Journal of Intelligent Manufacturing (2024) 35:2787–2810
https://doi.org/10.1007/s10845-023-02175-4

A novel interpretable predictive model based on ensemble learning and differential evolution algorithm for surface roughness prediction in abrasive water jet polishing

Shutong Xie[1] · Zongbao He[1] · Yee Man Loh[2] · Yu Yang[3] · Kunhong Liu[4] · Chao Liu[5] · Chi Fai Cheung[2] · Nan Yu[6] · Chunjin Wang[2]

Received: 18 February 2023 / Accepted: 26 June 2023 / Published online: 22 July 2023

Abstract

As an important indicator of the surface quality of workpieces, surface roughness has a great impact on production costs and the quality performance of the finished components. Effective surface roughness prediction can not only increase productivity but also reduce costs. However, the current methods for surface roughness prediction have some limitations. On the one hand, the prediction accuracy of classical experimental and statistical-based surface roughness prediction methods is low. On the other hand, the results of deep learning-based surface roughness prediction methods are uninterpretable due to their black-box learning mechanism. Therefore, this paper presents an ensemble learning with a differential evolution algorithm, applies it to the prediction of surface roughness of abrasive water jet polishing (AWJP), and conducts an interpretability analysis to identify key factors contributing to the prediction accuracy of surface roughness. First, we proposed automatically constructing features by an Evolution Forest algorithm to train the base regression models. The differential evolution algorithm with a simplified encoding mechanism was then used to search for the best weighted-ensemble to integrate the base regression models for obtaining highly accurate prediction results. Extensive experiments have been conducted on AWJP to validate the effectiveness of our proposed methods. The results show that the prediction accuracy of our proposed method is higher than the existing machine learning algorithms. In addition, this is the first of its time for the contributions of machining parameters (i.e., features) on surface roughness prediction by using interpretable analysis methods. The analysis results can provide a reference basis for subsequent experiments and studies.

Keywords Surface roughness prediction · Polishing · Finishing · Interpretable machine learning · Ensemble learning · Ultra-precision machining

Introduction

The machining process to obtain the ideal surface quality is usually a basic technical requirement of machining, the surface profile and roughness of the component are two important indicators of the surface quality. Surface roughness, as an important indicator widely used to measure the surface properties of the workpiece, has an important impact on the performance of mechanical parts as well as production cost. In polishing processing, reducing the surface roughness

✉ Yu Yang
cs-yu.yang@polyu.edu.hk

✉ Chunjin Wang
chunjin.wang@polyu.edu.hk

[1] School of Computer Engineering, Jimei University, Xiamen 361021, China

[2] State Key Laboratory of Ultra-precision Machining Technology, Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

[3] Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

[4] Department of Digital Media Technology, Xiamen University, Xiamen 361005, China

[5] College of Engineering and Physical Sciences, Aston University, Birmingham B47ET, UK

[6] Institute for Materials and Processes, School of Engineering, University of Edinburgh, Edinburgh EH9 3FB, UK

of the workpiece and improving its surface quality are the main goals of all kinds of polishing processes. To reduce the surface roughness of a mechanical part without changing the geometry, various polishing processes are often used, and these operations can be time-consuming and expensive. Therefore, in the field of polishing processing, more and more researchers focus on surface roughness. Optimizing and predicting surface roughness can effectively increase productivity and reduce costs, which have important research significance and application value for the machining processes.

Numerous studies have shown that the surface roughness produced during machining is influenced by several machining parameters. Different machining methods, such as turning, milling, and polishing, have different influences on surface roughness. The process of surface roughness reduction is influenced by the properties of the workpiece and various uncontrollable factors, making it very difficult to obtain an accurate prediction model directly (Lu, 2008). Therefore, many researchers have worked on the optimization of various machining parameters and their effects on the prediction of surface roughness. Through the research, it is found that the research methods on surface roughness of polishing processing are divided into two main categories: classical methods based on experimental and mathematical analysis and methods based on artificial intelligence (AI). The classical methods include experimental analysis (Huang et al., 2022), statistical analysis (Solheid et al., 2020) and response surface methodology (RSM) (Nguyen et al., 2020; Jian et al., 2022). In AI-based methods, a portion of the research applied neural network algorithms for surface roughness prediction. For example, Schneckenburger et al. (2020) developed an artificial neural network (ANN) model to predict the surface roughness and shape accuracy of the glass after polishing, which was helpful to reduce the polishing iterations, thus reducing the production time. Besides, swarm intelligence algorithms such as PSO (Wang et al., 2020) and GA (Khalick Mohammad et al, 2017) are common techniques applied in this field. The tendency during the latest years is the research of machine learning techniques. As a result, the combination of machine learning algorithms and swarm intelligence algorithms has become a common method for optimizing machining parameters, and this type of method can effectively reduce surface roughness(Fan et al., 2022a; Khalick Mohammad et al., 2017).

The above analysis shows that, for classical methods, simple mathematical models often could not achieve the desired prediction accuracy. Although AI-based methods can improve the prediction accuracy of the model to a certain extent, complex algorithms similar to neural networks often require a large amount of data support for model training to ensure the reliability of the prediction results. However, in the actual machining process, only a small amount of experimental data can be collected for each experiment due to various conditions such as experimental environment, technology, and cost budget. Too few data samples can easily lead to under-fitted neural network models, thus affecting the accuracy and reliability of prediction results. In addition, the black-box prediction model based on neural networks can not explicitly explain the hidden correlation between the control parameters and the predicted values, thus failing to meet the needs of practical production.

Ensemble learning is a framework that accomplishes learning tasks by building and combining multiple base learners. The framework often achieves better generalization performance than a single learner by combining multiple weak learners with simple structures. Ensemble learning, a widely used AI technique, provides new ideas for our research. On one hand, weak learners with simple structures require much fewer data than complex neural network algorithms. On the other hand, the ensembled strong learner can obtain higher prediction performance. In addition, given the successful application of swarm intelligence algorithms for machining parameter optimization, a small number of studies have been conducted to apply GA to the prediction of surface roughness. For example, Wang et al. (2022a, b) proposed a robust surface roughness prediction model (ELGA) based on ensemble learning of GA, which can be used for the prediction of the surface roughness of multi-jet polishing (MJP) of 3D printed 316 L stainless steel. Inspired by this, we propose to consider a combination of a swarm intelligence algorithm and an ensemble learning algorithm for surface roughness prediction in abrasive water jet polishing, since it has a similar material removal mechanism to MJP (Wang et al., 2017a, b). Given the successful applications of the differential evolution (DE) algorithm in industry in recent years (Yuan et al., 2021; Ibrahim & Tawhid, 2022), we combine the DE algorithm, an emerging technique for swarm intelligence, with an ensemble learning approach. Like other swarm intelligence algorithms, DE is a stochastic optimization algorithm that simulates biological evolution. Compared with GA, DE retains the population-based global search strategy and reduces the complexity of genetic operations by using real number encoding, simple difference-based variation operations, and one-to-one competitive survival strategies.

A novel ensemble framework for polished surface roughness prediction is designed in this paper. The method selects six classical regression algorithms, Random Forest (RF), Extreme Gradient Boosting (XGBoost), Least Absolute Shrinkage and Selection Operator (LASSO), Support Vector Regression (SVR), Gradient Boosting Regression (GBR), and Extra-Tree Regressor (ETR), as the base models and uses DE to search for optimal weight assignment schemes for different models. Notably, we design a simplified encoding mechanism for the individual design of DE for the characteristics of polishing processing, which can further improve the computational efficiency of the algorithm. The experimental results show that the ensemble framework can further improve the prediction accuracy of the algorithm for the low-dimensional small-sample experimental data used in this study. Due to the small amount of experimental data, although the underlying models are prone to underfitting, ensemble learning can maximize the advantages of each model to complement each other, so that the whole algorithm can achieve high prediction accuracy even with small sample data. In addition, we use genetic programming (GP) based Evolutionary Forest (EF) algorithm for automatic feature construction in the data pre-processing stage to further improve the prediction performance and interpretability of the model.

The rest of this paper is organized as follows. Section Research progress of surface roughness prediction reviews the related research on surface roughness in polishing processes. Section ELDEA for surface roughness prediction describes the ELDEA method based on ensemble learning and DE for surface roughness prediction. Section Validation experiments gives a detailed experimental setup. Section Results and discussion compares and analyzes the performance of different prediction methods and provides interpretability analyses of the data features. Section Conclusions and future work summarizes our work and proposes future research directions.

Research progress of surface roughness prediction

By investigating the research related to surface roughness, we have grouped the related research to surface roughness into two main categories: reducing surface roughness by optimizing machining parameters and direct prediction of surface roughness. These two approaches optimize surface roughness indirectly and directly, respectively, and have in common that both of them utilize the relationship between machining parameters and surface roughness. Currently, a large number of studies have focused on the optimization and prediction of surface roughness for turning and milling, and less research has been done on polished surface roughness. As a result, this section reviews and analyzes the current status of polished surface roughness research.

Reducing surface roughness by optimizing machining parameters

Reviewing the previous studies, the methods for minimizing surface roughness by optimizing machining parameters can be divided into three main categories, namely, optimization methods based on experimental and mathematical analysis, optimization methods based on machine learning, and optimization methods based on swarm intelligence.

Before machine learning was widely used in manufacturing, the traditional method of combining experimental models and RSM applied to the study of surface roughness was the dominant research direction at that time. Huai et al. (2017) developed a roughness ratio prediction model based on orthogonal experiments for the polishing processing bladeslade of aviation engines. Chen et al. (2019) performed a regression analysis based on the results of Central-Composite designed polishing experiments and established a prediction model for surface roughness and residual stress. In addition, a comprehensive analytical approach was proposed to optimize the process parameter intervals to obtain the globally optimal parameter intervals for improving the surface quality during the polishing of Ti-6Al-4 V blades.

With the development of swarm intelligence algorithms, researchers have started to apply them to the optimization of machining parameters. To achieve the optimum balance between high material removal rate (MRR) and low surface roughness, Chen et al. (2018) established a prediction model for MRR based on linear regression analysis of experimental data for the belt flap wheel polishing of TC4 aero-engine blisk blades and then solved a single-objective constrained optimization problem by using PSO. Fan et al. (2022b) proposed a novel finishing parameter optimization method considering dimensional accuracy for magnetic compound fluid finishing (MCFF), which makes use of a multi-objective PSO algorithm to optimize the finishing parameters.

In recent years, machine learning technology has emerged rapidly and has been successfully applied in the field of manufacturing. Wei & Wu (2022) proposed a graphical model and a conditional variational autoencoder to extract the features of surface topography in the CMP process. Moreover, process variables and the extracted features of surface topography are fed into an ensemble learning-based predictive model to predict the MRR. To achieve high quality polishing of an M300 mold steel curved surface, Tong et al. (2019) first obtained the influence degree of various factors on polished surface roughness and the combination of parameters to be optimized by the signal-to-noise ratio method and then combined BP and PSO (PSO-BP) to optimize the polishing parameters. The results showed that the method can effectively reduce surface polishing roughness. Wang et al. (2012) developed a series-parallel hybrid polishing machine tool based on the elastic polishing theory, applied it to finish mould surface with using bound abrasives and optimized its by using a combination of GA and ANN to obtain the optimal process parameters.

It can be found that most of the surface roughness optimization methods based on machine learning methods choose to combine swarm intelligence methods to further reduce the surface roughness of workpieces and improve their surface quality. Zhong et al.(2021) reviewed studies related to surface roughness of wood and advanced engineering materials after machining and its prediction and found that ANN trained using PSO and GA have better prediction performance than traditional ANN. The results of this study further demonstrated the advantages of combining swarm intelligence algorithms with machine learning methods applied to the prediction of surface roughness.

Surface roughness prediction method

In addition to the reduction of surface roughness by optimizing machining parameters, a part of the studies tried to model surface roughness prediction directly. Benardos et al. (2003) classified the modeling techniques for surface roughness prediction into three categories, namely, experimental models, analytical models, and AI-based models. Among them, surface roughness prediction models based on AI methods are the main research direction in this field, and the AI-based methods are divided into three branches, including swarm intelligence algorithms, traditional machine learning algorithms, and artificial neural networks. On this basis, we analyzed and summarized the related research on polished surface roughness prediction, and summarized the polished surface roughness modeling techniques into two categories, which are classical modeling methods based on experimental and mathematical analysis and modeling methods based on AI techniques.

Aimed to reduce surface roughness and improve surface quality of mold steel, Xie et al. (2022) optimized the robotic polishing process parameters and obtained the optimal range of each parameter according to a single factor experiment. A surface roughness prediction model was established by the central composite design experiments on the three polishing parameters. Xu et al. (2021) proposed a numerical model for the prediction of surface roughness after laser polishing, which coupled heat transfer, fluid flow, and material vaporization. The surface roughness prediction after laser polishing was tried through the numerical model.

