximum production rate, or an optimum combination of both to obtain a reasonable production quality through the machining [9]. Meanwhile, the prediction model has attracted interests in the industrial production area. Such interests are especially more pronounced in estimating the machining quality [10], remaining useful life [11, 12], energy consumption [13], and tool wear [14].

Depending on the types of data and the information characteristics of the interested systems, prognosis techniques can be classified into three main categories, including the physics-based, data-driven, and model-based techniques [15]. Many machining monitoring systems based on data-driven artificial intelligence process models have been developed in the past for the optimizing, predicting, or controlling diverse machining processes. Baruah and Chinnam [16] proposed a novel method employing hidden Markov models to conduct both diagnostic and prognostic activities for metal cutting tools. To ensure the stability and quality of machining processes in multistage machining processes (MMPs), a real-time quality monitoring and predicting model was proposed based on error propagation networks [17]. In order to perform the online prediction of the workpiece deformation during the machining, a purely data-driven sparse Bayesian learning-based method was developed [18], which predicts the future deformation of the workpiece through the historical displacement information.

Abnormal machining condition causes quality loss in the finished parts. To this end, Lu et al. [19] proposed a hybrid condition monitoring approach for detecting multiple abnormal conditions in complicated machining process through the deep forest and multiprocess information fusion. Furthermore, Chiu and Lee [20] proposed an intelligent prediction system of milling accuracy and surface quality for CNC machining parameters based on the adaptive neurofuzzy inference systems (ANFIS). They showed that this predicted system can help user to achieve the desired product quality and the machining productivity.

Interval type-2 fuzzy neural networks (IT2FNN) combine the reasoning ability of the interval type-2 fuzzy logic system (IT2FLS) and the self-learning ability of the neural network so that it is a powerful scheme in dealing with the uncertain and nonlinear problems. Although the IT2FNN method has been widely applied in diverse fields, it has been rarely applied in the machining. Type-2 fuzzy logic estimation provides a possibility to show uncertainties in the manufacturing process and can automate the process monitoring, which is of significant importance in maintaining the high quality production. Chen and Vachtsevanos [21] proposed an IT2FNN scheme to perform the multistep-ahead condition prediction in faulty bearings. Its results show a better prediction accuracy compared with the ANFIS method. Since the unforeseeable changes may occur frequently in the dynamic machining environment, Tseng et al. [22] proposed a new methodology to predict a system output by applying a data mining technique and a hybrid type-2 fuzzy system in CNC turning operations. Although the IT2FNN method has more advantages in processing uncertainty, incomplete, and imprecise information compared to its type-1 counterparts, a large number of parameters should be tuned in the IT2FNN method. The structure design of the IT2FNN method is the key step that affects the overall performance. Wu and Mendel [23] recommended some representative starting choices for designing an IT2 fuzzy system. The most challenging questions here are about output processing and type-reduction methods. It is worth noting that many type-reduction methods have been proposed so far [24]. In the present study, it is intended to apply the Begian–Melek–Mendel (BMM) [25] method to compute the crisp output of the IT2FLS. It introduces a closed form for the inference, which eliminates the need to use a type reducer or any complicated defuzzification techniques.

Wire electrical discharge machining (WEDM) is a widely accepted material removal process for components with intricate shapes and profiles. Moreover, wire electrical discharge turning (WEDT) is an emerging area and it can be used for generating cylindrical forms on difficult-to-machine materials by adding a rotary axes on the WEDM. The WEDT process can be modelled through the artificial neural network with feedforward backpropagation algorithm and ANFIS [26]. Sarkheyli et al. [27] proposed a new hybrid technique anchored in the ANFIS and modified genetic algorithm (MGA) to model the correlation between machining parameters and multi-performances, then applied it to the WEDM for predicting material removal rate (MRR) and surface roughness (Ra).

In the present study, it is intended to present a novel prediction model based on the simplified interval type-2 fuzzy neural network for predicting the MRR and Ra in the WEDT, which are important for increasing the productivity and quality of the products. To this end, the IT2FNN method with a complete rule base and the BMM method will be applied to simplify the type-reduction and defuzzified output of the IT2FLS.

The remainder of this article is organized as follows. Structure of the interval type-2 fuzzy neural network and parameters of the learning algorithm are presented in Section 2. Then, case study and discussion are presented in Section 3. Finally, main conclusions are presented in Section 4.

2. Prediction Model Based on the Interval Type-2 Fuzzy Neural Network

2.1. Structure of the Interval Type-2 Fuzzy Neural Network. In this section, it is intended to introduce the structure of the simplified IT2FNN method. With no loss of generality, multiple-input multiple-output (MIMO) fuzzy systems can be decomposed into a series of multiple-input single-output (MISO) fuzzy systems. Figure 1 illustrates the structure of a MISO IT2FNN scheme with six layers. This six-layered network forms an interval type-2 fuzzy logic system whose consequent part is a linear combination of input variables.

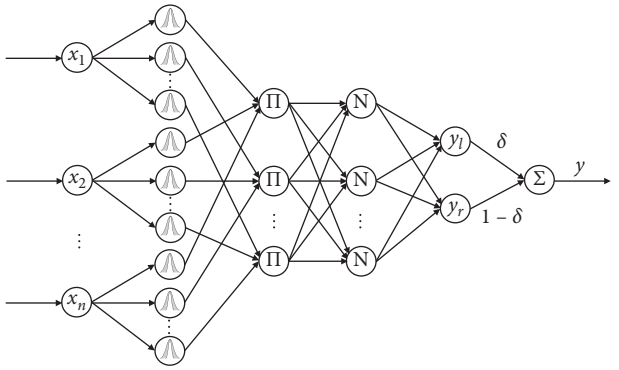

FIGURE 1: Structure of the interval type-2 fuzzy neural network.

