Big data-oriented wheel position and geometry calculation for cutting tool groove manufacturing based on AI algorithms

Guochao Li1,2 · Zhigang Liu1,2 · Jie Lu · Honggen Zhou1,2 · Li Sun1,2

Received: 28 October 2021 / Accepted: 15 January 2022 / Published online: 22 January 2022 © The Author(s), under exclusive licence to Springer-Verlag London Ltd., part of Springer Nature 2022

Abstract

Groove is a key structure of high-performance integral cutting tools. It has to be manufactured by 5-axis grinding machine due to its complex spatial geometry and hard materials. The crucial manufacturing parameters (CMP) are grinding wheel positions and geometries. However, it is a challenging problem to solve the CMP for the designed groove. The traditional trial-and-error or analytical methods have defects such as time-consuming, limited-applying, and low accuracy. In this study, the problem is translated into a multiple output regression model of groove manufacture (MORGM) based on the big data technology and AI algorithms. The inputs are 34 groove geometry features, and the outputs are 5 CMP. Firstly, two groove machining big data sets with different range are established, each of which includes 46,656 records. They are used as data resource for MORGM. Secondly, 7 AI algorithms, including linear regression, k nearest-neighbor regression, decision trees, random forest regression, support vector regression, and ANN algorithms, are discussed to build the model. Then, 28 experiments are carried out to test the big data set and algorithms. Finally, the best MORGM is built by ANN algorithm and the big data set with a larger range. The results show that CMP can be calculated accurately and conveniently by the built MORGM.

Keywords Groove manufacturing · Multiple output regression · AI · Big data

1 Introduction

Groove is one of the most important structures of integral cutting tools, such as end mill, drill, and reamer, because it determines the strength and sharpness of cutting edges, as well as the chip removal ability. Groove has to be machined by 5-axis grinding machine due to its complex spatial surface and tough blank material. With the emergence of diverse groove geometries, how to set the crucial manufacturing parameters (CMP), including grinding wheel positions and geometries, to get the desired groove geometry has been an urgent problem for groove machining and cutting tool manufacture.

During the groove grinding process, CMP and groove geometry follow the enveloping theory, namely, the common normal at the contact point between the wheel and the groove surface must intersect with the tool axis [1]. Accordingly, mathematical relationships (i.e., enveloping equation) between CMP and groove could be deduced. However, the enveloping equation was a transcendental equation with trigonometric functions, which was hard to be solved. To solve the problem, two methods were mainly used: divide-and-conquer method and objective function [2].

The divide-and-conquer method was an iterative approach. By finding the different influence rates of the wheel position on the machined groove parameters, the wheel position was searched step by step. Based on the directly proportional relationship between the wheel position and the two groove parameters rake angle and core radius, Kim et al. [3], Rababah and Chen [4] and Ren et al. [5] roughly searched and slightly adjusted the wheel position by the rake angle and the core radius respectively and successively. Furthermore, the calculation precision was improved by Nguyen and Ko [6]. Differently, Xiao et al. [7] and Jia et al. [8] changed the calculation order: The core radius was considered first, followed by the rake angle. Chen et al. [9]...
take the calculation procedure: core radius, groove width, and then rake angle.

The objective function method was an optimization approach to calculate the wheel position until the function was approximately solved. Wang et al. [10] built an objective function by the error of rake angle, core radius, and groove width between the desired and the machined groove. Then, the 1A1-type wheel position was obtained by searching the minimum value of the function. Ren et al. [11] created a system of nonlinear equations to calculate the wheel position, and the wheel position can be calculated accurately to ensure the accuracy of the groove parameters: rake angle, core radius, and flute width. Karpuschewski et al. [12] and Li et al. [13] built the objective function with machine errors and successfully searched the wheel position using intelligent algorithms, such as PSO and NPSO. Habibi and Chen [14] calculated the established constraint equations by using Newton iteration algorithm to improve the speed and reliability of the calculation.