The design of surface roughness prediction methods based on classical methods requires a great deal of expertise on the part of the researcher. Besides, these approaches often require researchers to perform complex mathematical modeling of the relationship between machining parameters and surface roughness. Due to the data-driven black-box mechanism of AI techniques, researchers can perform surface roughness prediction and modeling tasks without much expertise. As a result, more and more studies have started to use AI techniques to solve surface roughness prediction problems and found that AI-based surface roughness prediction models have higher prediction accuracy compared to classical methods. Electrochemical mechanical polishing (ECMP) results are influenced by many factors, predicting its surface quality and the determination of its processing parameters difficult to achieve. To address these issues, Xu et al. (2012) developed a least squares support vector machine with radial basis function based on ECMP prediction model and investigated the effect of polishing parameters on surface roughness through orthogonal experiments.

In addition to using classical machine learning algorithms, many researchers have explored the application of neural networks in surface roughness prediction. Schneckenburger et al. (2020) developed an ANN model for predicting the glass-surface roughness and shape accuracy. The experimental results showed that the prediction of ANN reduces the polishing iterations, thus reducing the production time. Deng et al. (2021) investigated the effects of chemical mechanical polishing (CMP) process parameters on the (MRR) and surface roughness of single-crystal SiC after polishing and their relationships based on the modified Preston equation and established a prediction model for CMP process parameters, MRR and post-polishing surface roughness based on BP neural networks. Qi et al. (2018) developed a prediction model for surface roughness in belt polishing based on a BP neural network with the maximum depth of cut of abrasive grains, the rotation speed of belt and the feed rate as input parameters. The model requires only fewer experimental samples to complete the prediction with higher accuracy, thus saving the experiment cost and time.

Current challenges and reflections

In summary, for polishing processes, the current surface roughness prediction methods based on classical methods and artificial neural networks have their advantages, but also have their limitations at the same time, as described in the introduction section.

As a widely used artificial intelligence technique, ensemble learning is widely used in the study of various regression prediction problems, and this method has better performance than traditional mathematical methods. Due to the simple structure of the base model and the excellent performance of the ensemble framework, ensemble learning can achieve a generalization performance no less than that of neural network models without the support of large amounts of experimental data, and the commonly used interpretable methods can provide interpretable analysis for the ensemble learning model. In addition, given that the swarm intelligence algorithm has been widely and successfully applied to machining parameter optimization problems, we intend to consider using a combination of the swarm intelligence algorithm and ensemble learning for surface roughness prediction modeling to achieve a more accurate prediction of surface roughness.

It is found that a large number of surface roughness studies based on swarm intelligence algorithms use group swarm intelligence algorithms such as GA or PSO, and few people have applied DE to the study of surface roughness. However, compared with other algorithms, DE retains the global search capability while its simple variational operation based on the difference principle makes the algorithm itself less complex and more efficient. Therefore, this paper proposes an ensemble learning model based on DE, which is used for the prediction of surface roughness of polishing processes. Compared with other methods, the proposed method in this paper has some interpretability while ensuring prediction accuracy. First, the ensemble model based on DE weighs the fitting effect of each basic regression model to realize the complementary advantages of the basic models, which makes the ensemble model avoid underfitting when training on small samples of low-dimensional processing data, and thus improves the prediction accuracy. Secondly, interpretability analysis methods can provide us with a more comprehensive understanding of the data and model, and provide more meaningful information to help us improve the data and model, creating a positive feedback loop. Besides, we designed a DE-based simplified encoding mechanism specifically for the surface roughness prediction problem of small sample polishing processing and applied it to the individual design of the DE algorithm, which solved the problems of low efficiency and poor accuracy of traditional coding mechanisms. In addition, we used a GP-based EF algorithm for feature reconstruction of experimental data to further improve the prediction accuracy and interpretability of the model. Finally, we performed interpretability analyses of the model and data at the level of features, and innovatively identified the main processing parameters affecting surface roughness based on the results of interpretability analysis, which provided a theoretical reference basis for subsequent experiments and research.

ELDEA for surface roughness prediction

In this paper, an ensemble surface roughness prediction framework based on DE is presented. The method is applicable in several practical machining cases under polishing, and the effects of different machining parameters on surface roughness for different workpiece materials are considered. Figure 1 shows the framework of our ELDEA method. The ELDEA consists of five parts: (1) data normalization, (2) feature construction, (3) multi-algorithm regression, (4) DE-based ensemble learning, and (5) interpretability analysis.

Data normalization

In machine learning, it is always necessary to adjust data of different scales to the same common scale, or to transform data of different distributions to a specific distribution, which is collectively called "nondimensionalization". In gradient and matrix-based algorithms, such as logistic regression, support vector machines, and neural networks, nondimensionalization can speed up the training; in distance-based algorithms, such as K-nearest neighbor and K-Means clustering, nondimensionalization can help improve model accuracy by avoiding the impact of a particularly large absolute value of a feature on the distance calculation. There are various ways to implement dimensionless scaling, such as standard scaler and Min-Max scaler. Since Min-Max normalization is very sensitive to outliers, most machine learning algorithms choose standard normalization for feature scaling. Therefore, we employ standard normalization to pre-process the experimental data as shown in Eq. (1), where μ is the mean of all sample data and σ is their standard deviation. After normalization, the data are in the standard normal distribution in which the mean is 0 and the standard deviation is 1.

$$x_{normalization} = \frac{x - \mu}{\sigma} \tag{1}$$

Feature construction

Extracting robust features from raw data is one of the most critical steps to train a highly accurate prediction model. Researchers often spend a lot of time and effort on feature engineering to obtain more important data information from raw data and train machine learning models for a highly accurate prediction. Feature construction is a process that extracts and embed characteristics, properties, attributes, and underlying patterns of data as features (Li et al., 2015), which is an important step in feature engineering and directly determine the quality of the analysis results in data mining.

Constructing high-quality features are extremely challenging due to a lack of relevant domain knowledge. To overcome this difficulty, some automatic feature construction methods based on evolutionary algorithms have been proposed (Virgolin et al., 2018; Tran et al., 2019). Inspired by these methods, we propose to employ an EF algorithm (Zhang et al, 2021) to construct features automatically for training the base regression models. By searching the most effective feature space to represent the data, our approach not only optimizes the feature quality but also improves the generalization ability of the regression model. The workflow of the EF algorithm is shown in Fig. 2. First, the algorithm initializes to form a population consisting of multi-tree GP individuals, i.e., each individual in the population is a set of GP trees. In each individual Φ_i, the EF algorithm then constructs a decision tree by using the original features of the artifact (e.g., φ_1, φ_2, φ_3, …) to obtain an initial set of features for each individual. Lastly, new features are obtained though the evolution. These new features can not

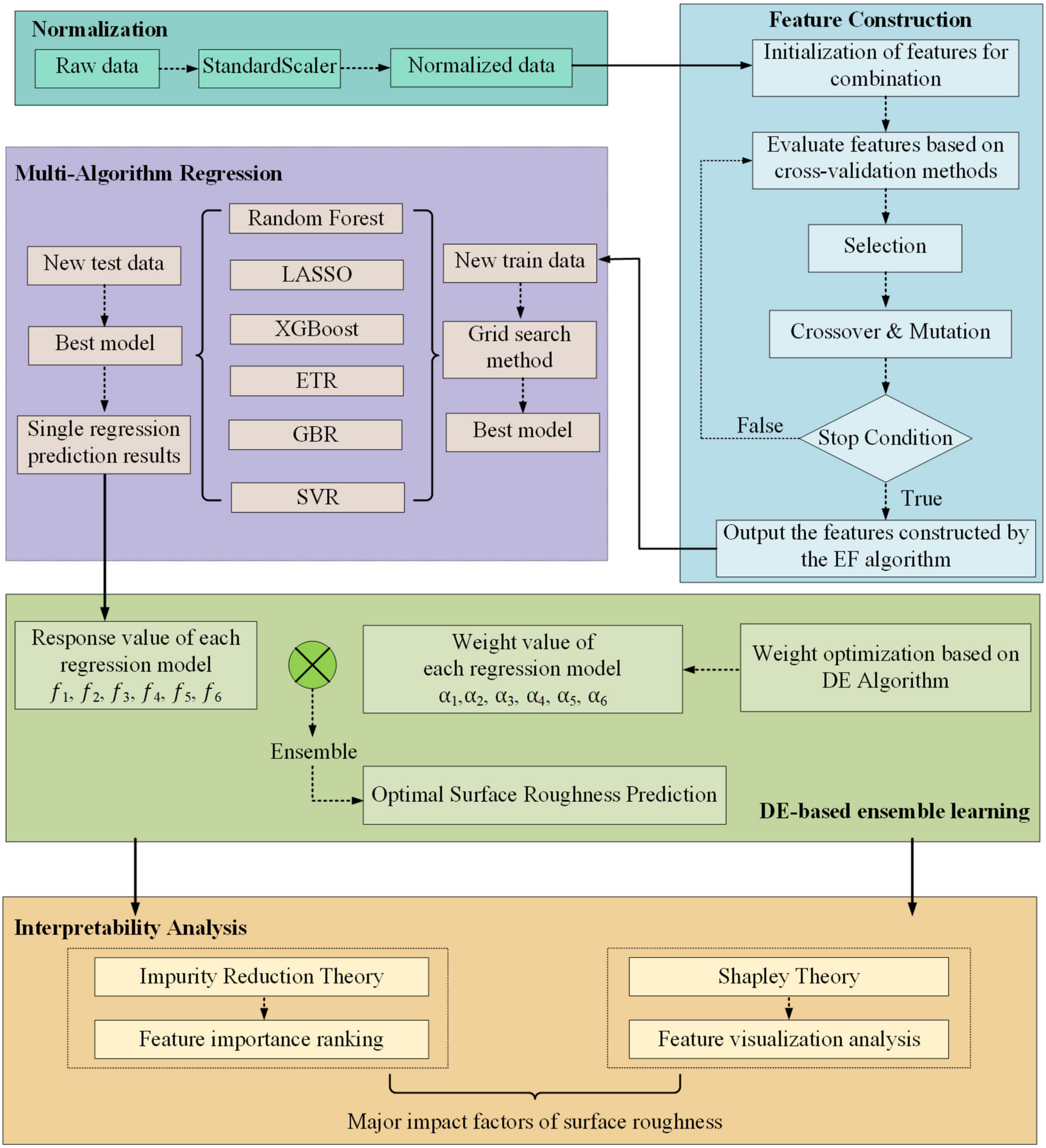

Normalization
Raw data
StandardScaler
Normalized data
Feature Construction
Initialization of features for combination
Evaluate features based on cross-validation methods
Selection
Crossover & Mutation
False
Stop Condition
True
Output the features constructed by the EF algorithm
Multi-Algorithm Regression
New test data
Best model
Single regression prediction results
Random Forest
LASSO
XGBoost
ETR
GBR
SVR
New train data
Grid search method
Best model
Response value of each regression model
$f_1, f_2, f_3, f_4, f_5, f_6$
Weight value of each regression model
$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6$
Weight optimization based on DE Algorithm
Ensemble
Optimal Surface Roughness Prediction
DE-based ensemble learning
Interpretability Analysis
Impurity Reduction Theory
Feature importance ranking
Shapley Theory
Feature visualization analysis
Major impact factors of surface roughness

Fig. 1 Framework of ELDEA

only increase the feature diversity and avoid model underfitting but also improve the prediction accuracy. During the evolutionary process, the EF algorithm uses crossover, mutation, and selection operators to update individuals in the population, as shown in Fig. 2. Taking the crossover operation as an example, for individual Φ_1 and individual Φ_2, we select a certain tree in each of the two individuals and swap the values of two nodes X2 and X3 in the tree, thus two new GP trees can be generated, corresponding to two new individual $\Phi_1{}'$ and individual $\Phi_2{}'$.