In the present study, an interval type-2 fuzzy logic system is considered with a complete rule base, and the consequent parts of the rules are of the Takagi–Sugeno–Kang (TSK) type. In this case, the $j_1, j_2, \ldots, j_{n^{\text{th}}}$ rule can be expressed as follows:

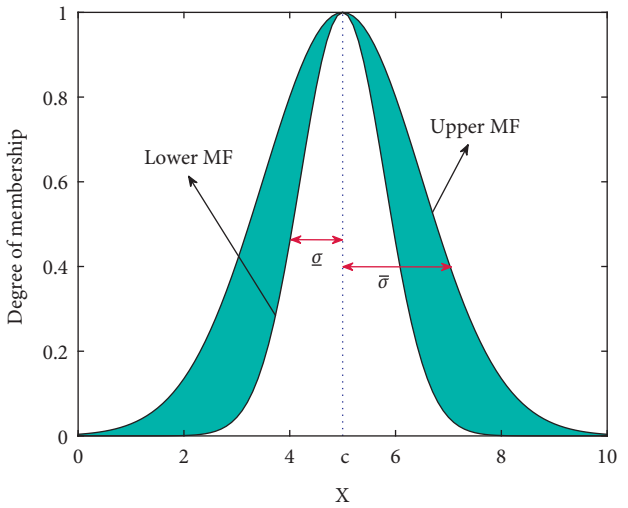

FIGURE 2: Distribution of the interval type-2 membership function with uncertain standard deviation.

$$R^{j_1,j_2,\ldots,j_n}\text{: if } x_1 \text{ is } \tilde{A}_1^{j_1} \text{ and } x_2 \text{ is } \tilde{A}_2^{j_2} \text{ and } \ldots \text{ and } x_n \text{ is } \tilde{A}_n^{j_n}, \quad \text{then } w_{j_1,j_2,\ldots,j_n} \text{ is } b_{j_1,j_2,\ldots,j_n} + \sum_{i=1}^{n} a_{i(j_1,j_2,\ldots,j_n)} x_i, \tag{1}$$

where $x_1, x_2, \ldots, x_n$ are the input variables, $w_{j_1,j_2,\ldots,j_n}$ is the output variable, and $\tilde{A}_i^{j_i}$ is the interval type-2 fuzzy set for the $j_{i^{\text{th}}}$ rule with the i^{th} input. Parameters in the consequent part of rules are $a_{i(j_1,j_2,\ldots,j_n)}$ and $b_{j_1,j_2,\ldots,j_n} (i = 1, 2, \ldots, n, j_i \in \{1, 2, \ldots, m\})$.

The specific expression of each layer is discussed as follows.

(i) Layer 1 (the input layer): this layer contains n neurons. Each node is a crisp input variable. It should be indicated that there are no weights to be adjusted in this layer.

(ii) Layer 2 (the membership function layer): in this article, the Gaussian primary membership function (MF) with uncertain standard deviation and fixed center value is applied in the interval type-2 fuzzy set $\tilde{A}_i^{j_i}$. Subsequently, the footprint of uncertainty (FOU) of this MF can be expressed as a bounded interval in terms of an upper MF $\bar{\mu}_{\tilde{A}_i^{j_i}}$ and a lower MF $\underline{\mu}_{\tilde{A}_i^{j_i}}$, as shown in Figure 2. mn shows the total number of neurons in the MF layer. Each node utilizes an interval type-2 MF to perform the fuzzification operation:

$$\mu_{\tilde{A}_i^{j_i}}(x_i) = \exp\left[-\frac{1}{2}\left(\frac{x_i - c_i^{j_i}}{\sigma_i^{j_i}}\right)^2\right] \equiv N\left(c_i^{j_i}, \sigma_i^{j_i}; x_i\right), \tag{2}$$

where $\mu_{\tilde{A}_i^{j_i}}$ is the $j_{i^{\text{th}}}$ MF with the i^{th} input, while $c_i^{j_i}$ and $\sigma_i^{j_i}$ denote the center and widths of the MF, respectively.

Then, the output of each node can be represented as an interval $[\underline{\mu}_{\tilde{A}_i^{j_i}}, \bar{\mu}_{\tilde{A}_i^{j_i}}]$, with the upper and lower bounds defined as follows:

$$\underline{\mu}_{\tilde{A}_i^{j_i}}(x_i) = \exp\left[-\frac{1}{2}\left(\frac{x_i - c_i^{j_i}}{\underline{\sigma}_i^{j_i}}\right)^2\right] \equiv N\left(c_i^{j_i}, \underline{\sigma}_i^{j_i}; x_i\right),$$

$$\bar{\mu}_{\tilde{A}_i^{j_i}}(x_i) = \exp\left[-\frac{1}{2}\left(\frac{x_i - c_i^{j_i}}{\bar{\sigma}_i^{j_i}}\right)^2\right] \equiv N\left(c_i^{j_i}, \bar{\sigma}_i^{j_i}; x_i\right). \tag{3}$$

(iii) Layer 3 (the rule layer): this layer contains $M = m^n$ neurons. Each node in this layer represents the antecedent part of a fuzzy rule computing the firing strength. This layer performs the product operation to obtain the fuzzy firing strength. Since the MF layer has an interval value of $[\underline{\mu}_{\tilde{A}_i^{j_i}}, \bar{\mu}_{\tilde{A}_i^{j_i}}]$, the firing strength $F^{j_1,j_2,\ldots,j_n}$ for the rule $j_1, j_2, \ldots, j_{n^{\text{th}}}$ is considered as an interval type-1 fuzzy set $[\underline{f}^{j_1,j_2,\ldots,j_n}, \bar{f}^{j_1,j_2,\ldots,j_n}]$, $j_i \in \{1, 2, \ldots, m\}$ so that it can be computed through the following expression:

$$\underline{f}^{j_1,j_2,\ldots,j_n}(x_i) = \prod_{i=1}^{n} \underline{\mu}_{\tilde{A}_i^{j_i}}(x_i), \overline{f}^{j_1,j_2,\ldots,j_n}(x_i) = \prod_{i=1}^{n} \overline{\mu}_{\tilde{A}_i^{j_i}}(x_i). \tag{4}$$