Besides, Fang et al. [15] discussed an approach to calculating the wheel location and orientation for CNC flute-grinding by re-formulating the wheel’s determination model and analyzing the geometrical constraints for interference, over-cut, and undercut in a unified framework. The model was integrated with the evolution algorithm and local search operator to optimize the wheel’s location and orientation. Li et al. [2] built a general model that is established to calculate the wheel path for complex groove machining based on a mathematical optimization model, which have three constraints and one objective. Considering that the worn wheel would result in groove grinding error without a proper compensation, Liu et al. [16] proposed a compensation algorithm of worn wheel by analyzing the boundary contact condition which is influenced by wheel wear. Uhlmann et al. [17] presented a method for designing application-oriented grinding wheels to improve the productivity and the quality of grinding processes so that grinding wheels with different layers over its width were developed to compensate the varying and complex contact conditions. Wasif et al. [18] proposed an approach to determine the non-standard grinding wheel that can be economically produced or dressed to accurately grind the end-mill cutters using the five-axis CNC grinding process.

In conclusion, the method discussed above focused on solving complex equation set, which was time-consuming and limited-applying. Essentially, the groove machining problem could be translated into a multiple output regression problem. The input parameters were groove geometry features and the output targets were CMP. Considering that artificial intelligence (AI) algorithms were good at building regression models, a multiple output regression model of groove manufacture (MORGM) is opted based on AI algorithms and big data of groove machining processes so that the CMP could be readily calculated to manufacture the desired groove.

2 Big data generation and feature extraction

2.1 Modeling of groove machining process

The cutting tool groove was generated by the principle of “envelope forming.” Namely, the groove was machined by the profile of the grinding wheel, which moved helically around the tool axis. Thus, the groove was determined by the grinding wheel geometry, as well as its position that relative to the tool.

The grinding wheel process was modeled by space analytical geometric theory, as presented in Fig. 1. Two Cartesian coordinate systems were established. One was the grinding wheel coordinate, denoted as \( \mathbb{O}_wX_wY_wZ_w \), whose \( Z_w \)-axis was coincided with the wheel axis, and
XW-YW plane was coincided with the end face of the wheel. The other was the tool coordinate, denoted as OT-XT YT ZT, whose ZT-axis was coincided with the tool axis, and XW-YW plane was coincided with the end face of the tool. During the machining process, the OT-XT YT ZT was motionless, while the OW-XWYWZW was moved helically around the tool axis together with the wheel.

The wheel position was defined by 3 parameters: the distance between XW and XT (i.e., Δy), the distance between ZW and ZT (i.e., Δx), and the angle that rotated from ZT-axis to ZW-axis around XT-axis (i.e., Δαx).

The 1A1 type grinding wheel geometry was expressed by 3 variable parameters: the thickness BW, the taper angle θW, and the fillet radius gr.

According to the method introduced in reference [13], the groove geometry could be calculated with the known wheel geometry and its position. And the groove was predicted by discrete points that belong to a series of annulus, which had the same width and different radius. Then, the groove profile as well as its features could be defined by these points.

The groove in the reference was defined by only 3 features, namely, rake angle, core radius, and groove width. Obviously, three features were too few for AI algorithms to establish the MORGM. Therefore, much more features should be defined. As presented in Fig. 2, 34 features were used in the study, including the commonly used two features core radius (rc) and groove width (Φ), and the Euclidean distance between points that located on the groove profile. In this study, 30 annuluses were used so that 61 points were deduced. The first Euclidean distance d1 was the distance between p1 and prc; the second d2 was the distance between p61 and prc. Then, other Euclidean distances were between the two points that located in the same annulus, which were denoted as d3, d4, d5 … d32.

2.2 Big data generation

In order to establish MORGM by AI algorithms, two big datasets were established. And for every set, 6 parameters were selected, namely, Δαx, Δx, Δy, θW, gr, and BW. The wheel diameter Dw was excluded because its influence on the groove could be offset by Δx.

Then, each parameter was assigned to 6 values. Differently, the first set (denoted as SETA) had a small variable space, and the second set (denoted as SETB) had a big one, as listed in Table 1. Thus, every set was consisted of 46,656 records, and each record included 6 output labels and 34 input features. The structure and some data for SETB is listed in Table 2. The corresponding groove geometries for SETA and SETB are presented in Figs. 3 and 4.