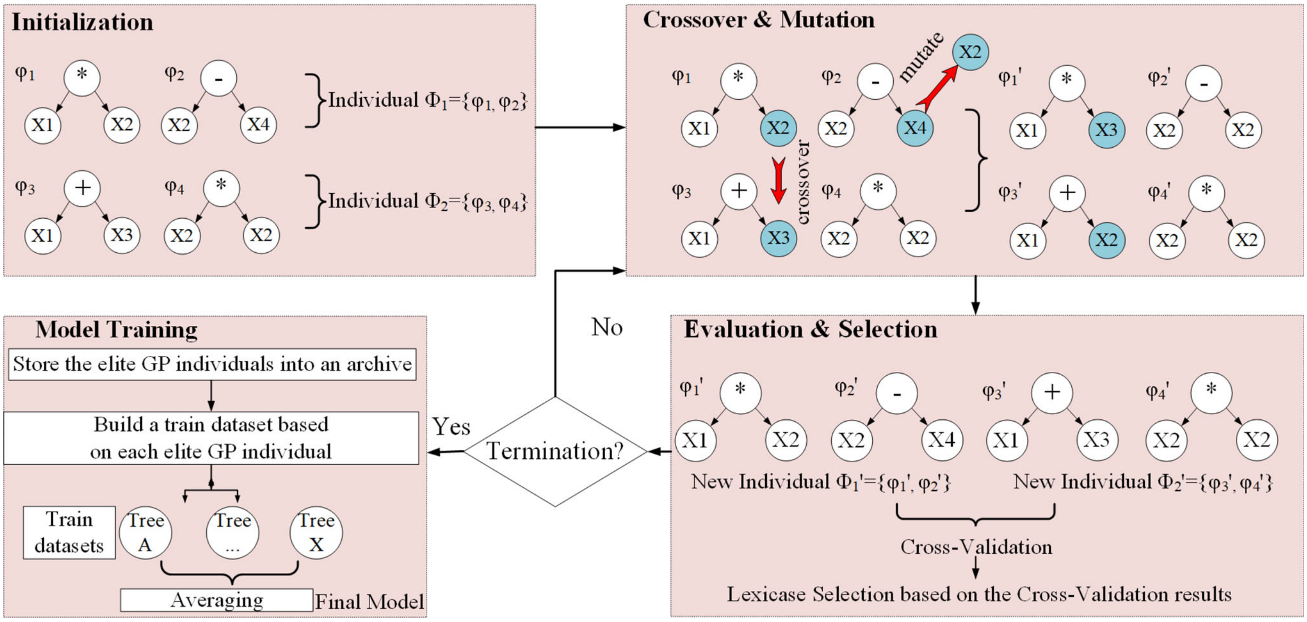


Fig. 2 An illustration of the workflow of EF algorithm

To make the algorithm has better generalization ability on the regression tasks, the EF algorithm uses 5-fold cross-validation to estimate the generalization loss when evaluating each individual's fitness value. First, the raw dataset generated by the AWJP experiments will be standardized and randomly divided into two subsets: the training set and the test set in the proportion of 3:1. And then, the training set is divided into 5 equal-sized folds. Among them, 4 folds are used for automatic feature construction to fit each decision tree model in the EF, and the other is used as a validation set to verify the validity of the automatic feature construction. The training set and the validation set are used for the 5-fold cross-validation during the evolutionary process to determine the features. In our implementation, we adopt the absolute deviation as the loss value, as shown in Eq. (2), where f represents a decision tree, X_j represents the j-th sample, and Y_j represents the j-th target values. A sample input to a decision tree model will return a prediction value $f\ (X_j)$. The difference between the predicted value and the true target value is the prediction error L_j of the j-th sample. The prediction error of all validation data points is recorded each time the decision model is built. This error is used as the fitness function of the algorithm to guide the iterations of the algorithm.

$$L_j = \left| f\left(X_j\right) - Y_j \right| \tag{2}$$

The well-performing GP individuals are stored in an archive, and each GP individual in that archive is used to construct a training set. Then, we trained a decision tree for each constructed training set. At the end of the evolutionary process, all these decision trees form a forest, i.e., a target forest. At this point, we can construct valid features from this forest for subsequent ensemble learning.

Multi-algorithm regression

In the multi-algorithm regression module, we leverage the base regression models to build an ensemble surface roughness prediction framework. We first trained ten base regression models and tuned the hyperparameters for each of them to reach the optimal performance. In the experiment, we found that some of the regression algorithms were underfitting in the experimental dataset. Therefore, we selected six of them who are well trained and converged to build the multi-algorithm regression module. The six selected models are ETR (Geurts et al., 2006), RF (Breiman, 2001), XGBoost (Chen & Guestrin, 2016), SVR (Awad & Khanna, 2015), LASSO (Tibshirani, 1996) and GBR (Friedman, 2001). Table 1 lists the details of these models and their regression errors. It should be noted that the experimental data are derived from AWJP experiments based on the 3D-printed components CoCr. Detailed descriptive information about the data is given in Sect. 4.1. The training data are fitted by each algorithm and the values of surface roughness are predicted by each of them according to the test data. Specifically, we normalized the raw data and then used the hold-out method to initially validate the performance of each base model. In the process of data segmentation, the raw data will be divided into four folds, three of which will be used to train the base model, and the remaining part of the data will be used to test the trained model to obtain its prediction error. The above process is repeated five times, and the average of the five

Table 1 Different machine learning models

No.	Regression Model	Refs.	MSE (μm)	MAE (μm)
1	Extra Trees Regression	(Geurts et al., 2006)	0.2054	0.2035
2	Random Forests (RF)	(Breiman, 2001)	0.2458	0.1909
3	Extreme gradient boosting (XGBoost)	(Chen & Guestrin, 2016)	0.2122	0.2372
4	Support Vector Regression (SVR)	(Awad & Khanna, 2015)	0.1987	0.3420
5	Least absolute shrinkage and selection operator (LASSO)	(Tibshirani, 1996)	0.2640	0.4319
6	Gradient Boosting Regression(GBR)	(Friedman, 2001)	0.2640	0.4689
7	Linear Regression	(Freedman, 2009)	0.4079	0.4173
8	Elastic Net Regression (ENR)	(Zou & Hastie, 2005)	0.4221	0.5216
9	Ridge Regression (RR)	(Hilt & Seegrist, 1977)	0.5489	0.5335
10	Stochastic Gradient Descent Regression	(Shalev-Shwartz et al., 2011)	0.5755	0.5280

results are taken as the final prediction error of the current base regression model. Finally, the prediction results of the test dataset on each regression model and the corresponding errors are recorded as the baseline performance of the ensemble framework.

DE-ensemble module

Due to the very small size of the data and low dimensional features, each base regression model easily underfits in the training set, thus leading to very low prediction accuracy. In contrast, the ensemble framework can integrate the output results of multiple base regression models, so that the advantages of each model complement each other. Therefore, we employ an ensemble framework for surface roughness prediction and propose using the DE algorithm to pursue the best ensemble weights for each base regression model so as to maximize the integration effect and results in the best prediction accuracy.

DE algorithm, first proposed by Storn and Price in 1995 (Ahmad et al., 2021), is a population-based adaptive global optimization algorithm for solving real number optimization problems. As a common evolutionary algorithm, DE is widely used in various fields such as data mining, pattern recognition, digital filter design, artificial neural networks, and electromagnetism by its simple structure, easy implementation, fast convergence, and robustness. The evolutionary process of DE is very similar to that of GA, both include mutation, difference calculation, and selection operations. However, compared with GA, DE retains the population-based global search strategy, and the simple mutation operation based on difference and one-to-one competitive survival strategy reduces the complexity of the genetic operation.

Since DE has a simple structure and is easy to implement, we use the DE algorithm as a search engine in this module and design a simplified encoding mechanism for the individual design of the DE algorithm according to the problem to be optimized. This module is mainly responsible for integrating and processing the output of multiple algorithm modules, searching for the best weight assignment for each base regression model through the DE-based simplified encoding mechanism, and integrating each regression module according to the optimal solution obtained from the search to obtain the final surface roughness prediction value.

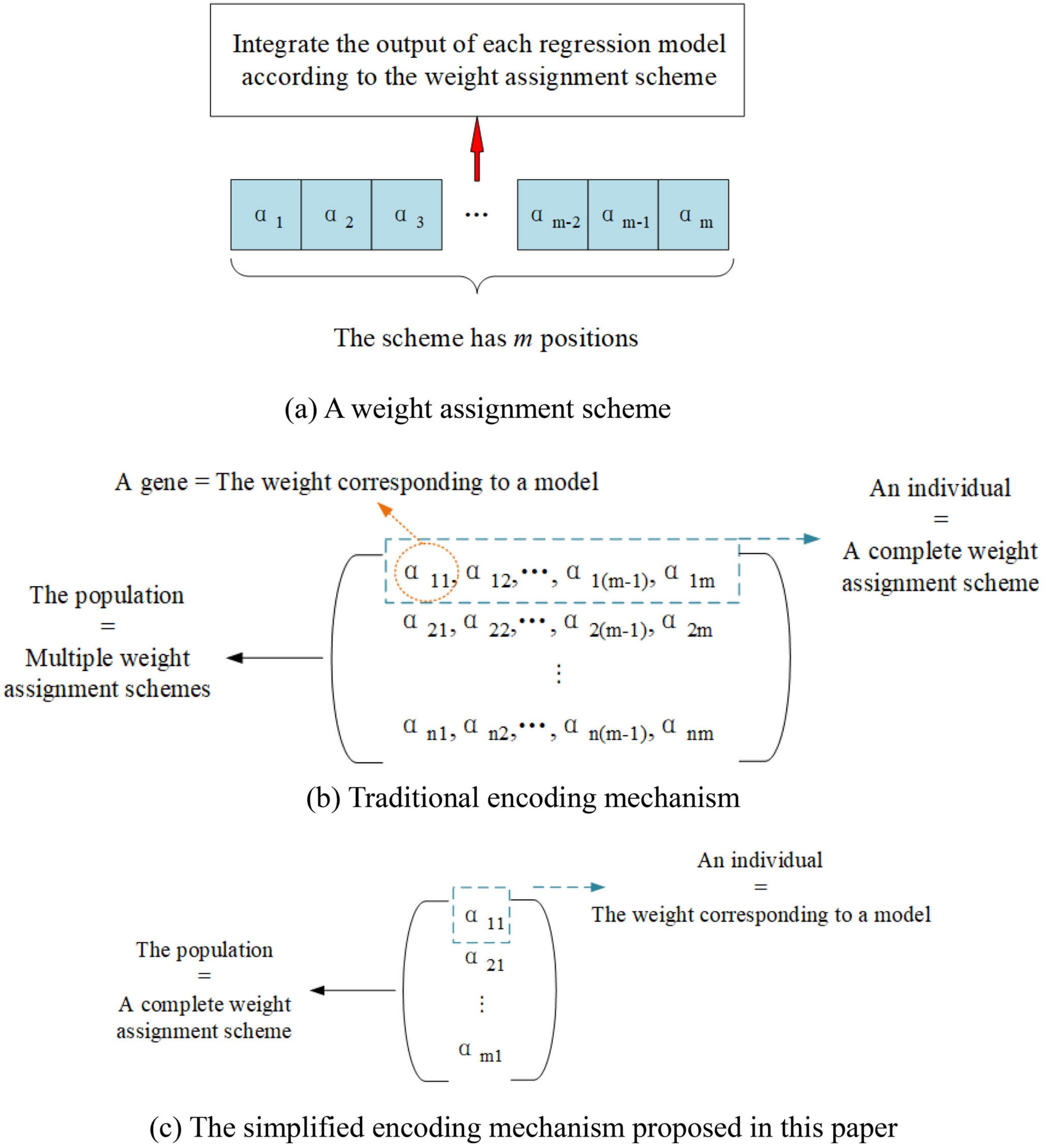
Individual design

DE is a population-based heuristic search algorithm which is very efficient in search the global optimal. In DE, each individual in the population corresponds to a solution vector, and each individual is regarded as a target vector, as shown in Eq. (3), where *D* is the dimension of the target vector, and *NP* denotes the number of individuals in the population.

$$\bar{x_i} = \left(x_{i,1}, x_{i,2}, \ldots, x_{i,D}\right) \quad i = 1, 2, \ldots, NP \tag{3}$$

The individual design scheme of the algorithm is shown in Fig. 3a. When applying DE to search the weights of each regression model, each individual usually represents a weight assignment scheme that integrates each regression model. Where α_i denotes the specific weight assignment value of the *i*-th regression model in that individual, $i = 1,2,\ldots,m$, and m denotes the number of base models, which is a traditional coding method as shown in Fig. 3b. In the traditional encoding method, a population contains *n* individuals with *m* dimensions in each individual, indicating that the algorithm can provide a total of *n* weight assignment schemes for *m* models.