(iv) Layer 4 (the normalization layer): the number of neurons in this layer is also $M = m^n$, which is the same as the number of neurons in layer 3. The outputs of this layer are called normalized firing strengths. The normalization follows the formula below:

$$\widehat{\underline{f}}^{j_1,j_2,\ldots,j_n} = \frac{\underline{f}^{j_1,j_2,\ldots,j_n}}{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\underline{f}^{j_1,j_2,\ldots,j_n}},$$

$$\widehat{\overline{f}}^{j_1,j_2,\ldots,j_n} = \frac{\overline{f}^{j_1,j_2,\ldots,j_n}}{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\overline{f}^{j_1,j_2,\ldots,j_n}}. \tag{5}$$

(v) Layer 5 (the type-reduction layer): the outputs of this layer can be obtained by the firing nodes and the connecting weight vectors, which are defined in the form below:

$$y_l = \sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\widehat{\underline{f}}^{j_1,j_2,\ldots,j_n} w_{j_1,j_2,\ldots,j_n} = \frac{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\underline{f}^{j_1,j_2,\ldots,j_n} w_{j_1,j_2,\ldots,j_n}}{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\underline{f}^{j_1,j_2,\ldots,j_n}},$$

$$y_r = \sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\widehat{\overline{f}}^{j_1,j_2,\ldots,j_n} w_{j_1,j_2,\ldots,j_n} = \frac{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\overline{f}^{j_1,j_2,\ldots,j_n} w_{j_1,j_2,\ldots,j_n}}{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\overline{f}^{j_1,j_2,\ldots,j_n}}, \tag{6}$$

where $w_{j_1,j_2,\ldots,j_n}$ is a linear combination of input variables, which can be expressed as follows:

$$w_{j_1,j_2,\ldots,j_n} = b_{j_1,j_2,\ldots,j_n} + \sum_{i=1}^{n} a_{i(j_1,j_2,\ldots,j_n)} x_i. \tag{7}$$

(vi) Layer 6 (the output layer): each node in this layer computes the output variable through the defuzzification operation. The output value of this layer can be expressed in the form below:

$$y = \delta y_l + (1-\delta) y_r, \tag{8}$$

where the design factor $\delta \in [0,1]$ adjusts the proportion of the upper to the lower bound, depending on the certainty of the system. Therefore, there is no need to apply the Karnik–Mendel (KM) iterative procedure to find the end points [24] and reduce the computational complexity of the type-2 fuzzy logic system for large number of rules.

2.2. Learning Algorithm. In this section, it is intended to use the gradient descent method to derive the parameter learning of the IT2FNN system. For clarification, a single-output system is studied to minimize the objective error. The error function is defined as follows:

$$E(t) = \frac{1}{2}[e(t)]^2 = \frac{1}{2}[y(t) - y_a(t)]^2, \tag{9}$$

where $y(t)$ and $y_a(t)$ are the model output and the actual output, respectively. Applying the gradient descent method, the updated law of parameters can be expressed as follows:

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \Delta\mathbf{W}|_t, \tag{10}$$

where $\Delta\mathbf{W} = -\eta \cdot \partial E(t)/\partial\mathbf{W}$ and $0 \le \delta \le 1$ is the learning rate. Moreover, adjustable parameters are represented as $\mathbf{W} = [\overline{w}, \underline{w}, w_{ab}, \delta]^T$, where $\overline{w}, \underline{w}, w_{ab}$, and δ are the parameters of upper membership functions, lower membership functions, consequent weights, and the proportion factor, respectively. The term with the partial derivative can be expressed as follows:

$$\frac{\partial E(t)}{\partial \mathbf{W}} = \frac{\partial E(t)}{\partial y(t)} \cdot \frac{\partial y(t)}{\partial \mathbf{W}} = [y(t) - y_a(t)] \cdot \frac{\partial y(t)}{\partial \mathbf{W}} = e(t) \cdot \frac{\partial y(t)}{\partial \mathbf{W}}. \tag{11}$$

Therefore, the parameters of the updated law equation can be rewritten as follows:

$$\mathbf{W}(t+1) = \mathbf{W}(t) - \eta \cdot e(t) \cdot \frac{\partial y(t)}{\partial \mathbf{W}}. \tag{12}$$

Then, consequent parameters used in the layer 4 are tuned by the following equations:

$$a_{i(j_1,j_2,\ldots,j_n)}(t+1) = a_{i(j_1,j_2,\ldots,j_n)}(t) - \eta \cdot \frac{\partial E}{\partial a_{i(j_1,j_2,\ldots,j_n)}}, \tag{13}$$

where

$$\begin{aligned}
\frac{\partial E}{\partial a_{i(j_1,j_2,\ldots,j_n)}} &= \frac{\partial E}{\partial y}\frac{\partial y}{\partial w_{j_1,j_2,\ldots,j_n}}\frac{\partial w_{j_1,j_2,\ldots,j_n}}{\partial a_{i(j_1,j_2,\ldots,j_n)}} \\
&= e(t)\cdot\left(\frac{\delta\cdot \underline{f}^{j_1,j_2,\ldots,j_n}}{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\underline{f}^{j_1,j_2,\ldots,j_n}} - \frac{(1-\delta)\cdot \overline{f}^{j_1,j_2,\ldots,j_n}}{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\overline{f}^{j_1,j_2,\ldots,j_n}}\right)\cdot x_i(t),
\end{aligned}$$