2.3 Big data analysis and feature extraction

As features would have great impact on the accuracy and efficiency for AI algorithms, the big datasets were analyzed to select efficient features. Dispersion coefficient was a relative statistic to measure the degree of dispersion of data, which was mainly used to compare the degree of dispersion of different sample data. The large dispersion coefficient indicated that the degree of dispersion of data was also large. If the units were different, standard deviation could not be used to compare the degree of dispersion, but the ratio of standard deviation to the mean (relative value) should be used to compare the dispersion. Therefore, coefficient of dispersion (VS) for the 34 features is analyzed

\[ V_S = \frac{\sigma}{\mu} \] (1)

The result is presented in Fig. 5. The values of VS for SETA were between 0.145 and 0.185. The values of VS for SETB were between 0.235 and 0.332. The second and the
third features had bigger values than others, but the difference were indistinctively. Vs for SETA is always lower than SETB for all 34 features, because SETA had a small variable space while SETB had a big one, as listed in Table 1. Considering that Vs is a relative statistic to evaluate the degree of data dispersion, and that the large dispersion coefficient indicates that the degree of data dispersion is also large, Vs for SETA is always lower than SETB is an acceptable result. Thus, all the 34 features were selected.

2.4 Data preprocessing

Considering that the 34 features had different units, the data in the set should be preprocessed. There were usually two ways: one was the standardization method and the other was the normalization method. The standardization method was based on the mean and standard deviation of the raw data:

\[
x^* = \frac{x - \mu}{\sigma}
\]

where \(\mu\) is the mean value and \(\sigma\) is the standard deviation. After processed, all the data were in accordance with standard normal distribution.

Normalization was a way of simplifying the calculation by transforming a dimensionless expression into a dimensionless expression:

\[
x^* = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}
\]

where \(x_{\text{max}}\) and \(x_{\text{min}}\) are the maximum and minimum values of the data. After processed, all the data were mapped to the range \([0,1]\).

3 AI methods for MORGM

The problem discussed in the study was essentially a multiple output regression problem. So far, the main algorithms that support multiple output regressions are as follows: linear regression (LR), \(k\) nearest-neighbor regression (\(k\)NN), decision trees (DTs), random forest regression (RFR), and ANN. However, some algorithms had to be modified to support multiple output, such as SVR. In this paper, the wrapper multiple output regression algorithm and the chained multiple output regression algorithm were used to reconstruct the SVR algorithm respectively. Therefore, in order to find the

| Labels | Variable (unit) | Variable space of SETA | Variables of SETA | Variable space of SETB | Variables of SETB |
|--------|----------------|------------------------|-------------------|------------------------|-------------------|
| Lab1   | \(\Delta \alpha_x\) (degree) | [35, 38] | (35, 35.6, 36.2, 36.8, 37.4, 38) | [33, 40] | (33, 34.4, 35.8, 37.2, 38.6, 40) |
| Lab2   | \(\Delta x\) (mm) | [76, 78] | (76, 76.4, 76.8, 77.2, 77.6, 78) | [75, 79] | (75, 75.8, 76.6, 77.4, 78.2, 79) |
| Lab3   | \(\Delta y\) (mm) | [-16, -14] | (-16, -15.6, -15.2, -14.8, -14.4, -14) | [-17, -13] | (-17, -16.2, -15.4, -14.6, -13.8, -13) |
| Lab4   | \(\theta_w\) (degree) | [70, 80] | (70, 72, 74, 76, 78, 80) | [65, 85] | (65, 69, 73, 77, 81, 85) |
| Lab5   | \(gr\) (mm) | [0, 4] | (0, 0.8, 1.6, 2.4, 3.2, 4) | [0.8] | (0, 1.6, 3.2, 4.8, 6.4, 8) |
| Lab6   | \(B_w\) (mm) | [10, 30] | (10, 14, 18, 22, 26, 30) | [10, 30] | (10, 14, 18, 22, 26, 30) |

Table 1 Variables to produce big data SETA and SETB

| No | \(\Delta \alpha_x\) | \(\Delta x\) | \(\Delta y\) | \(\theta_w\) | \(gr\) | \(B_w\) | Ft1 | Ft2 | Ft3 | … | Ft34 |
|----|----------------|---|---|---|---|---|----|----|----|----|---|---|
| 1  | 0.628 | 75.8 | -13.0 | 1.484 | 1.6 | 26.0 | 219.357 | 2.433 | 219.357 | … | 3.6989 |
| 2  | 0.576 | 75.0 | -14.6 | 1.484 | 1.6 | 10.0 | 146.440 | 1.956 | 146.440 | … | 3.5129 |
| 3  | 0.600 | 79.0 | -14.6 | 1.135 | 0.0 | 10.0 | 66.187 | 5.894 | 66.187 | … | 2.1440 |
| 46,656 | 0.663 | 76.8 | -14.8 | 1.327 | 2.4 | 22.0 | 130.339 | 3.959 | 48.992 | … | 4.0056 |

The produced big data structure (SETB)

The units of \(\Delta \alpha_x\) and \(\theta_w\) in this table were rad
best method, seven AI methods was respectively employed to build the MORGM.