However, during the evolution, a large number of individual solutions are generated, and since this optimization problem is a real number encoding problem, it may lead


Integrate the output of each regression model according to the weight assignment scheme
α 1
α 2
α 3
…
α m-2
α m-1
α m
The scheme has m positions
A gene = The weight corresponding to a model
An individual = A complete weight assignment scheme
The population = Multiple weight assignment schemes
α 11, α 12, …, α 1(m-1), α 1m
α 21, α 22, …, α 2(m-1), α 2m
α n1, α n2, …, α n(m-1), α nm
An individual = The weight corresponding to a model
The population = A complete weight assignment scheme
α 11
α 21
α m1

(a) A weight assignment scheme

(b) Traditional encoding mechanism

(c) The simplified encoding mechanism proposed in this paper

Fig. 3 Difference between the traditional encoding mechanism of DE and our simplified encoding mechanism

to an excessive dimensionality of the search space and thus consume a large amount of computation time. Therefore, we design a simplified encoding mechanism to tackle this problem. This simplified coding mechanism reduces the dimensions of the search space from *D* to 1, which speeds up the search. In addition, it allows us to further improve the efficiency of the algorithm by eliminating the need to spend a lot of time and effort on tuning the hyperparameters (i.e., the size of the population). Figure 3b and c depict the differences between the traditional encoding mechanism and our simplified encoding mechanism of DE proposed in this paper. In our proposed simplified encoding mechanism, the dimension of each individual is only one and it is a real number in the range [0,1], indicating the weight assignment of the regression model corresponding to that individual. The whole population corresponds to the set of weight assignments of a complete ensemble framework, and the sum of all individual values in the population is 1, indicating that the weights of each model sum to 1. Based on this design scheme, an individual is a real number α_i of length 1 ($\alpha \in [0,1]$) and α_i denotes the weight of model i in the integration framework as α. A population is a sequence of real numbers [α_1, α_2, α_3, …, α_{m-1}, α_m] of length 6 and $\sum \alpha_i = 1$.

Crossover, mutation, and selection operators

Before the iteration of the algorithm, we first randomly generate a number of new individuals that satisfy the conditions according to the individual design rules to form the initial population. Next, the population is iteratively updated using crossover, mutation, and selection operators to search for the optimal solution. The evolutionary process of DE is very similar to that of GA, containing the basic operations of mutation, crossover, and selection, but the specific definitions of these operations are different from those of the genetic algorithm. For the variation operation, the mutation operator generates a mutation vector v_i for each target vector x_i (i.e., individual) according to Eq. (4), where NP denotes the number of individuals in the population and $r1$, $r2$ and $r3$ are [1, NP] randomly selected integers representing three randomly selected individuals from the population. f is the scale factor and is usually set to 0.8. x_{r2}-x_{r3} is the difference vector.

$$\overline{v}_i = \overline{x}_{r1} + F * \left(\overline{x}_{r1} * \overline{x}_{r2}\right), i = 1, 2, \ldots, NP \quad (4)$$

For the crossover operation, a new vector u_i is generated using the binomial crossover operator based on Eq. (5) for each pair of target vector x_i and variant vector v_i, where $rand_j$ is a random number in [0, 1] following uniform distribution, j_{rand} is a randomly chosen integer between 1 and D, and CR is a crossover control parameter and its range is in [0,1].

$$u_{i,j} = \begin{cases} v_{i.j}, & if rand_j \quad lt; CR \; or \; j = j_{rand} \\ x_{i,j}, & otherwise \end{cases}, \quad i = 1, 2, \ldots, NP, \quad j = 1, 2, \ldots, D \quad (5)$$

For the selection operation, we will compare the target vector x_i with the new vector u_i based on the fitness objective function f. The vector with a higher fitness value is selected for the next iteration.

$$u_{i,j} = \begin{cases} u_{i.j}, \; if \quad f(u_i) \geq f(x_i) \\ \quad x_{i,j}, \; otherwise \end{cases}, i = 1, 2, \ldots, NP \quad (6)$$

After a series of mutation, crossover, and selection operations, the population produces new individuals. This process is repeated until the maximum number of iterations is satisfied and the algorithm stops.

DE-based ensemble

In the multi-algorithm regression module, we can obtain the predicted values for each test sample in each base regression model, and these predicted values are called response values. Before the algorithm iteration, a number of individuals are first randomly initialized to form a population. Each individual in the population represents a certain regression model weight assignment, so the whole population corresponds to a model weight assignment scheme. During the iteration of the algorithm, the population is updated after evolutionary operations such as crossover, mutation, and selection, resulting in new weight assignment schemes. TThe weight value α_m of model m can be obtained from the weight assignment scheme formed by DE. The results of each base regression model can be integrated. Combining the response value f_m and weight value α_m of model m, the final surface roughness prediction value of a certain sample can be calculated according to Eq. (7). Then, the error between the predicted and actual values of the sample is used as the fitness function of DE to guide the DE algorithm for weight optimization of the integration framework. The detailed workflow of the ensemble calculation based on the DE algorithm is shown in Fig. 4. The integration is determined as shown in Eq. (7), where $\widehat{y}$ denotes the predicted surface roughness; f_m is the response value of each regression model; α_m is the weight of each model, and M is the number of models.

$$\widehat{y} = \sum_{m=1}^{M} \alpha_m f_m subject \sum_{m=1}^{M} \alpha_m = 1 \quad (7)$$

Interpretability analysis

Although machine learning algorithms such as neural networks have been successfully applied in various fields, it is a black-box that prevents end-users from understanding why such prediction results are made by the model. As a result, it

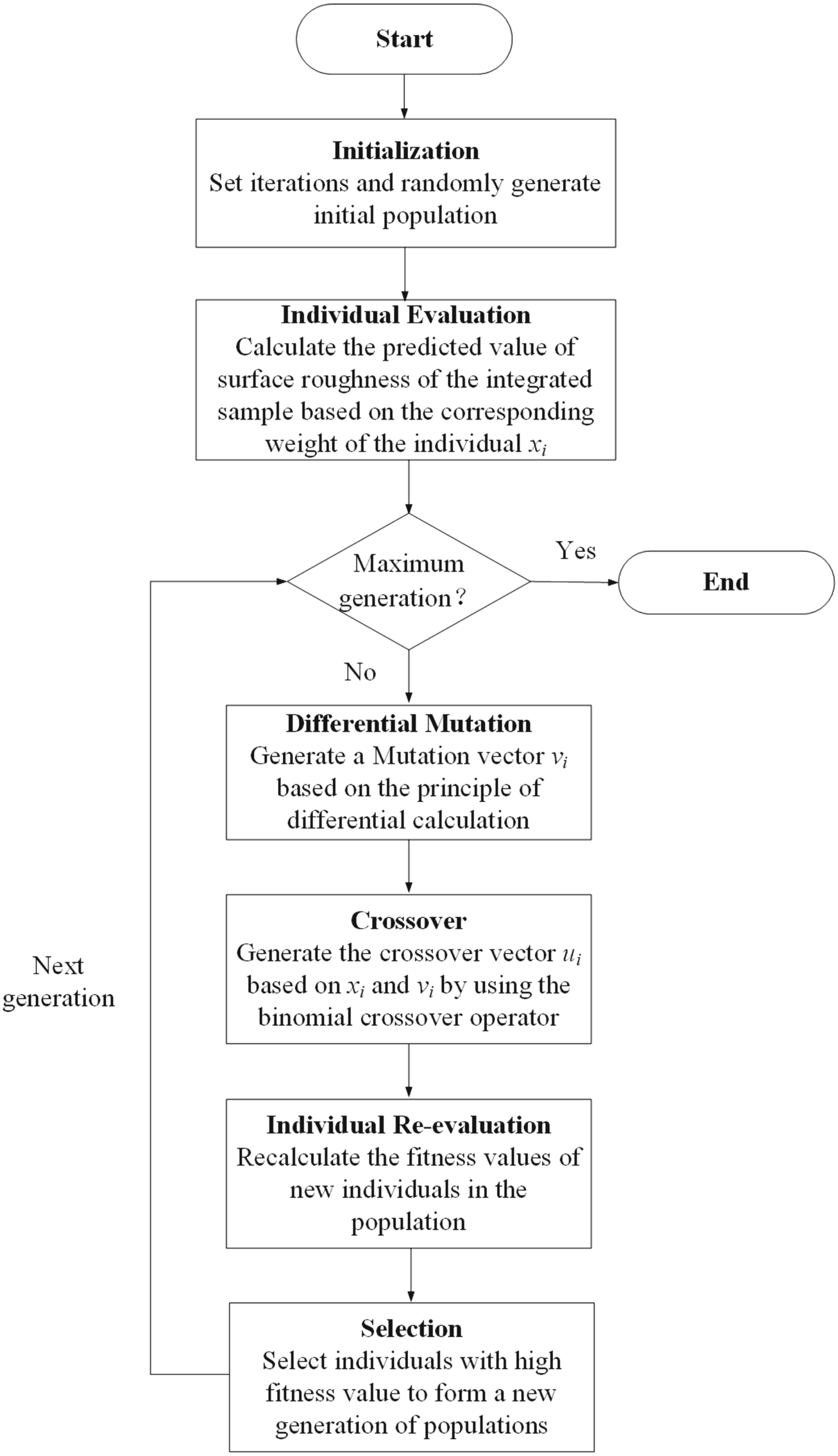

Start
Initialization
Set iterations and randomly generate initial population
Individual Evaluation
Calculate the predicted value of surface roughness of the integrated sample based on the corresponding weight of the individual x_i
Maximum generation?
Yes
End
No
Differential Mutation
Generate a Mutation vector v_i based on the principle of differential calculation
Crossover
Generate the crossover vector u_i based on x_i and v_i by using the binomial crossover operator
Individual Re-evaluation
Recalculate the fitness values of new individuals in the population
Selection
Select individuals with high fitness value to form a new generation of populations
Next generation

Fig. 4 Flow chart of DE-ensemble module

is difficult for data-driven machine learning models to be deployed and applied in many application fields, such as biomedicine and autonomous driving. To fully understand the decision basis of some models, the concept of interpretable machine learning has been proposed and become one of the hottest research areas in AI. Currently, the most popular approach in interpretating machine learning is interpretable analyses by looking at feature importance and feature relevance.

Therefore, we leverage interpretability analysis methods on the original features and the newly constructed features to uncover the insightful information contained in the original features and validate the usefulness of newly constructed features. By synthesizing the results of the analysis, we tried to identify the main processing parameters that affect the surface roughness work.

Interpretability analysis based on impurity reduction concept

The concept of impurity is derived from decision trees, where a lower impurity reduction indicates a more reasonable division of the corresponding branches in a decision tree. The Information Entropy and Gini Index are two common impurity indicators, which are often used to measure the effectiveness of feature selection. In the GP-based EF algorithm, it first randomly constructs a number of features, then calculates their importance scores based on impurity reduction. Then, it ranks the constructed features according to their importance scores. After that, the model constructs new features from these important features by optimizing the loss function so as to supervise the algorithm to search for high quality features. Ultimately, the constructed features can effectively improve the generalization ability of the model. Based on the concept of impurity reduction, we rank the constructed features according to their importance. At the end, by comparing the difference between the constructed features and the original features, we can explain the effect of the processing parameters corresponding to the feature on the surface roughness.

Interpretability analysis based on Shapley theory

Shapley was proposed by Shapley (1953) for solving the problem of contribution and benefit distribution in cooperative games. The concept is derived from game theory, where several gamblers cooperate to play a game, and how the final benefit is distributed is determined by the contribution of each gambler in the game. Now, borrowing this problem to machine learning, each feature is equivalent to a gambler, and the coefficient of the feature (under the linear model) is equivalent to the contribution, so the coefficient multiplied by the average value is the average contribution. For a single instance x_m, the contribution of the -th feature to the prediction ϕ_{ij} is the contribution of x_{ij} minus the average value, which in the linear case is $\beta_j E(X)$.

Shapley additive explanation (SHAP) is a Python package that can interpret the output of any machine learning model. The idea of SHAP is to calculate the marginal contribution of features to the model output and then interpret the black-box model at both global and local levels. SHAP constructs a linear model based on the best Shapley value in game theory. For each sample, the machine learning model gives a prediction value and SHAP considers all features as "contributors" and the Shapley value is the value assigned to each feature in that sample. SHAP represents the Shapley values as an additive feature imputation method that expresses the predicted value of the model as the sum of the imputed values of each input feature. The SHAP method is a typical post-hoc model interpretation method, which is mainly used to provide a visual interpretation of the samples and features in the model.

The Shapley theory differs from the impurity reduction theory. In the node division process of decision tree, based on the impurity reduction theory, the model will aim to reduce the impurity and find a suitable feature and feature value to make the result of this division can make the class attribution clearer. Therefore, the idea is often applied to the construction of new features. And Shapley theory is to study the contribution of current feature values to the model when the features and the model are known.

Thus, in addition to using interpretable analysis based on impurity reduction, we use the SHAP method to visualize the original features as well as the newly constructed features and explain the impact of the new features constructed based on the EF algorithm on the model performance by comparing the differences in Shapley values.