$$b_{j_1,j_2,\ldots,j_n}(t+1) = b_{j_1,j_2,\ldots,j_n}(t) - \eta \cdot \frac{\partial E}{\partial b_{j_1,j_2,\ldots,j_n}}, \tag{14}$$

$$\begin{aligned}
\frac{\partial E}{\partial b_{j_1,j_2,\ldots,j_n}} &= \frac{\partial E}{\partial y}\frac{\partial y}{\partial w_{j_1,j_2,\ldots,j_n}}\frac{\partial w_{j_1,j_2,\ldots,j_n}}{\partial b_{j_1,j_2,\ldots,j_n}} \\
&= e(t)\cdot\left(\frac{\delta\cdot \underline{f}^{j_1,j_2,\ldots,j_n}}{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\underline{f}^{j_1,j_2,\ldots,j_n}} - \frac{(1-\delta)\cdot \overline{f}^{j_1,j_2,\ldots,j_n}}{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\overline{f}^{j_1,j_2,\ldots,j_n}}\right).
\end{aligned}$$

Antecedent parameters used in layer 2 are tuned as follows:

$$c_i^{j_i}(t+1) = c_i^{j_i}(t) - \eta \cdot \frac{\partial E}{\partial c_i^{j_i}}, \tag{15}$$

where

$$\begin{aligned}
\frac{\partial E}{\partial c_i^{j_i}} &= \sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\frac{\partial E}{\partial y}\left(\frac{\partial y}{\partial y_l}\frac{\partial y_l}{\partial \underline{f}^{j_1,j_2,\ldots,j_n}}\frac{\partial \underline{f}^{j_1,j_2,\ldots,j_n}}{\partial \underline{\mu}_{\underline{A}_i}^{\sim j_i}}\frac{\partial \underline{\mu}_{\underline{A}_i}^{\sim j_i}}{\partial c_i^{j_i}} + \frac{\partial y}{\partial y_r}\frac{\partial y_r}{\partial \overline{f}^{j_1,j_2,\ldots,j_n}}\frac{\partial \overline{f}^{j_1,j_2,\ldots,j_n}}{\partial \overline{\mu}_{\underline{A}_i}^{\sim j_i}}\frac{\partial \overline{\mu}_{\underline{A}_i}^{\sim j_i}}{\partial c_i^{j_i}}\right), \\
\underline{\sigma}_i^{j_i}(t+1) &= \underline{\sigma}_i^{j_i}(t) - \eta \cdot \frac{\partial E}{\partial \underline{\sigma}_i^{j_i}}, \\
\frac{\partial E}{\partial \underline{\sigma}_i^{j_i}} &= \sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\frac{\partial E}{\partial y}\frac{\partial y}{\partial y_l}\frac{\partial y_l}{\partial \underline{f}^{j_1,j_2,\cdots,j_n}}\frac{\partial \underline{f}^{j_1,j_2,\cdots,j_n}}{\partial \underline{\mu}_{\underline{A}_i}^{\sim j_i}}\frac{\partial \underline{\mu}_{\underline{A}_i}^{\sim j_i}}{\partial \underline{\sigma}_i^{j_i}} \\
\overline{\sigma}_i^{j_i}(t+1) &= \overline{\sigma}_i^{j_i}(t) - \eta \cdot \frac{\partial E}{\partial \overline{\sigma}_i^{j_i}}, \\
\frac{\partial E}{\partial \overline{\sigma}_i^{j_i}} &= \sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\frac{\partial E}{\partial y}\frac{\partial y}{\partial y_r}\frac{\partial y_r}{\partial \overline{f}^{j_1,j_2,\cdots,j_n}}\frac{\partial \overline{f}^{j_1,j_2,\cdots,j_n}}{\partial \overline{\mu}_{\underline{A}_i}^{\sim j_i}}\frac{\partial \overline{\mu}_{\underline{A}_i}^{\sim j_i}}{\partial \overline{\sigma}_i^{j_i}}.
\end{aligned} \tag{16}$$

Table 1: Datasets for designing the experiment [26].

No.	Pulse off-time (μs)	Spark gap (μm)	Servo feed (level)	Rotational speed (rpm)	Flushing pressure (bar)	MRR (mm^3/min)	Ra (μm)
1	30	30	3	30	3.267	1.24	2.396
2	30	50	5	70	3.267	2.21	3.756
3	30	80	8	100	3.267	2.60	4.134
4	34	30	5	100	3.267	1.73	3.172
5	34	50	8	30	3.267	3.78	5.009
6	34	80	3	70	3.267	1.45	2.827
7	42	30	8	70	3.267	1.50	2.899
8	42	50	3	100	3.267	0.95	2.116
9	42	80	5	30	3.267	1.68	3.133
10	30	30	3	30	1.893	1.16	2.311
11	30	50	5	70	1.893	2.20	3.539
12	30	80	8	100	1.893	2.47	4.003
13	34	30	5	100	1.893	1.40	2.694
14	34	50	8	30	1.893	3.72	4.897
15	34	80	3	70	1.893	1.37	2.502
16	42	30	8	70	1.893	1.46	2.895
17	42	50	3	100	1.893	0.82	2.069
18	42	80	5	30	1.893	1.65	3.124
19	30	30	3	30	1.263	1.10	2.264
20	30	50	5	70	1.263	1.48	2.860
21	30	80	8	100	1.263	2.22	3.794
22	34	30	5	100	1.263	1.34	2.506
23	34	50	8	30	1.263	3.20	4.576
24	34	80	3	70	1.263	1.35	2.520
25	42	30	8	70	1.263	1.37	2.588
26	42	50	3	100	1.263	0.78	2.048
27	42	80	5	30	1.263	1.54	2.987
28	30	50	8	70	3.267	2.80	4.281
29	34	80	5	50	1.893	1.44	2.744
30	42	50	3	50	1.263	0.98	2.149