3.1 Linear regression (LR)

Linear regression fitted a linear model with coefficients
$$w = (w_1, \ldots, w_p)$$
to minimize the residual sum of squares between the observed targets in the dataset, and the targets predicted by the linear approximation. It tried to build a function that predicts by a linear combination of properties, namely:

$$y = \omega^T x + b$$  \hspace{1cm} (4)

where, in this study, $y = (\text{Lab1}; \text{Lab2}; \text{Lab3}; \text{Lab4}; \text{Lab5}; \text{Lab6})$, $\omega = (\omega_1; \omega_2; \ldots; \omega_{34})$, and $x = (Ft1; Ft2; Ft3; \ldots; Ft34)$.

3.2 k-Nearest neighbor (kNN)

$k$-Nearest neighbor was a kind of supervised learning methods. It could be used for solving classification and regression problems. The target was predicted based on the $k$ nearest neighbors of each query point. In this study, 5 neighbors (namely, $k = 5$), and uniform weights were used: That is, each point in the local neighborhood contributes uniformly to the classification of a query point.

3.3 Random forest regressor (RFR)

A random forest was a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. In this study, the number of trees in the forest was 10. The maximum depth of the tree was none, which mean that nodes were expanded until all leaves were pure or until all leaves contained less than 2 samples.

3.4 Decision trees (DTs)

Decision trees (DTs) were a non-parametric supervised learning method used for regression. In order to support multi-output problems, decision trees were changed by two ways: (1) store 6 output values in leaves and (2) use

---

Fig. 3 Some typical groove geometries corresponding to SETA (initial parameters: $\Delta \alpha_x = 35$, $\Delta x = 76$, $\Delta y = -16$, $\theta_w = 70$, $gr = 0$, and $B_w = 10$)
Fig. 4 Some typical groove geometries corresponding to SETB (initial parameters: $\Delta \alpha_x = 35$, $\Delta x = 76$, $\Delta y = -16$, $\theta_W = 70$, $gr = 0$, and $B_W = 10$)

Fig. 5 Coefficient of dispersion ($V_s$) for the 34 features
splitting criteria that compute the average reduction across all 7 outputs.

3.5 **SVR-multi-output-regressor (SVR-MOR)**

Support vector regression (SVR) was a typical single output model. For this study, it was modified by two methods to meet the multiple output requirement. One was multi-target regression strategy, and the other is chain strategy.

Multi-target regression strategy consists of fitting one regressor per target, which was a commonly used strategy for extending regressors that do not natively support multi-target regression.

3.6 **SVR-regressor-chain (SVR-RC)**

The chain strategy was a multi-label model that arranges regressions into a chain. Each model made a prediction in the order specified by the chain using all of the available features provided to the model plus the predictions of models that were earlier in the chain.

3.7 **Artificial neural network (ANN)**

The fully connected artificial neural network was established. It included 7 layers, and each layer consists of 384, 192, 96, 48, 24, 12, and 6 nodes. “Sigmoid” was selected as the activation function, and “RMSProp” was employed.