Validation experiments

Validation experiment of abrasive water jet polishing on 3D-printed CoCr alloy

CoCr alloy has been widely used as artificial implant owing to its excellent corrosion resistance and superior mechanical properties. For now, 3D printing has become more and more popular in the manufacturing of customized CoCr implants. However, the surface roughness after printing is very rough. Post-processing is always required before use. In this study, AWJP was used to polish the 3D-printed CoCr alloys, which has been widely used in polishing many other complicated surfaces made of different materials (Fähnle et al, 1998; Wang et al., 2022; Wang et al., 2017a, b). Before this study, a lot of trial and error had to be performed to obtain feasible polishing parameters that would allow the machined surface

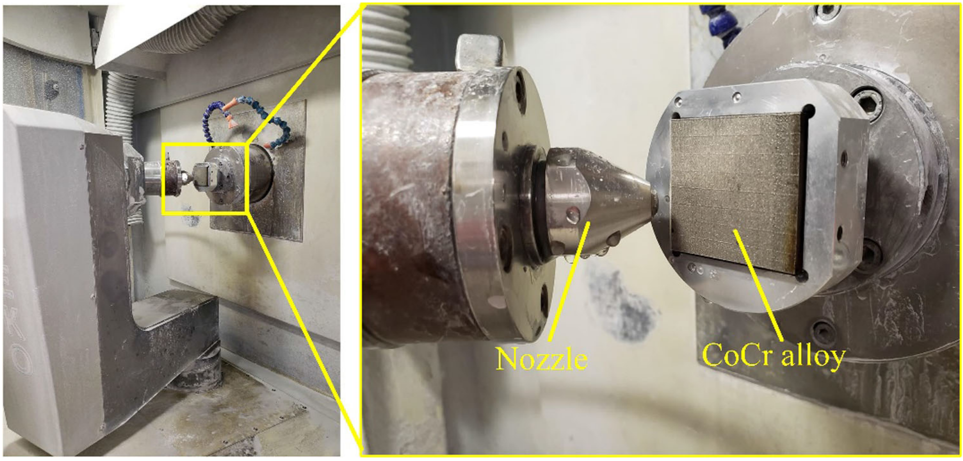


Fig. 5 Experimental set-up for abrasive water jet polishing

Table 2 Parameters setting of AWJP experiments

Polishing parameter	Range
Feed rate (f)	10,15,20,25,30,40,60,80 mm/min
Fluid pressure (P)	4,5,6,7,8,9,10 bar
Tool offset (TO)	4,6,8,10,12,14 mm
Step distance (d)	0.1,0.2,0.3,0.4,0.5,0.6 mm

roughness to meet specific requirements. Therefore, there is an urgent need to develop surface roughness prediction models to save time and cost for practical mass production.

The polishing equipment used in this study is ZEEKO IRP200 polishing machine as shown in Fig. 5. To explore the relationship between parameters of AWJP and surface roughness, this study conducted AWJP experiments based on 3D-printed components CoCr. The dimension of the sample was 42 mm × 40.5 mm × 10 mm, and the polishing region for each test was 6 × 3 mm. The polishing slurry used in this study was 1000# aluminum oxide (FUJIMI Inc., Japan) with the weight% of 10%. Sapphire nozzle with the diameter of 1 mm was used and the polishing slurry was impinged vertically to the target surface. Key polishing parameters such as feed rate, fluid pressure, stand-off distance, and step distance were considered and investigated in this study, while others were kept constant. After polishing, the surface roughness in surface mean height (Sa) was measured by ZYGO Nexview white light interferometer. Three different positions were measured randomly to obtain the final experimental results (Avg_Sa). 40 datasets were obtained under different polishing conditions including feed rate, pressure, tool offset, and step distance. The detailed parameters are shown in Table 2, and the experimental results are listed in Table 3.

Parameter setting

We randomly divided the above experimental data into training and test sets in appropriate proportions of 3:1 and normalized the data according to Eq. (1). The normalized training data are fed into the EF algorithm, and four folds are used for automatic feature construction to fit each decision tree model in the EF and the other is used as a validation set to verify the validity of the automatic feature construction. After that, we used the grid search method to tune the parameters based on the new training data to obtain the best hyperparameters for each model. Then, we performed regression prediction on the test dataset based on the best training models. Finally, the prediction results of each regression model were used as the input to the ensemble learning framework based on DE, which calculates the integrated prediction values by weighting the input according to the best weight assignment scheme obtained from the search. We randomly divided the data five times and repeated the above process, and took the average of the five prediction errors as the final prediction result. The error between the predicted and true values was used as the fitness calculation function of DE to guide the DE algorithm to search of a better solution, and we used common error evaluation metrics such as mean square error (MSE) and mean absolute error (MAE) to calculate the error, respectively, whose detailed calculation is given by Eqs. (8) and (9). And the units of both MAE and MSE are "μm".

In this paper, the maximum number of iterations of DE was 150,000 and the number of populations was 6. In addition, to avoid the fluctuating effect of the randomness of the data on the algorithm results, we repeated the experiments five times and took the average of the five experiments as the final experimental results.

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}|\widehat{y}_i - y_i| \tag{8}$$

Table 3 AWJP experiments data

No.	Feed rate (m/min)	Pressure (bar)	Tool offset (mm)	Step distance (mm)	Avg_Sa (um)
1	10	8	8	0.3	0.3362
2	20	8	8	0.3	0.5633
3	30	8	8	0.3	0.8086
4	40	8	8	0.3	1.6408
5	60	8	8	0.3	2.3625
6	80	8	8	0.3	2.2951
7	10	5	8	0.3	0.4358
8	10	6	8	0.3	0.6905
9	10	7	8	0.3	0.1513
10	10	8	8	0.3	0.2125
11	10	9	8	0.3	0.2211
12	10	10	8	0.3	0.2091
13	10	8	4	0.3	0.2547
14	10	8	6	0.3	0.2231
15	10	8	8	0.3	0.1800
16	10	8	10	0.3	0.2999
17	10	8	12	0.3	0.2727
18	10	8	14	0.3	0.6423
19	10	8	8	0.1	0.1034
20	10	8	8	0.2	0.1443
21	10	8	8	0.3	0.3773
22	10	8	8	0.4	0.2967
23	10	8	8	0.5	0.8901
24	10	8	8	0.6	1.4509
25	4	10	6	0.1	0.3798
26	4	15	8	0.2	1.6405
27	4	20	10	0.3	2.1245
28	4	25	12	0.4	2.5081
29	6	10	8	0.3	0.6506
30	6	15	6	0.4	1.7621
31	6	20	12	0.1	0.3846
32	6	25	10	0.2	1.1872
33	8	10	10	0.4	0.3705
34	8	15	12	0.3	0.3694
35	8	20	6	0.2	0.5596
36	8	25	8	0.1	0.6685
37	10	10	12	0.2	0.1065
38	10	15	10	0.1	0.0774
39	10	20	8	0.4	0.4767
40	10	25	6	0.3	0.5114

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left(\widehat{y_i} - y_i\right)^2 \tag{9}$$

Results and discussion

Performance comparison before and after feature construction based on the EF algorithm

As shown in Figs. 6 and 7, we compared the changes in model performance before and after applying the EF algorithm for feature construction (using MAE and MSE as a measure of model performance). The blue and red bars in the figure indicate the results of the five experiments of the ELDEA before and after feature construction, respectively. The dark blue dashed line indicates the average error of the five experiments of the ELDEA before feature construction, and the dark red dotted line indicates the average error of the five experiments of the ELDEA after feature construction, which visually demonstrates the effectiveness of feature construction using the EF algorithm. Specifically, its MAE decreases from 0.2425 to 0.2302, which is about 5.4%, and its MSE also decreases from 0.1611 to 0.1137, which is about 41.6%. In addition, the experimental results based on different data distributions demonstrate the higher robustness of the ELDEA method.

To further verify the effectiveness of the EF algorithm, we also compared the performance changes of each base regression model before and after applying the EF algorithm for feature construction, as shown in Fig. 8a and b. It can be found that applying the newly constructed features to most of the base regression models achieves lower error prediction results, which reveals the effectiveness of our proposed feature construction method for improving the accuracy of surface roughness prediction.

For the reason that the prediction performance of the method based on EF feature construction is better than that of the original method without feature construction, we had the following analysis. As an evolutionary algorithm, the EF algorithm performs automatic feature construction with the goal of minimizing the prediction error, i.e., it uses the prediction error as a fitness function to guide the direction of evolution. The algorithm retains the features that can reduce the prediction error and eliminates the features that are not helpful for prediction in the process of iterative evolution, so that the new features obtained in the end can effectively improve the prediction performance compared with the original features.

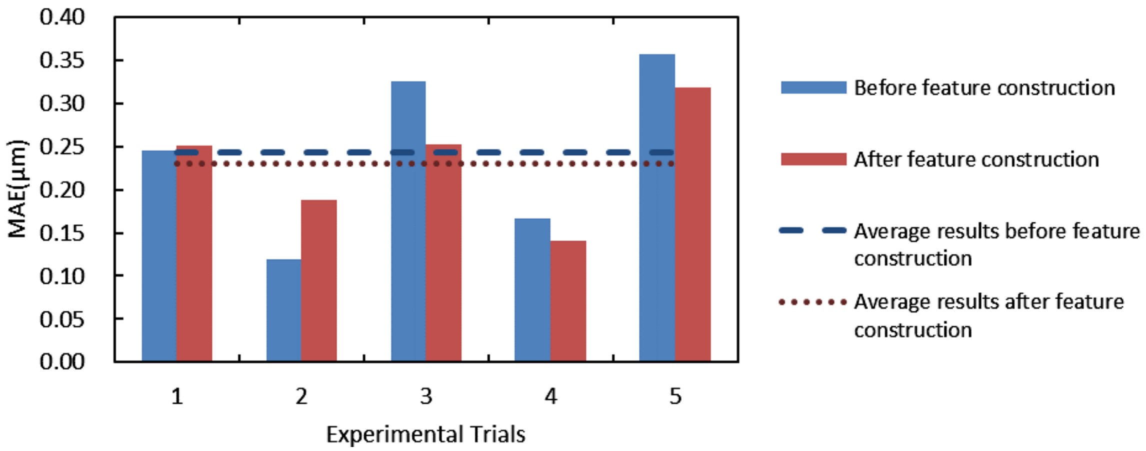


Fig. 6 Performance comparison between before and after feature construction (MAE)

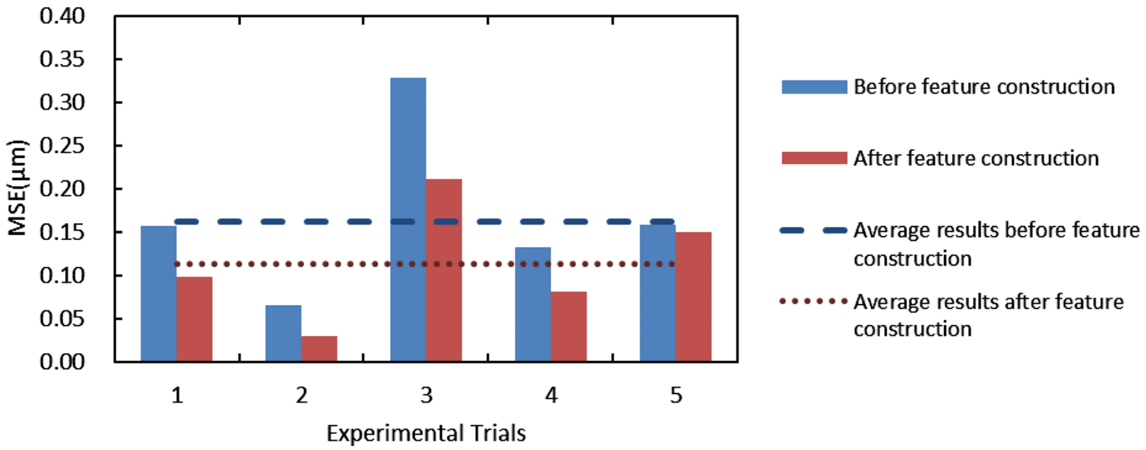


Fig. 7 Performance comparison between before and after feature construction (MSE)