Partial derivatives in the foregoing expressions can be expressed as follows:

$$\frac{\partial E}{\partial y} = y(t) - y_a(t) = e(t),$$

$$\frac{\partial y}{\partial y_l} = \delta,$$

$$\frac{\partial y}{\partial y_r} = 1 - \delta,$$

$$\frac{\partial y_l}{\partial \underline{f}^{j_1,j_2,\ldots,j_n}} = \frac{w_{j_1,j_2,\ldots,j_n} - y_l}{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\underline{f}^{j_1,j_2,\ldots,j_n}},$$

$$\frac{\partial y_r}{\partial \bar{f}^{j_1,j_2,\ldots,j_n}} = \frac{w_{j_1,j_2,\ldots,j_n} - y_r}{\sum_{j_1=1}^{m}\sum_{j_2=1}^{m}\cdots\sum_{j_n=1}^{m}\bar{f}^{j_1,j_2,\ldots,j_n}}$$

$$\frac{\partial \underline{f}^{j_1,j_2,\ldots,j_n}}{\partial \underline{\mu}_{\underline{A}_i}^{j_i}} = \prod_{\substack{k=1\\k\neq i}}^{n} \overline{\mu}_{\underline{A}_k}^{j_k}(x_k),$$

$$\frac{\partial \bar{f}^{j_1,j_2,\ldots,j_n}}{\partial \overline{\mu}_{\bar{A}_i}^{j_i}} = \prod_{\substack{k=1\\k\neq i}}^{n} \overline{\mu}_{\bar{A}_k}^{j_k}(x_k),$$

$$\frac{\partial \underline{\mu}_{\underline{A}_i}^{j_i}}{\partial c_i^{j_i}} = \exp\left[-\frac{1}{2}\left(\frac{x_i - c_i^{j_i}}{\underline{\sigma}_i^{j_i}}\right)^2\right]\cdot\frac{x_i - c_i^{j_i}}{\left(\underline{\sigma}_i^{j_i}\right)^2},$$

$$\frac{\partial \overline{\mu}_{\bar{A}_i}^{j_i}}{\partial c_i^{j_i}} = \exp\left[-\frac{1}{2}\left(\frac{x_i - c_i^{j_i}}{\bar{\sigma}_i^{j_i}}\right)^2\right]\cdot\frac{x_i - c_i^{j_i}}{\left(\bar{\sigma}_i^{j_i}\right)^2},$$

$$\frac{\partial \underline{\mu}_{\underline{A}_i}^{j_i}}{\partial \underline{\sigma}_i^{j_i}} = \exp\left[-\frac{1}{2}\left(\frac{x_i - c_i^{j_i}}{\underline{\sigma}_i^{j_i}}\right)^2\right]\cdot\frac{\left(x_i - c_i^{j_i}\right)^2}{\left(\underline{\sigma}_i^{j_i}\right)^3},$$

$$\frac{\partial \overline{\mu}_{\bar{A}_i}^{j_i}}{\partial \bar{\sigma}_i^{j_i}} = \exp\left[-\frac{1}{2}\left(\frac{x_i - c_i^{j_i}}{\bar{\sigma}_i^{j_i}}\right)^2\right]\cdot\frac{\left(x_i - c_i^{j_i}\right)^2}{\left(\bar{\sigma}_i^{j_i}\right)^3}. \tag{17}$$

It is worth noting that the parameter δ adjusts the lower or upper portions in the final output. Subsequently, the proportion factor can be tuned as follows:

Table 2: Design factors in the IT2FNN architecture.

Parameter	MRR	Ra
Layer 1: Number of input nodes	5	5
Layer 2: Number of membership functions	10	10
Layer 3: Number of rules	32	32
Layer 4: Number of normalization nodes	32	32
Layer 5: Number of type-reduction nodes	2	2
Layer 6: Number of output nodes	1	1
Footprint of uncertainty	Gaussian with uncertain standard deviation	Gaussian with uncertain standard deviation