![Fig. 6 Structure of artificial neural network](image_url)

**Table 3 Experimental design for MORGM**

| Groups | No | Datasets and methods | Mark | Groups | No | Datasets and methods | Mark |
|--------|----|----------------------|------|--------|----|----------------------|------|
| Group1 | 1  | A1 B1 C1             | A1B1C1 | Group3 | 15 | A2 B1 C1             | A2B1C1 |
|        | 2  | A1 B1 C2             | A1B1C2 |        |    | A2 B1 C2             | A2B1C2 |
|        | 3  | A1 B1 C3             | A1B1C3 |        |    | A2 B1 C3             | A2B1C3 |
|        | 4  | A1 B1 C4             | A1B1C4 |        |    | A2 B1 C4             | A2B1C4 |
|        | 5  | A1 B1 C5             | A1B1C5 |        |    | A2 B1 C5             | A2B1C5 |
|        | 6  | A1 B1 C6             | A1B1C6 |        |    | A2 B1 C6             | A2B1C6 |
|        | 7  | A1 B1 C7             | A1B1C7 |        |    | A2 B1 C7             | A2B1C7 |
| Group2 | 8  | A1 B2 C1             | A1B2C1 | Group4 | 22 | A2 B2 C1             | A2B2C1 |
|        | 9  | A1 B2 C2             | A1B2C2 |        |    | A2 B2 C2             | A2B2C2 |
|        | 10 | A1 B2 C3             | A1B2C3 |        |    | A2 B2 C3             | A2B2C3 |
|        | 11 | A1 B2 C4             | A1B2C4 |        |    | A2 B2 C4             | A2B2C4 |
|        | 12 | A1 B2 C5             | A1B2C5 |        |    | A2 B2 C5             | A2B2C5 |
|        | 13 | A1 B2 C6             | A1B2C6 |        |    | A2 B2 C6             | A2B2C6 |
|        | 14 | A1 B2 C7             | A1B2C7 |        |    | A2 B2 C7             | A2B2C7 |

Datasets (A1, SETA; A2, SETB); data preprocessing (B1, normalization; B2, standardization); methods (C1, LR; C2, kNN; C3, RFR; C4, DT; C5, SVR-MOR; C6, SVR-RC; C7, ANN)
as the optimizer algorithm. The expression of ‘Sigmoid’ is \( f(x) = \frac{1}{1 + e^{-x}} \). It is one of the most widely used activation functions. Its main advantage is that its output is a range of probabilities from 0 to 1, and the sum of all probabilities is 1. Thus, it is suitable for multi-class problems. Considering that our study is a multiple output regression problem, which is similar to the multi-class problem, “Sigmoid” is employed as activation function. RMSProp, root mean square propagation, is an optimization algorithm/method designed for artificial neural network (ANN) training. RMSProp lies in the realm of adaptive learning rate methods, which have been growing in popularity in recent years because it is the extension of stochastic gradient descent (SGD) algorithm, momentum method, and the foundation of Adam algorithm. RMSProp is able to continually reduce the loss throughout the training process until the local minimum is reached. One of the applications of RMSProp is the stochastic technology for mini-batch gradient descent. Therefore, “RMSProp” is employed as optimizer algorithm in the study.

### Table 4 Functions corresponding to approaches

| Approaches | Functions |
|------------|-----------|
| LR         | sklearn.linear_model.LinearRegression() |
| KNN        | sklearn.neighbors.KNeighborsRegressor() |
| RFR        | sklearn.ensemble.RandomForestRegressor() |
| DT         | sklearn.tree.DecisionTreeRegressor() |
| SVR-MOR    | sklearn.svm.LinearSVR() |
|            | sklearn.multioutput.MultiOutputRegressor() |
| SVR-RC     | sklearn.svm.LinearSVR() |
|            | sklearn.multioutput.RegressorChain() |
| ANN        | keras.models.Sequential() |