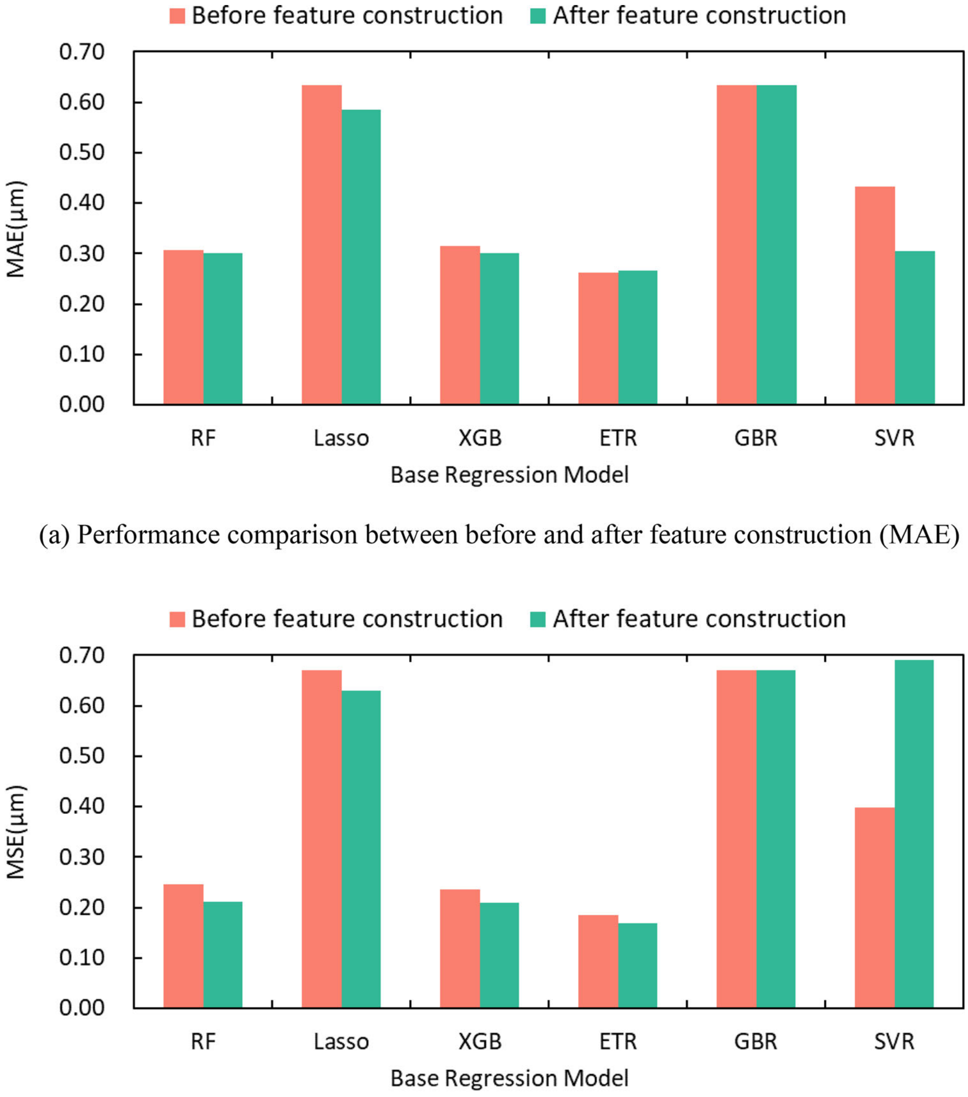
Comparison of ELDEA and classical machine learning

We compared the prediction performance of the ELDEA method with six classical regression algorithms on the AWJP experimental data and used MAE and MSE as evaluation metrics to measure the prediction performance of the methods to verify the effectiveness of our proposed ensemble method. As shown in Table 4, the ELDEA method has the best prediction results compared with the six classical regression methods, and its results are all better than those of the underlying regression model to varying degrees. Specifically, through the weight optimization of the DE algorithm, the mean values of MSE and MAE of the ELDEA compared to the six classical regression algorithms are reduced by 0.3164 and 0.1682, respectively. In addition, we can also find that the optimal values of MSE and MAE of the ELDEA compared to the six basic regression models are also reduced by about 49.0% and 15.7%. This shows that the integration effect of the ELDEA method is significant.

Using the MAE experiment on the AWJP dataset as an example, the ELDEA gives the best results, the ETR algorithm gives the second best results, and the GBR and LASSO algorithms give the worst results. The integration result of the ELDEA indicates that the optimal weights of the six regression models determined by the ELDEA, namely RF, LASSO, XGBoost, ETR, GBR, and SVR, are 0.0451, 0.0056, 0.1742, 0.5432, 0.0061, and 0.2257. The ELDEA finds that the ETR algorithm performed better on this experimental data, so the weights of the ETR base regression model were elevated during the weight assignment process. Relatively, the GBR and LASSO algorithms had the lowest final weight assignments due to their poor performance.

The possible reason for that the ETR model weight is high is: the features used by the ELDEA algorithm are constructed by the EF algorithm, which constructs features by integrating multiple decision trees, so the constructed features may be more suitable for integrated decision tree-based models such as ETR and RF. ETR is a decision-tree-based regression algorithm that uses multiple decision trees for prediction based on

Fig. 8 Performance comparison of each base regression algorithm before and after feature construction



(a) Performance comparison between before and after feature construction (MAE)

(b) Performance comparison between before and after feature construction (MSE)

randomly selected feature subsets. Compared to other decision tree-based algorithms, ETR uses more randomness in constructing decision trees to avoid overfitting. The algorithm is more stochastic. For a particular decision tree, the prediction results with it are often inaccurate because its best bifurcation attributes are chosen randomly, but multiple decision trees are combined to achieve good prediction results. And since the sample size of the current experimental dataset is inherently small, this randomness brings more possibilities for the results, which may also be the reason why ETR outperforms other algorithms.

The above results fully demonstrate the effectiveness of the Integration framework of the ELDEA, i.e., our algorithm is not only able to select the base regression model with better performance and give it a larger weight assignment ratio, but also to obtain a better prediction performance than the optimal base model through the integration framework.

Comparison of ELDEA and neural network algorithms

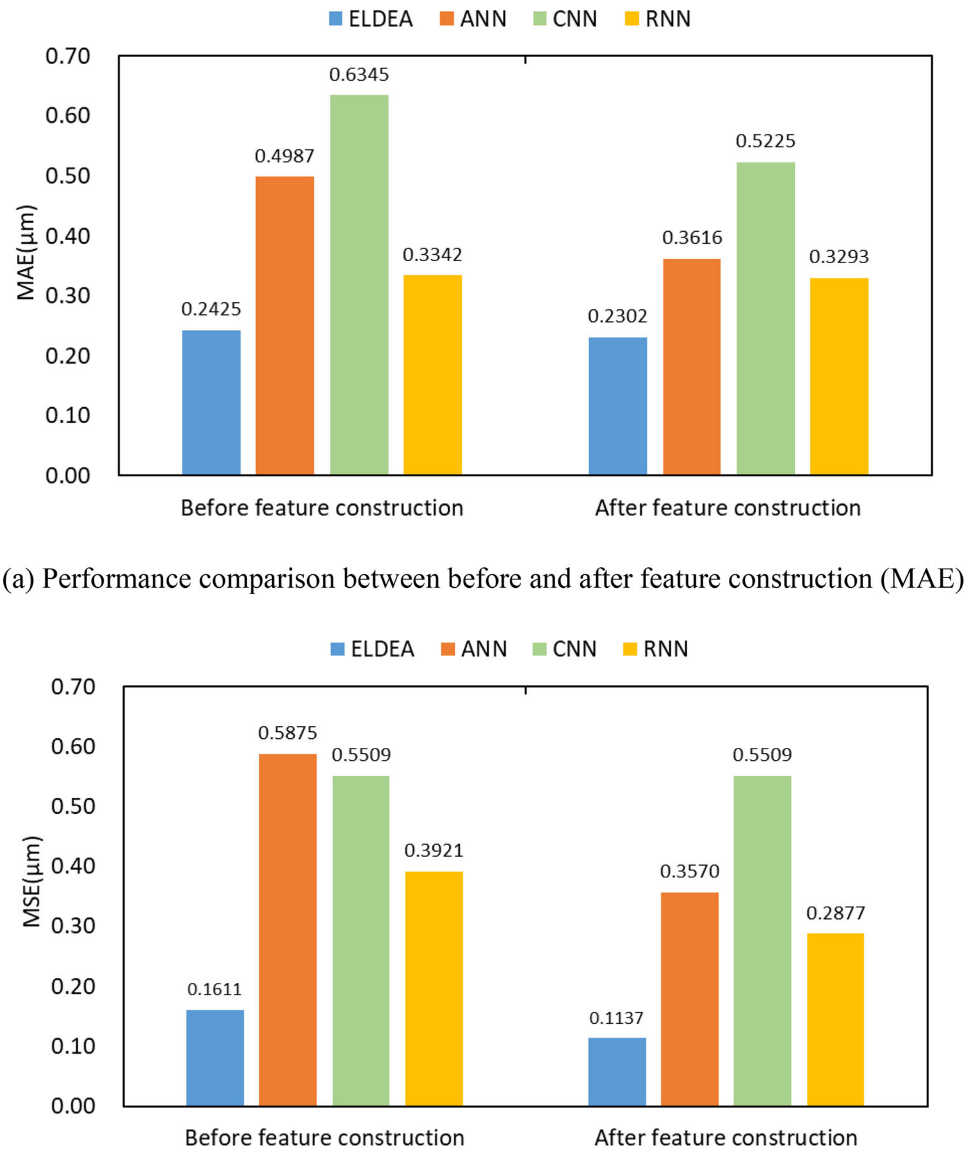
To further verify the performance of the proposed algorithm, we compared the prediction performance of the ELDEA method with three neural network algorithms, namely Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and Recurrent Neural Network (RNN) on the AWJP experimental data and still used MAE and MSE as evaluation metrics to measure the prediction performance of the methods, the results are shown in Fig. 9. For the neural network algorithms, experiments are conducted using the hold-out

Table 4 Comparison of MSE & MAE of different methods in AWJP. (Unit: μm)

	RF	LASSO	XGBoost	ETR	GBR	SVR	ELDEA
MSE	0.2111	0.6303	0.2093	0.1695	0.6704	0.6901	**0.1137**
MAE	0.3002	0.5852	0.3007	0.2664	0.6333	0.3042	**0.2302**

Bold values indicate the best results for the corresponding item

Fig. 9 erformance comparison with neural network algorithms before and after feature construction



(a) Performance comparison between before and after feature construction (MAE)

(b) Performance comparison between before and after feature construction (MSE)

method based on the raw data and the feature reconstruction data, respectively. Specifically, we divided the data into training and testing sets in a 3:1 ratio, and trained and predicted based on three neural network models. To ensure the fairness of the comparison, we still used different random seeds for five experiments, and took the average of the five experiments' results as the final prediction result. For ANN, the activation function uses ReLU, the optimization algorithm uses Adam, the regularization term coefficient is 0.01, and the maximum number of training iterations is 200. For CNN, the activation function uses ReLU, the optimization algorithm uses Adam, the loss function is absolute error, the network structure contains a fully connected layer, convolutional layer, and pooling layer, the window size of the convolutional layer is 8, and the epoch of training process is 200. For the RNN, the activation function is ReLU, the optimization algorithm is Adam, the loss function is absolute error, the network structure contains a fully connected layer, a GRU layer, and a pooling layer, and the units of the GRU layer are set to 16, and the epoch of the training process is 200.

From the results in Fig. 9a, it can be found that the MAE of the ELDEA algorithm is the smallest on both the data before and after feature construction compared to the other three neural network algorithms, thus it can be shown that the ensemble model can obtain better performance than the neural network algorithm on this type of dataset.

In addition, to further verify the performance advantage of the proposed algorithm, we also conducted a comparative experiment by using MSE as an evaluation metric. The results are shown in Fig. 9b, and it can be found that the performance of the proposed algorithm is still significantly due to the neural network algorithm. We analyzed the possible reasons for this phenomenon as follows: due to the small amount of current experimental data, without the support of a large amount of training data, the neural network algorithms are prone to overfitting, and thus their prediction performance is not necessarily better than that of the ensemble learning methods.

Comparison of ELDEA and other algorithms

Similar to our proposed algorithm, the ELGA algorithm (Wang et al., 2022) is an ensemble framework based on evolutionary algorithms for surface roughness prediction, so we compared the performance differences between ELDEA and ELGA algorithms on the MJP experimental dataset of 3D-printed 316 L stainless steel. In the MJP experiments, 3D-printed 316 L stainless steel parts were polished by a ZEEKO IRP200 machine with different process parameters. After polishing, the surface roughness Sa was measured three times and averaged to obtain the final Sa value. We obtained 43 experimental data items with different polishing parameters, including feed rate (f), fluid pressure (P), tool offset (TO), step distance (d), and surface roughness (Sa$_{\mathrm{f}}$). The detailed experimental data are shown in Table 5.