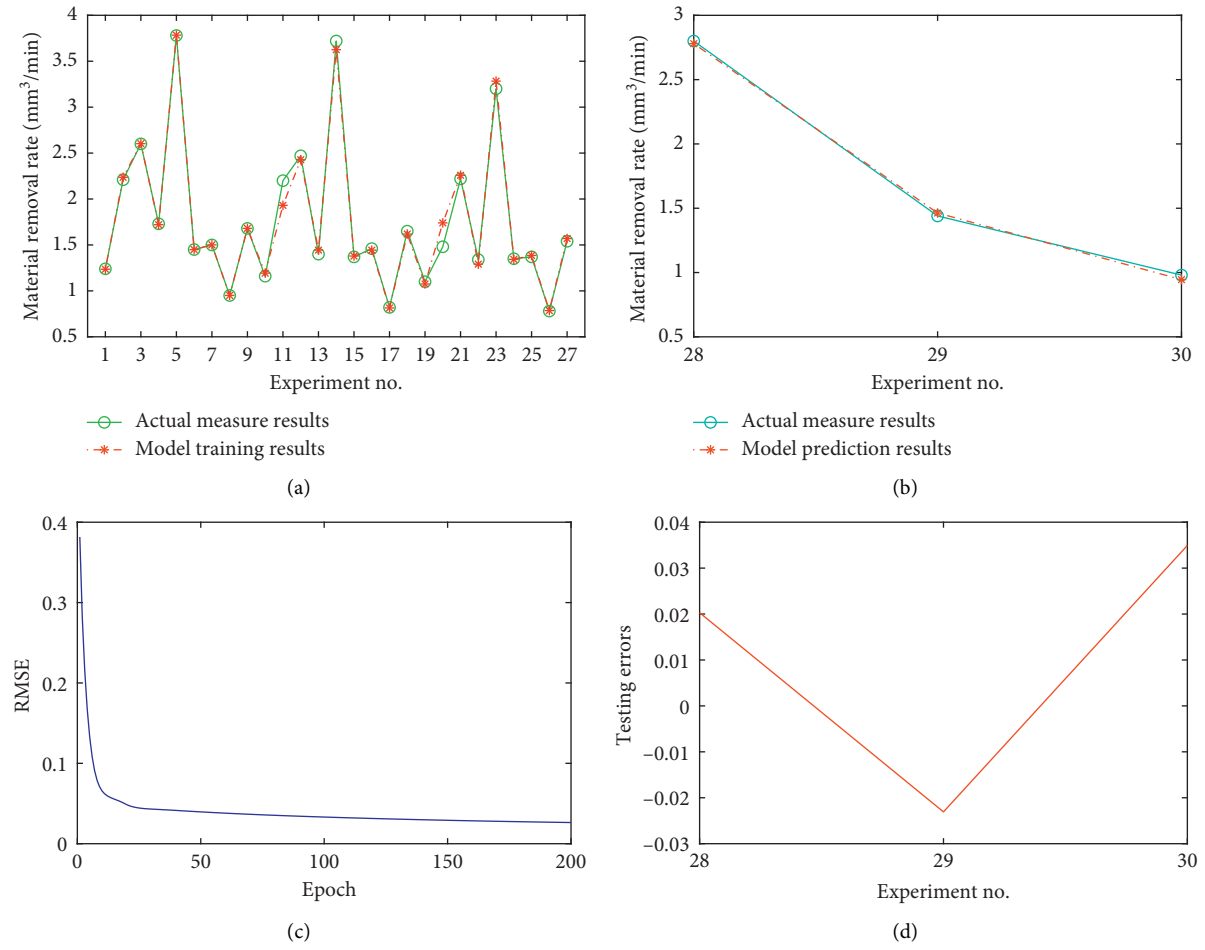

Figure 3: IT2FNN training process and prediction results for MRR.

$$\delta(t+1) = \delta(t) - \eta \cdot \frac{\partial E}{\partial \delta}, \tag{18}$$

where

$$\frac{\partial E}{\partial \delta} = \frac{\partial E}{\partial y}\frac{\partial y}{\partial \delta} = e(t) \cdot \left(\frac{\sum_{j=1}^{M} \underline{f}^j w_j}{\sum_{j=1}^{M} \underline{f}^j} - \frac{\sum_{j=1}^{M} \overline{f}^j w_j}{\sum_{j=1}^{M} \overline{f}^j} \right). \tag{19}$$

An important issue for each learning algorithm is the convergence. The convergence of the gradient descent method depends on the selection of the initial value for the learning rate. The convergence is guaranteed when the following inequality is satisfied [28]:

$$0 < \eta(t) < \frac{2}{\left(\max_t \|\partial y(t)/\partial \mathbf{W}\|\right)^2}. \tag{20}$$

3. Case Study and Discussion

3.1. Designing the Experiment. In this section, it is intended to evaluate the performance of the proposed method in predicting the MRR and Ra in the WEDT. The experimental dataset in this paper is published in [26]. The selected material for conducting the experiments is AISI D3 (DIN X210Cr12) tool steel with the Brinell hardness of about 212–248 HB. The diameter of the workpieces and the designed depth of cut are set to 10 mm and 0.1 mm,