### Table 5 Values of MSE for the 28 experiments

| No | Mark    | Lab1       | Lab2       | Lab3       | Lab4       | Lab5       | Lab6       | MSE (Average, except Lab6) |
|----|---------|------------|------------|------------|------------|------------|------------|--------------------------|
| 1  | A1B1C1  | 0.08955    | 0.00934    | 0.10263    | 0.00309    | 0.00309    | 0.07181    | 0.043449                 |
| 2  | A1B1C2  | 0.00495    | 0.00114    | 0.01405    | 0.00021    | 0.00005    | 0.11067    | 0.005087                 |
| 3  | A1B1C3  | 0.00361    | 0.00146    | 0.01509    | 0.00051    | 0.00005    | 0.10640    | 0.005169                 |
| 4  | A1B1C4  | 0.01045    | 0.00325    | 0.02978    | 0.00184    | 0.00026    | 0.17556    | 0.011331                 |
| 5  | A1B1C5  | 0.09308    | 0.00941    | 0.10453    | 0.00312    | 0.00321    | 0.07294    | 0.010640                 |
| 6  | A1B1C6  | 0.09291    | 0.00956    | 0.10387    | 0.00312    | 0.00321    | 0.07294    | 0.011331                 |
| 7  | A1B1C7  | 0.00222    | 0.00050    | 0.00354    | 0.00020    | 0.00012    | 0.11733    | 0.000163                 |
| 8  | A1B2C1  | 0.76760    | 0.08007    | 0.89766    | 0.02653    | 0.02652    | 0.61551    | 0.372424                 |
| 9  | A1B2C2  | 0.08072    | 0.01867    | 0.21830    | 0.00303    | 0.00052    | 0.11721    | 0.008179                 |
| 10 | A1B2C3  | 0.03106    | 0.01240    | 0.12974    | 0.00449    | 0.00045    | 0.11243    | 0.004421                 |
| 11 | A1B2C4  | 0.08736    | 0.02746    | 0.25771    | 0.01561    | 0.00185    | 1.48904    | 0.097034                 |
| 12 | A1B2C5  | 0.79103    | 0.08261    | 0.89320    | 0.02691    | 0.02921    | 0.62529    | 0.382036                 |
| 13 | A1B2C6  | 0.79098    | 0.08537    | 0.94591    | 0.02859    | 0.02759    | 0.62118    | 0.381530                 |
| 14 | A1B2C7  | 0.01143    | 0.00212    | 0.02033    | 0.00063    | 0.00110    | 0.57225    | 0.008746                 |
| 15 | A2B1C1  | 0.10030    | 0.00836    | 0.09507    | 0.00667    | 0.00667    | 0.08056    | 0.047667                 |
| 16 | A2B1C2  | 0.01504    | 0.01144    | 0.01892    | 0.00032    | 0.00048    | 0.11721    | 0.008992                 |
| 17 | A2B1C3  | 0.00942    | 0.00147    | 0.01615    | 0.00054    | 0.00025    | 0.11243    | 0.006892                 |
| 18 | A2B1C4  | 0.02191    | 0.00349    | 0.03330    | 0.00213    | 0.00087    | 0.18502    | 0.015207                 |
| 19 | A2B1C5  | 0.10102    | 0.00860    | 0.09675    | 0.00701    | 0.00735    | 0.08442    | 0.049282                 |
| 20 | A2B1C6  | 0.10089    | 0.00852    | 0.09675    | 0.00797    | 0.00726    | 0.08435    | 0.049397                 |
| 21 | A2B1C7  | 0.01960    | 0.00171    | 0.01894    | 0.00068    | 0.00030    | 0.06914    | 0.010233                 |
| 22 | A2B2C1  | 0.94212    | 0.07791    | 0.88628    | 0.06222    | 0.05701    | 0.69049    | 0.429235                 |
| 23 | A2B2C2  | 0.15130    | 0.01503    | 0.18222    | 0.03318    | 0.00431    | 0.99143    | 0.089216                 |
| 24 | A2B2C3  | 0.08363    | 0.01353    | 0.14854    | 0.00515    | 0.00216    | 0.96575    | 0.062712                 |
| 25 | A2B2C4  | 0.18351    | 0.03239    | 0.31464    | 0.01877    | 0.00755    | 1.60833    | 0.137554                 |
| 26 | A2B2C5  | 0.94893    | 0.07950    | 0.89652    | 0.06381    | 0.06213    | 0.72106    | 0.440222                 |
| 27 | A2B2C6  | 0.95102    | 0.08097    | 0.91379    | 0.06957    | 0.06259    | 0.73013    | 0.448615                 |
| 28 | A2B2C7  | 0.05924    | 0.00796    | 0.07057    | 0.00259    | 0.00262    | 0.57382    | 0.035098                 |
The structure of the network was presented in Fig. 6.

4 Experimental design

Generally, training datasets, data preprocessing methods, and algorithms were main factors for regression models. In order to get the best MORGM, 28 experiments were designed, as listed in Table 3. Variables included 2 different datasets, 2 data preprocessing methods, and 7 modeling algorithms. To increase readability, all experiments were marked with letters and numbers and divided into 4 groups on average.

Based on the datasets introduced in Sect. 2.2, all the 46,656 records were divided into two sets. One was the training set including 40,000 records, and the other was the testing sets including 6656 records.