We referred to the experimental setup of Wang et al. (2022a, b) in their paper and divided the dataset randomly into training and testing sets in the ratio of 9:1, and used the ratio of MAE as the evaluation index, as shown in Eq. (10). Finally, ten experiments were run and averaged as the final result. Smaller ratio of MAE indicates higher prediction accuracy of the prediction method. In addition, we also compared with other two ensemble learning algorithms, namely Averaging and Stacking algorithms, and the results are shown in Table 6. It can be found that the prediction effect of the ELDEA

Table 5 MJP experiment data

	f (mm/min)	P (bar)	TO (mm)	d (mm)	Sa$_{\mathrm{f}}$ (nm)
1	10	8	5	0.2	21
2	20	8	5	0.2	30
3	30	8	5	0.2	32
4	40	8	5	0.2	88
5	60	8	5	0.2	127
6	80	8	5	0.2	144
7	20	4	5	0.2	88
8	20	5	5	0.2	85
9	20	6	5	0.2	83
10	20	7	5	0.2	39
11	20	8	5	0.2	39
12	20	9	5	0.2	37
13	20	10	5	0.2	33
14	20	8	2.5	0.2	27
15	20	8	5	0.2	40
16	20	8	7.5	0.2	32
17	20	8	10	0.2	72
18	20	8	12.5	0.2	26
19	20	8	15	0.2	36
20	20	8	5	0.1	31
21	20	8	5	0.2	43
22	20	8	5	0.3	47
23	20	8	5	0.4	85
24	20	8	5	0.5	98
25	20	8	5	0.6	138
26	20	8	5	0.7	99
27	20	8	5	0.8	131
28	10	5	2.5	0.2	44
29	15	5	5	0.4	146
30	20	5	7.5	0.6	183
31	25	5	10	0.8	229
32	10	6	5	0.6	76
33	15	6	2.5	0.8	151
34	20	6	10	0.2	65
35	25	6	7.5	0.4	165
36	10	7	7.5	0.8	121
37	15	7	10	0.6	111
38	20	7	2.5	0.4	86
39	25	7	5	0.2	66
40	10	8	10	0.4	61
41	15	8	7.5	0.2	30
42	20	8	5	0.8	133
43	25	8	2.5	0.6	144

Table 6 Comparison of the ratio of MAE among different ensemble strategies

Ensemble Strategy	The ratio of MAE
Stacking (Breiman, 1996)	0.261
Averaging (Liu & Kuo, 2016)	0.252
ELGA(Wang., 2022)	0.195
ELDEA(our method)	**0.164**

Bold value indicates the best results for the corresponding item

method is significantly better than other three ensemble methods.

$$\text{The ratio of MAE} = \frac{1}{m}\sum_{i=1}^{m}(|\widehat{y}_i - y_i|/\widehat{y}_i) \tag{10}$$

Time complexity analysis of ELDEA

In this section, we examined the time complexity of ELDEA starting from three parts: the EF-based automatic feature construction module, the multi-algorithm regression module, and the DE-based ensemble module, since they are the most time-consuming parts of the whole algorithm. For the EF-based automatic feature construction module, we assumed that the time complexity of the model training process is $O(1)$, and then the overall time complexity of this module is $O(G_1|P_1|)$, where G_1 and $|P_1|$ represents the maximum number of iterations and the number of populations of the EF algorithm, respectively. For the multi-algorithm regression module, hyperparameter tuning is the most time-consuming part of the module. We assumed that the time complexity of the hyperparameter tuning process of a model is $O(m)$, then the overall time complexity of the module is $n*O(m)$, and n is the number of base models, which is a constant. Therefore, the time complexity of the module can be simplified to $O(m)$. For the DE-based ensemble module, similar to the EF algorithm, as an evolutionary algorithm, the time complexity of the module is $O(G_2|P_2|)$, where G_2 and $|P_2|$ represent the maximum number of iterations and the number of populations of the DE algorithm, respectively. In summary, the overall time complexity of the ELDEA algorithm is determined by $\max[O(G_1|P_1|), O(m), O(G_2|P_2|)]$.

All methods were implemented in Python and the program was run on a computer equipped with an i9-9750 H 2.60 GHz CPU and an NVIDIA GEFORCE RTX 3060 GPU. Table 7 shows the single run time and five average run times of the ELDEA algorithm. (Unit: seconds)

Discussion of interpretability analysis results

Discussion of interpretability analysis based on impurity reduction concept

In this section, we ranked the constructed features based on the concept of impurity reduction and conduct interpretable analysis for exploring the feature importance. The importance ranking of the constructed features is shown in Fig. 10, containing the top 15 most important features among the newly constructed features, where X_0, X_1, X_2, and X_3 represent feed rate, fluid Pressure, tool offset, and step distance, respectively. It can be found that 13 of the first 15 important features are related to feed rate, which indicates that the machining parameter of feed rate has a significant effect on the surface roughness. We also find that only a very small number of the newly constructed features are related to tool offset, which indicates that this machining parameter is not the key factor for predicting surface roughness. Figure 10 shows that the most important feature is the product of feed rate and fluid pressure. It can be speculated that the joint action of the two parameters may have a greater effect on the surface roughness, and in the subsequent machining experiments, we can design more interactive experiments based on the two machining parameters for verification. Comparing the third important feature and the fourth one, we find that the main difference is that the fourth important feature adds the role of the product of feed rate parameter based on the third feature, and the former feature is more important than the latter. That indicates that, after adding feed date, the combined effect of the three processing parameters has a more significant effect on the surface roughness. This result demonstrates the importance of the processing parameter Feed Rate. In addition, the sum of two machining parameters, fluid pressure, and step distance, is noteworthy as an important one. In future, we can explore the interaction of these two machining parameters in conjunction with related domain knowledge for exploring new insights for advancing the polishing technologies.

Discussion of interpretability analysis based on Shapley theory

In this section, we used the SHAP method based on Shapley theory for further analyzing the original features in an interpretable manner. For each feature, we selected the average of the absolute values of its Shapley values before and after feature construction as the feature importance measure, which is presented in the form of a bar chart in Fig. 11a and b. Figure 11a represent the four features extracted from the original data, which are four main processing parameters. Figure 11b represent the top five most important newly constructed features in order of feature importance. Comparing

Table 7 Running time of ELDEA algorithm

No. of experiments	1	2	3	4	5	Average
Running time (second)	≈253	≈249	≈221	≈210	≈267	≈240

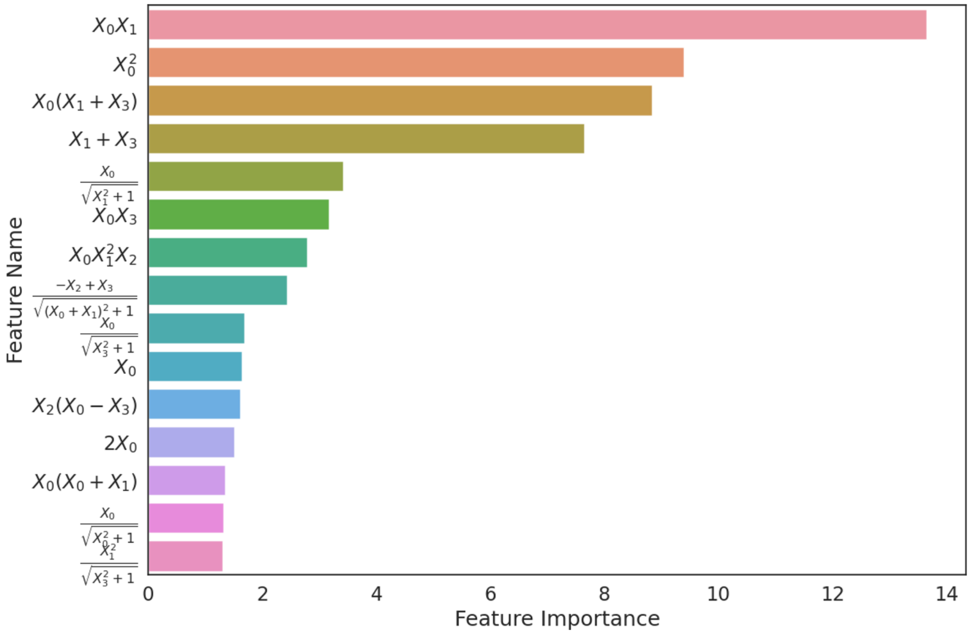
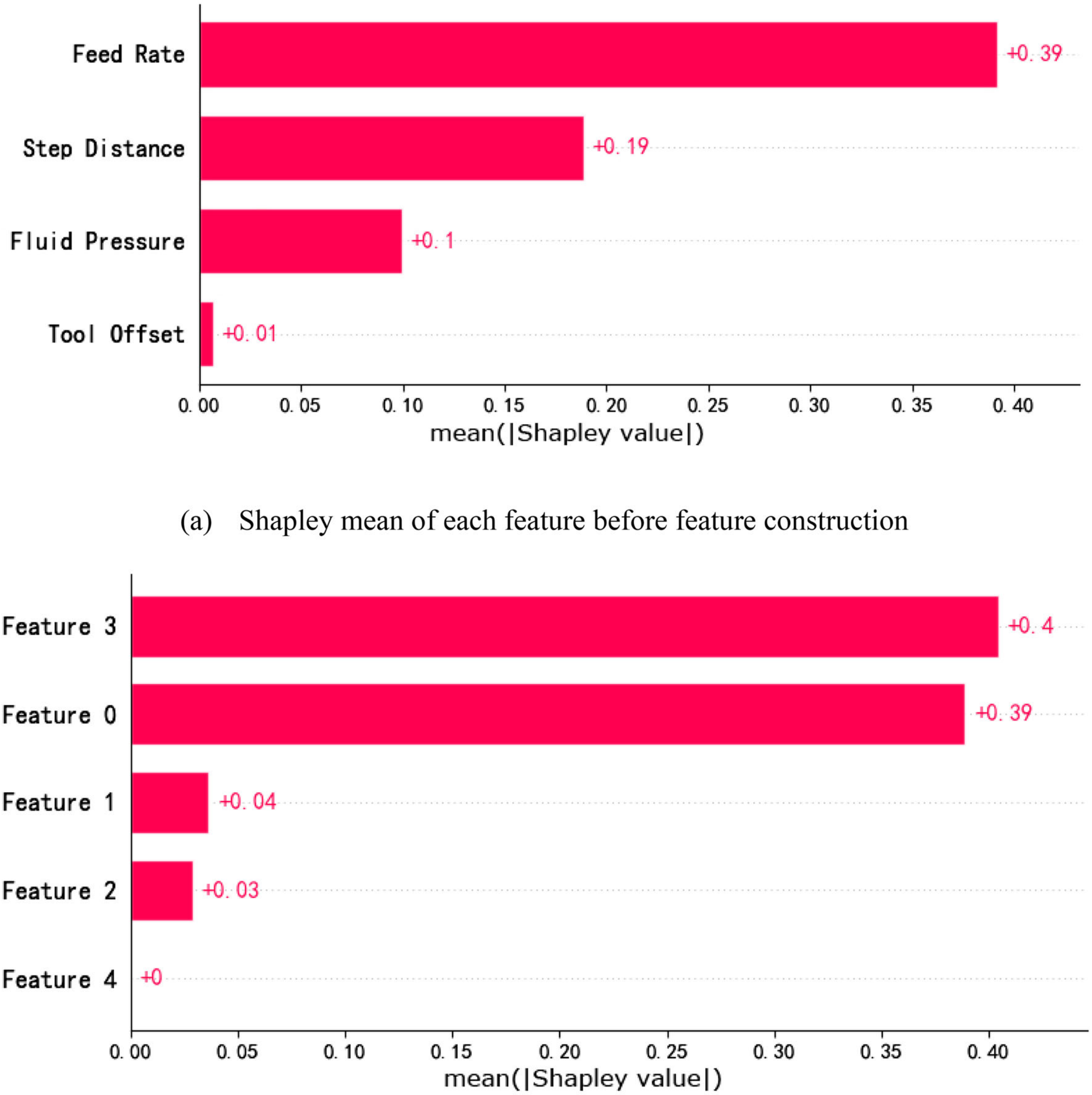


Fig. 10 Feature importance ranking based on impurity reduction concept

subplots (a) and (b), it can be found that more features with higher Shapley values are constructed by the EF algorithm. Compared with the original features, the newly constructed features have better diversity and are more robust, therefore improving the prediction performance. For subplot (a) in Fig. 11, we found that the feature importance ranking obtained based on the SHAP method is largely consistent with the results exhibited in Fig. 10, which further illustrates the reliability of both interpretation methods. In addition, slightly different from the results in Fig. 10, the results in subplot (b) of Fig. 11 indicate that some of the newly constructed features do not contribute to the model based on the calculation method of Shapley's theory. Different methods of calculating feature importance may have significantly different results for the same feature, and we need to explore both methods further and combine the nature of the features themselves to weigh which features play a major role in modeling.

For each sample, we plotted the Shapley values for each of its features, as shown in Fig. 12. The plot helps us to better understand the overall pattern from a global view and possibly identify predicted outliers. The figure contains all the samples. Each point represents a sample. The horizontal coordinates are the samples ranked by Shapley value. The vertical coordinates are the features ranked by importance, and the depth of the color indicates the magnitude of the feature value (high in red, low in blue). It is worth noting that the feature value here is different from the Shapley value and is the value taken by the feature itself.

From Fig. 12, it can be found that the feed rate is the most important. As the feed rate value increases, the corresponding Shapley value also increases (i.e., the color of the sample points gradually changes from blue to red as the x-axis moves). When the feed rate is too low, the contribution of this feature to the prediction model of surface roughness is reduced, and even has a negative effect on the prediction of surface roughness. Therefore, in the subsequent experiments, we can set a minimum threshold value for feed rate so that this processing parameter under condition that it is not lower than this threshold value. This method can help us reduce the number of experiments and thus improve experimental efficiency. In addition, fluid pressure and step distance are also the main influencing factors, and tool offset has little effect on the surface roughness, which is consistent with the analysis results in the previous section, thus further determining the main and secondary influencing factors of surface roughness. The graph also shows that Fluid Pressure and step distance are positively correlated with the prediction of surface roughness in a certain range, i.e. the larger the value the more significant the effect on surface roughness. In addition, we can also find some outliers in the plot, which can help us to detect outliers. We will analyze and explore these outliers in the subsequent study.