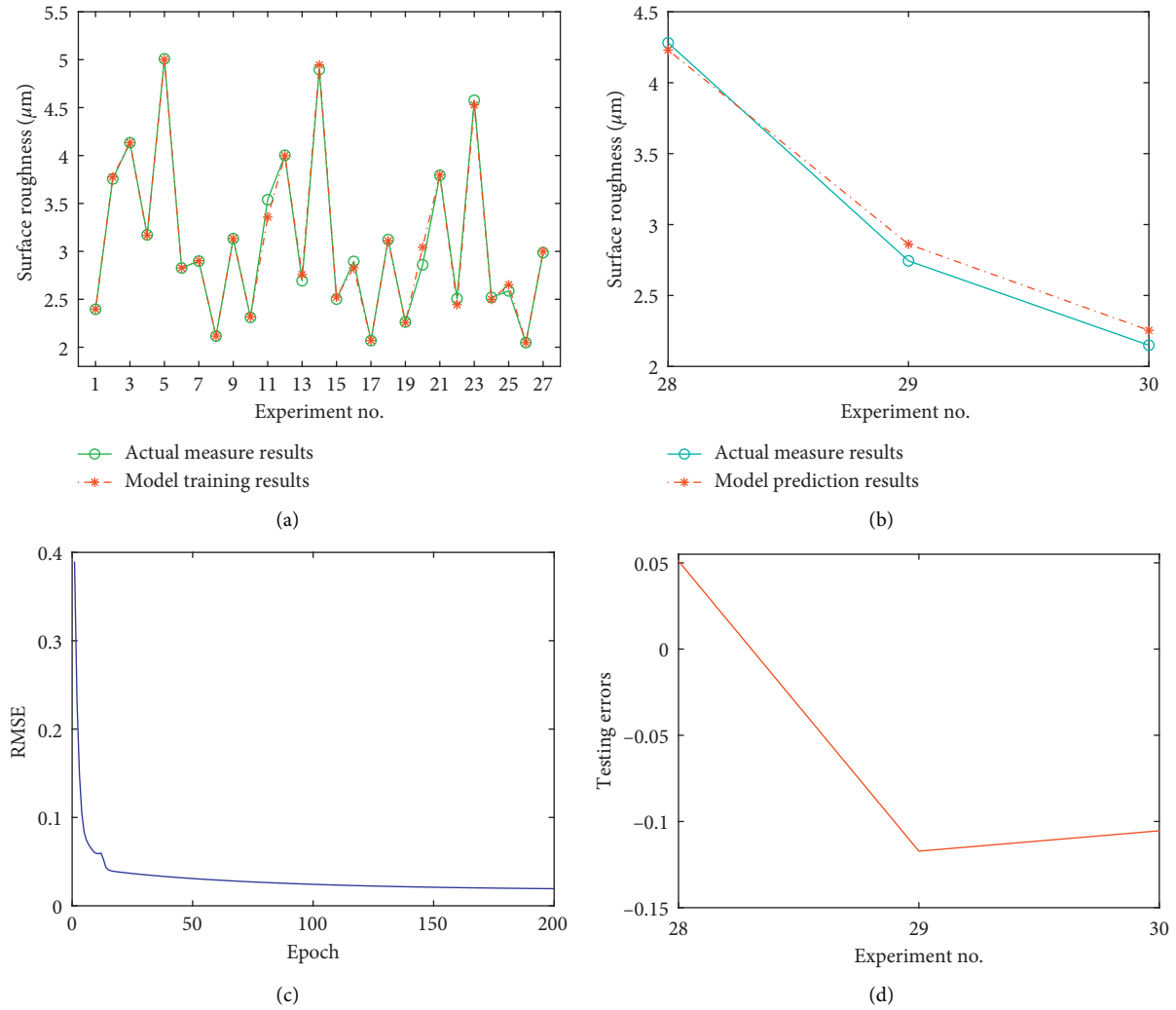

Figure 4: IT2FNN training process and prediction results for Ra.

Table 3: Obtained results from different schemes.

NO.	Experimental results		Prediction result of ANN [26]		Prediction result of ANFIS-GA [27]		Prediction result of ANFIS-MGA [27]		Prediction result of IT2FNN	
	MRR	Ra	MRR	Ra	MRR	Ra	MRR	Ra	MRR	Ra
28	2.80	4.281	2.46	3.734	2.48	3.78	2.644	3.894	2.7797	4.2299
29	1.44	2.744	1.32	2.599	1.36	2.874	1.549	2.979	1.4631	2.8612
30	0.98	2.149	0.92	2.552	1.03	1.791	0.925	2.138	0.9451	2.2544
RMSE			0.2110	0.4011	0.1926	0.3633	0.1144	0.2615	**0.0269**	**0.0950**

respectively. Pulse off-time, spark gap, servo feed, rotational speed, and flushing pressure are selected, as shown in Table 1, to analyse their impact on the MRR and Ra in the WEDT.

3.2. Designing Factors of the Model Architecture. MRR and Ra are separately predicted through the proposed IT2FNN scheme. In this regard, the IT2FNN architecture factors are presented in Table 2. The sample data should be initially normalized prior to the training and testing to remove the effect of different dimension scales on the results. The normalization process is defined as follows:

$$\overline{x}_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}, \tag{21}$$

where x_i and $\overline{x}_i$ denote the sampling data and the corresponding normalized data, respectively. Moreover, $x_{\max}$ and

$x_{\min}$ are the maximum and minimum values of the sampling data, respectively.

3.3. Results and Discussion. In the present study, the first 27 data among 30 sets of experimental data are used for training, while the last 3 ones are considered for testing. It should be indicated that process parameters, which were previously discussed in Section 3.1, are employed as the inputs of the IT2FNN method and then MRR and Ra are obtained separately as the output of the model. In order to evaluate the performance of the proposed method, the root mean square error (RMSE) scheme is applied to measure the prediction accuracy. The RMSE scheme is defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[y(k) - y_a(k)\right]^2}, \tag{22}$$

where N is the prediction length, while $y(k)$ and $y_a(k)$ denote the k^{th} prediction and actual observation, respectively.

Figures 3 and 4 show the proposed IT2FNN training process and prediction results for MRR and Ra, respectively. The sample training results (a) show the correlation between the actual and training outputs, while results of the testing sample (b) show the correlation between the actual and the predicted outputs. Moreover, RMSE variation of the training sample during the iterative learning (c) shows the changing trend of the RMSE during the training, while the corresponding prediction errors of the testing sample (d) show the difference between the predicted and the actual results after the training.

Obtained results show that, after 200 training iterations, the RMSE approached to its asymptotic value and obtained outputs from the model match with the actual measured results during the training so that the prediction accuracy of the test samples increases. Furthermore, it is found that the proposed model has a good generalization ability. Table 3 shows the prediction results of MRR and Ra obtained from different schemes, including the ANN [26], ANFIS-GA [27], and ANFIS-MGA [27]. Compared with other models, it is observed that the proposed IT2FNN scheme has the lowest testing RMSE value. Based on the aforementioned results, it is concluded that the proposed method can achieve better prediction performance for MRR and Ra in the WEDT.

4. Conclusions

Studies show that predicting the performance of the machining process is of great significance in the intelligent manufacturing. In practical applications, the precision and the efficiency of the predicting model are two key factors. Obtained results in the present study demonstrate that the IT2FNN is a powerful scheme in dealing with uncertain and nonlinear problems, which is beneficial to predict the performance of the machining process. The proposed method has been used to predict the material removal rate and surface roughness in the wire electrical discharge turning process, which improves the efficiency and precision compared with the related research results. In the near future, it is intended to apply this method for virtual metrology of the machining precision in other machining process fields.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Key R&D Program of China (Grant no. 2018AAA0101802).

References

[1] F.-T. Cheng and Y.-C. Chiu, "Applying the Automatic Virtual Metrology system to obtain tube-to-tube control in a PECVD tool," *IIE Transactions*, vol. 45, no. 6, pp. 670–681, 2013.