All the algorithms were implemented by Python 3.9.1 and Keras 2.4.0 with Tensorflow2.4.1 as their backend, and running on one Intel(R) UHD Graphics GPU. The server operating system was Windows 10. The functions corresponding to the 7 approaches are listed in Table 4.

5 Results and discussion

5.1 Model performance metrics

Generally, mean square error (MSE), mean absolute error (MAE), and R square ($R^2$) were used to evaluate regression models. Considering that they had the similar effects, MSE

![Fig. 7 Values of MSE for the 28 experiments. a MSE for Group 1, b MSE for Group 2, c MSE for Group 3, d MSE for Group 4](image-url)
was selected, and the best model would have the minimum MSE, because the smaller MSE means the better accurate. Therefore, the next work was to find the minimum MSE from the 28 experiments.

The result of MSEs for the 28 experiments is listed in Table 5. The 6 labels were analyzed respectively, as shown in Fig. 7. Obviously, Lab6 had the largest MSE nearly for all the 28 experiments. It was because that there were only two valid samples for Lab6 (as shown in Figs. 3 and 4), which was insufficient to establish a regression model. Besides, according to Fig. 4, Lab6 (i.e., the thickness of the wheel) had little impact on the groove. Therefore, Lab6 was removed from label groups, and the average of the prior 5 labels were used for evaluation. For convenience, the symbol “SME” means the average of the prior 5 labels in the following section.

5.2 Datasets analysis

As dataset was a foundation for a regression model, SETA and SETB were used to evaluate the influence of dataset on MORGM.

The results were presented in Fig. 8. It indicated that MSE changed with different datasets and SETA (signed as A1) performed better than SETB (signed as A2). Besides,
the change rate corresponding to the 7 methods was also different. It changed little for C1, C5, and C6 methods (under 20%), while it reached more than 300% for C7. Thus, datasets would have little impact on the MORGM by using C1, C5, and C6 methods and have great impact by using C7 method.

5.3 Data preprocessing analysis

The influences of standardization and normalization methods on MSE were analyzed, as shown in Fig. 9. The results showed that B1 was much better than B2 for all the 7 methods, as well as for all the 2 datasets. Therefore, B1 (normalization) method was selected to build MORGM.

5.4 AI methods analysis

The average MSE of the prior 5 labels were used to compare the performances of the 7 methods, as shown in Fig. 10. It indicated that C1, C5, and C6 had poor performances, while C2, C3, C4, and C7 had good accurate, and C7 was the best among them. Therefore, C7 (ANN) method was selected to build MORGM.

5.5 Modeling and validating of MORGM

As has been discussed above, MORGM was built by the ANN methods (C7), and the normalization method (B1) was selected to preprocess the datasets. The MORGM have high accuracy both for SETA and SETB (see Fig. 10). The biggest MSE of the testing set (including 6656 records) that produced by A1B1C7 was picked out, as listed in Table 6. It showed that the biggest MSE value was 0.0256, which was so little for a multi-output regression model. Furthermore, two grooves that machined respectively with true and predicted parameters were plotted in Fig. 11. It showed that the groove produced by the predicted parameters was in good consistence with the true parameters.

Table 6 Results corresponding to the biggest MSE produced by A1B1C7

|       | Lab1 | Lab2 | Lab3  | Lab4  | Lab5 |
|-------|------|------|-------|-------|------|
| True  | 0.610865 | 76   | -16   | 1.221731 | 0.8  |
| Predicted value | 0.612814 | 75.97774 | -16.0003 | 1.222485 | 0.825596 |
| MSE   | 0.0256 |      |       |       |      |

Fig. 11 Comparison of grooves produced by true and predicted parameters
6 Conclusion

In order to meet the increasing manufacturing requirements of diverse cutting tool grooves, MORGM was established based on AI algorithms and big data technology, so that the CMP could be readily calculated to manufacture the desired groove. The main contributions can be concluded as follows:

1. Different from previous studies, the groove geometry was determined by 34 features, which could recognize the groove more exactly and be more appropriate for AI algorithms.
2. Big data of groove machining process was built, which was consisted of 46,656 records and each record concluding 6 output labels and 34 input features. It would be a valuable data resource for groove machining.
3. MORGM was built by the ANN algorithm and the normalization data preprocess method. And the wheel position and geometry could be readily calculated by the desired groove. The results showed that the built MORGM had high accuracy and will have wide application prospect in groove manufacture.

Author contribution Guochao Li: conceptualization, methodology, writing-original draft preparation and editing. Zhigang Liu: software, data curation. Jie Lu: investigation. Honggen Zhou: supervision, software, validation. Li Sun: writing-reviewing.

Funding This work was supported by the National Natural Science Foundation of China (No. 51605207) and the Natural Science Foundation of Jiangsu Province of China (No. BK20160563).

Declarations

Ethics approval Not applicable.
Consent to participate Not applicable.
Consent for publication Not applicable.
Conflict of interest The authors declare no competing interests.

References

1. Sheth DS, Malkin S (1990) CAD/CAM for geometry and process analysis of helical groove machining. CIRP Ann 39:129–132
2. Li G, Dai L, Liu J, Zhou H, Tian G, Li L (2020) An approach to calculate grinding wheel path for complex end mill groove grinding based on an optimization algorithm. J Manuf Process 53:99–109
3. Kim JH, Park JW, Ko TJ (2008) End mill design and machining via cutting simulation. Comput Aided Des 40:324–333
4. Rababah MM, Chen ZC (2013) An automated and accurate CNC programming approach to five-axis flute grinding of cylindrical end-mills using the direct method. J Manuf Sci Eng 135
5. Ren L, Wang S, Yi L (2018) A Generalized and efficient approach for accurate five-axis flute grinding of cylindrical end-mills, J Manuf Sci Eng 140
6. Nguyen VH, Ko SL (2013) Determination of workpiece profile and influence of singular point in helical grooving. CIRP Ann 62:323–326
7. Xiao S, Wang L, Chen ZC, Wang S, Tan A (2013) A new and accurate mathematical model for computer numerically controlled programming of 4Y1 wheels in 2½-axis flute grinding of cylindrical end-mills, J Manuf Sci Eng 135
8. Jia K, Hong J, Zheng S, Zhang Y (2017) An approach on wheel position and orientation calculation for helical broaching tool sharpening. Int J Adv Manuf Technol 92:1991–2000
9. Chen Z, Ji W, He G, Liu X, Wang L, Rong Y (2018) Iterative based calculation of position and orientation of grinding wheel for solid cutting tool flute grinding. J Manuf Process 36:209–215
10. Wang L, Chen ZC, Li J, Sun J (2016) A novel approach to determination of wheel position and orientation for five-axis CNC flute grinding of end mills. Int J Adv Manuf Technol 84:2499–2514
11. Ren L, Wang S, Yi L, Sun S (2016) An accurate method for five-axis flute grinding in cylindrical end-mills using standard 1V1/1A1 grinding wheels. Precis Eng 43:387–394
12. Karpuschewski B, Jandeka K, Mourek D (2011) Automatic search for wheel position in flute grinding of cutting tools. CIRP Ann 60:347–350
13. Li G, Zhou H, Jing X, Tian G, Li L (2017) An intelligent wheel position searching algorithm for cutting tool grooves with diverse machining precision requirements. Int J Mach Tools Manuf 122:149–160
14. Habibi M, Chen ZC (2017) A generic and efficient approach to determining locations and orientations of complex standard and worn wheels for cutter flute grinding using characteristics of virtual grinding curves. J Manuf Sci Eng 139
15. Fang Y, Wang L, Yang J, Li J (2020) An accurate and efficient approach to calculating the wheel location and orientation for CNC flute-grinding. Appl Sci 10:4223
16. Liu X, Chen Z, Ji W, Wang L (2019) Iteration-based error compensation for a worn grinding wheel in solid cutting tool flute grinding. Procedia Manuf 34:161–167
17. Uhlmann E, Schröer N, Muthulingam A, Gülzow B (2019) Increasing the productivity and quality of flute grinding processes through the use of layered grinding wheels. Procedia Manuf 33:754–761
18. Wasif M, Iqbal SA, Ahmed A, Tufail M, Rababah M (2019) Optimization of simplified grinding wheel geometry for the accurate generation of end-mill cutters using the five-axis CNC grinding process. Int J Adv Manuf Technol 105:4325–4344

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.