[2] F.-T. Cheng, C.-A. Kao, C.-F. Chen, and W.-H. Tsai, "Tutorial on applying the VM technology for TFT-LCD manufacturing," *IEEE Transactions on Semiconductor Manufacturing*, vol. 28, no. 1, pp. 55–69, 2015.

[3] F.-T. Cheng, C.-F. Chen, Y.-S. Hsieh, H.-H. Huang, and C.-C. Wu, "Intelligent sampling decision scheme based on the AVM system," *International Journal of Production Research*, vol. 53, no. 7, pp. 2073–2088, 2015.

[4] H. Tieng, H. C. Yang, M. H. Hung et al., "A novel virtual metrology scheme for predicting machining precision of machine tools," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 264–269, Karlsruhe, Germany, May 2013.

[5] H.-C. Yang, H. Tieng, and F.-T. Cheng, "Total precision inspection of machine tools with virtual metrology," *Journal of the Chinese Institute of Engineers*, vol. 39, no. 2, pp. 221–235, 2016.

[6] H.-C. Yang, H. Tieng, and F.-T. Cheng, "Automatic virtual metrology for wheel machining automation," *International Journal of Production Research*, vol. 54, no. 21, pp. 6367–6377, 2016.

[7] H. Tieng, T.-H. Tsai, C.-F. Chen, H.-C. Yang, J.-W. Huang, and F.-T. Cheng, "Automatic virtual metrology and deformation fusion scheme for engine-case manufacturing," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 934–941, 2018.

[8] M. Chandrasekaran, M. Muralidhar, C. M. Krishna, and U. S. Dixit, "Application of soft computing techniques in machining performance prediction and optimization: a literature review," *The International Journal of Advanced Manufacturing Technology*, vol. 46, no. 5-8, pp. 445–464, 2010.

[9] R. V. Rao, *Advanced Modeling and Optimization of Manufacturing Processes*, Springer-Verlag, London, UK, 2011.

[10] C. Ahilan, S. Kumanan, N. Sivakumaran et al., "Modeling and prediction of machining quality in CNC turning process using intelligent hybrid decision making tools," *Applied Soft Computing*, vol. 13, no. 3, pp. 1543–1551, 2013.

[11] Y. G. Lei, N. P. Li, S. Gontarz et al., "A model-based method for remaining useful life prediction of machinery," *IEEE Transactions on Reliability*, vol. 65, no. 3, pp. 1314–1326, 2016.

[12] M. Xia, T. Li, T. Shu et al., "A two-stage approach for the remaining useful life prediction of bearings using deep neural networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3703–3711, 2019.

[13] G. Y. Zhao, Z. Y. Liu, Y. He et al., "Energy consumption in machining: classification, prediction, and reduction strategy," *Energy*, vol. 133, pp. 142–157, 2017.

[14] D. Wu, C. Jennings, J. Terpenny et al., "A comparative study on machine learning algorithms for smart manufacturing: tool wear prediction using random forests," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 7, pp. 1–9, 2017.

[15] R. Gao, L. Wang, R. Teti et al., "Cloud-enabled prognosis for manufacturing," *CIRP Annals*, vol. 64, no. 2, pp. 749–772, 2015.

[16] P. Baruah and R. B. Chinnam, "HMMs for diagnostics and prognostics in machining processes," *International Journal of Production Research*, vol. 43, no. 6, pp. 1275–1293, 2005.

[17] P. Y. Jiang, F. Jia, Y. Wang, and M. Zheng, "Real-time quality monitoring and predicting model based on error propagation networks for multistage machining processes," *Journal of Intelligent Manufacturing*, vol. 25, no. 3, pp. 521–538, 2014.

[18] Y. Yuan, H. T. Zhang, Y. Wu et al., "Bayesian learning-based model-predictive vibration control for thin-walled workpiece machining processes," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 509–520, 2017.

[19] Z. Y. Lu, M. Q. Wang, W. Dai, and J. H. Sun, "In-process complex machining condition monitoring based on deep forest and process information fusion," *The International Journal of Advanced Manufacturing Technology*, vol. 104, no. 5-8, pp. 1953–1966, 2019.

[20] H. W. Chiu and C. H. Lee, "Prediction of machining accuracy and surface quality for CNC machine tools using data driven approach," *Advances in Engineering Software*, vol. 114, pp. 246–257, 2017.

[21] C. C. Chen and G. Vachtsevanos, "Bearing condition prediction considering uncertainty: an interval type-2 fuzzy neural network approach," *Robotics and Computer-Integrated Manufacturing*, vol. 28, pp. 509–516, 2012.

[22] T. L. Tseng, F. H. Jiang, and Y. J. Kwon, "Hybrid Type II fuzzy system & data mining approach for surface finish," *Journal of Computational Design and Engineering*, vol. 2, no. 3, pp. 137–147, 2015.

[23] D. Wu and J. M. Mendel, "Recommendations on designing practical interval type-2 fuzzy systems," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 182–193, 2019.

[24] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Springer International Publishing AG, Cham, Switzerland, 2nd edition, 2018.

[25] M. B. Begian, W. W. Melek, and J. M. Mendel, "Stability analysis of type-2 fuzzy systems," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, pp. 947–953, Hong Kong, China, June 2008.

[26] S. Aravind Krishnan and G. L. Samuel, "Multi-objective optimization of material removal rate and surface roughness in wire electrical discharge turning," *The International Journal of Advanced Manufacturing Technology*, vol. 67, no. 9-12, pp. 2021–2032, 2013.

[27] A. Sarkheyli, A. M. Zain, and S. Sharif, "A multi-performance prediction model based on ANFIS and new modified-GA for machining processes," *Journal of Intelligent Manufacturing*, vol. 26, pp. 703–716, 2015.

[28] R. H. Abiyev and O. Kaynak, "Type 2 fuzzy neural structure for identification and control of time-varying plants," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 12, pp. 4147–4159, 2010.