(a) Shapley mean of each feature before feature construction

(b) Shapley mean of each feature after feature construction

Fig. 11 Comparison of Shapley values of each feature before and after feature construction

Conclusions and future work

In this paper, a novel surface roughness prediction method based on DE algorithm and ensemble learning is proposed and applied to surface roughness prediction for abrasive water jet polishing (AWJP). Our method uses six machine learning models as the base models that form the ensemble framework, namely RF, LASSO, XGBoost, ETR, GBR, and SVR, which are responsible for modeling the relationship between machining parameters and surface roughness. First, a feature reconstruction method based on the EF algorithm is used to reconstruct the features of the experimental data, and this data is used for the training and prediction of the base regression models. Then, the DE algorithm is used to optimize the weight assignment of each model. For this optimization problem, a simplified encoding mechanism is proposed in this paper for the individual design of the DE algorithm, which can further improve the search efficiency of the DE algorithm. We validated the proposed method on the AWJP dataset and compare the performance of the method with other methods. The results showed that the ELDEA has significant advantages over classical machine learning algorithms and other ensemble learning algorithms.

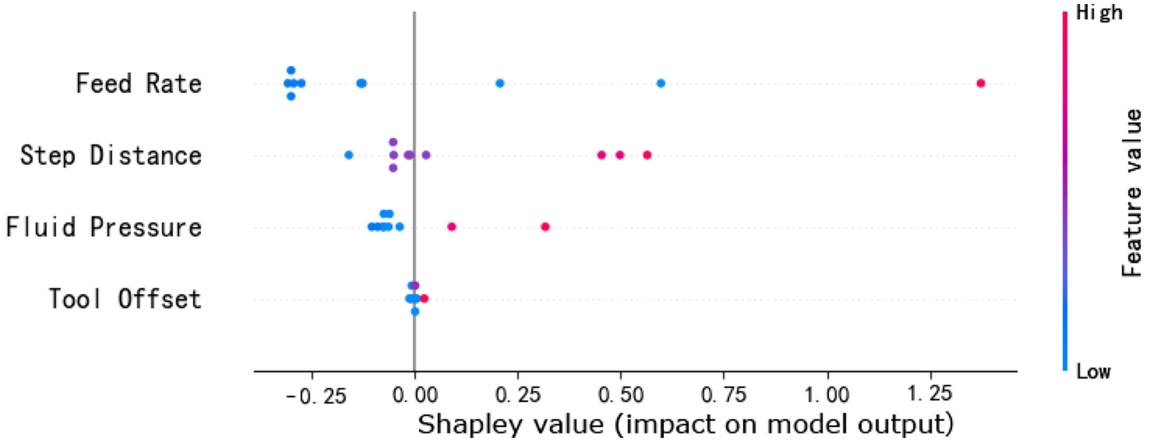


Fig. 12 Distribution of Shapley values for each sample

On the experimental dataset of AWJP, the ELDEA algorithm reduces its MSE and MAE by about 33–84% and 14–64%, respectively, compared to the classical machine learning algorithms. For the MJP experiments, the ELDEA approach reduces the ratio of MAE by about 37%, 35%, and 16%, respectively, compared to those obtained by other three ensemble learning methods. Finally, the data features are analyzed and discussed using an interpretable approach to identify the main processing parameters that affect surface roughness, and the results of the interpretable analysis provide a theoretical reference for future experiments and research.

In this paper, the ELDEA can receive different inputs through an ensemble model and is expected to be transferable to surface roughness prediction for other machining methods such as turning and milling. In addition, the interpretable methods used in the current study all perform interpretable analysis from the level of features, by combining data and models. The research on the self-explanation of the model itself is yet to be further explored. In the future, we will aim to combine the interpretability study of the model with the interpretability study of the features and use a more comprehensive interpretability approach for analysis and discussion. Therefore, future work will be devoted to investigating and improving the ELDEA method so that it is expected to become a general method for surface roughness prediction with more comprehensive interpretability.

Acknowledgements The work described in this paper was mainly supported by the Natural Science Foundation of Fujian Province of China (Project No: 2020J01697) and Research Grants Council (Project No. 15200119) of the Government of the Hong Kong Special Administrative Region, China. The research was partially supported by the National Natural Science Foundation of China (Project No. 52105534). The work was also supported by the Research and Innovation Office of the Hong Kong Polytechnic University (Project No: BBR8 and BD4A). The work was also supported by Shenzhen-Hong Kong-Macau Technology Research Programme (Project No: SGDX20201103095203029 and SGDX20220530110804030).

Data availability The authors confirm that the data supporting the findings of this study are available within the article.

References

Ahmad, M. F., Isa, N. A. M., Lim, W. H., & Ang, K. M. (2021). Differential evolution: A recent review based on state-of-the-art works. *Alexandria Engineering Journal*, *61*(5), 3831–3872. https://doi.org/10.1016/j.aej.2021.09.013.

Awad, M., & Khanna, R. (2015). Support vector regression in efficient learning machines. *Apress Berkeley CA*. https://doi.org/10.1007/978-1-4302-5990-9_4

Benardos, P., & Vosniakos, G. C. (2003). Predicting surface roughness in machining: A review. *International Journal of Machine Tools and Manufacture, 43*(8), 833–844. https://doi.org/10.1016/S0890-6955(03)00059-2

Breiman, L. (1996). Stacked regressions. *Machine Learning, 24*(1), 49–64. https://doi.org/10.1007/BF00117832

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system*. Paper presented at the Proceedings of the 22nd acm Sigkdd International Conference on Knowledge Discovery and Data Mining. https://doi.org/10.1145/2939672.2939785.

Chen, Z., Shi, Y., & Lin, X. (2018). Evaluation and improvement of material removal rate with good surface quality in TC4 blisk blade polishing process. *Journal of Advanced Mechanical Design Systems and Manufacturing, 12*(4), JAMDSM0083-JAMDSM. https://doi.org/10.1299/jamdsm.2018jamdsm0083

Chen, Z., Shi, Y., Lin, X., Yu, T., Zhao, P., Kang, C., He, X., & Li, H. (2019). Analysis and optimization of process parameter intervals for surface quality in polishing Ti-6Al-4V blisk blade. *Results in Physics*, *12*, 870–877. https://doi.org/10.1016/j.rinp.2018.12.056.

Deng, J., Zhang, Q., Lu, J., Yan, Q., Pan, J., & Chen, R. (2021). Prediction of the surface roughness and material removal rate in chemical mechanical polishing of single-crystal SiC via a back-propagation neural network. *Precision Engineering*, *72*, 102–110. https://doi.org/10.1016/j.precisioneng.2021.04.012.

Fähnle, O. W., Van Brug, H., & Frankena, H. J. (1998). Fluid jet polishing of optical surfaces. *Applied optics*, *37*(28), 6771–6773. https://doi.org/10.1364/AO.37.006771.

Fan, J., Ren, X., Pan, R., Wang, P., & Tao, H. (2022). Model and optimize the magnetic composite fluid (MCF) polishing process with machine learning modeling and Intelligent optimization algorithm.

International Journal of Precision Engineering and Manufacturing. https://doi.org/10.1007/s12541-022-00668-5

Fan, J., Ren, X., Pan, R., Wang, P., & Tao, H. (2022). Optimization of finishing parameters for magnetic compound fluid finishing (MCFF) of copper alloy. *The International Journal of Advanced Manufacturing Technology*. https://doi.org/10.21203/rs.3.rs-989241/v1

Freedman, D. A. (2009). *Statistical models: Theory and practice*. Cambridge University Press. https://doi.org/10.1198/tech.2006.s403

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics*. https://doi.org/10.2307/2699986

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1.

Hilt, D. E., & Seegrist, D. W. (1977). *Ridge, a computer program for calculating ridge regression estimates* (236 vol.). Department of Agriculture, Forest Service, Northeastern Forest Experiment Station.

Huai, W., Tang, H., Shi, Y., & Lin, X. (2017). Prediction of surface roughness ratio of polishing blade of abrasive cloth wheel and optimization of processing parameters. *The International Journal of Advanced Manufacturing Technology*, *90*(1), 699–708. https://doi.org/10.1007/s00170-016-9397-3.

Huang, X., Wang, Z., Shen, B., Lei, P., & Fu, Z. (2022). Research on optimization method of process parameters for SiC components robotic bonnet polishing. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*. https://doi.org/10.1177/09544062221104024

Ibrahim, A. M., & Tawhid, M. A. (2022). An improved artificial algae algorithm integrated with differential evolution for job-shop scheduling problem. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-021-01888-8

Jian, Y., Ri, P., Fan, J., Wang, Z., Zhang, D., & Ji, S. (2022). Optimization of magnetic composite fluid polishing process based on response surface method. *Journal of the Chinese Institute of Engineers*, *45*(1), 35–41. https://doi.org/10.1080/02533839.2021.1983460.

Khalick Mohammad, A. E., Hong, J., & Wang, D. (2017). Polishing of uneven surfaces using industrial robots based on neural network and genetic algorithm. *The International Journal of Advanced Manufacturing Technology*, *93*(1), 1463–1471. https://doi.org/10.1007/s00170-017-0524-6.

Li, Y., Wang, S., Tian, Q., & Ding, X. (2015). Feature representation for statistical-learning-based object detection: A review. *Pattern Recognition*, *48*(11), 3542–3559. https://doi.org/10.1016/j.patcog.2015.04.018.

Liu, C. A., & Kuo, B. S. (2016). Model averaging in predictive regressions. *The Econometrics Journal*, *19*(2), 203–231. https://doi.org/10.1111/ectj.12063.

Lu, C. (2008). Study on prediction of surface quality in machining process. *Journal of Materials Processing Technology, 205*(1–3), 439–450. https://doi.org/10.1016/j.jmatprotec.2007.11.270

Nguyen, D. N., Dao, T. P., Prakash, C., Singh, S., Pramanik, A., Krolczyk, G., & Pruncu, C. I. (2020). Machining parameter optimization in shear thickening polishing of gear surfaces. *Journal of Materials Research and Technology*, *9*(3), 5112–5126. https://doi.org/10.1016/j.jmrt.2020.03.028.

Qi, J., Zhang, D., Li, S., & Chen, B. (2018). Modeling and prediction of surface roughness in belt polishing based on artificial neural network. *Proceedings of the Institution of Mechanical Engineers Part B: Journal of Engineering Manufacture*, *232*(12), 2154–2163. https://doi.org/10.1177/0954405416683737.

Schneckenburger, M., Garcia-Barth, L., & Börret, R. (2020). Machine learning model for robot polishing cell. *Paper presented at the Seventh European Seminar on Precision Optics Manufacturing*. https://doi.org/10.1117/12.2564633

Shalev-Shwartz, S., Singer, Y., Srebro, N., & Cotter, A. (2011). Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, *127*(1), 3–30. https://doi.org/10.1007/s10107-010-0420-4.

Shapley, L. (1953). Quota solutions op n-person games. *Edited by Emil Artin and Marston Morse*, 343.

Solheid, J., Elkaseer, A., Wunsch, T., Charles, A., Seifert, H., & Pfleging, W. (2020). Effect of process parameters on surface texture generated by laser polishing of additively manufactured Ti-6Al-4V. Paper presented at the Laser-based micro-and nanoprocessing XIV. https://doi.org/10.1117/12.2545623.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, *58*(1), 267–288. https://doi.org/10.2307/2346178.

Tong, X., Wu, X., Zhang, F., Ma, G., Zhang, Y., Wen, B., & Tian, Y. (2019). Mechanism and parameter optimization in grinding and polishing of M300 steel by an elastic abrasive. *Materials*, *12*(3), 340. https://doi.org/10.3390/ma12030340.

Tran, B., Xue, B., & Zhang, M. (2019). Genetic programming for multiple-feature construction on high-dimensional classification. *Pattern Recognition*, *93*, 404–417. https://doi.org/10.1016/j.patcog.2019.05.006.

Virgolin, M., Alderliesten, T., Bel, A., Witteveen, C., & Bosman, P. A. (2018). *Symbolic regression and feature construction with GP-GOMEA applied to radiotherapy dose reconstruction of childhood cancer survivors*. Paper presented at the Proceedings of the Genetic and Evolutionary Computation Conference. https://doi.org/10.1145/3205455.3205604.

Wang, G., Wang, Y., Zhao, J., & Chen, G. (2012). Process optimization of the serial-parallel hybrid polishing machine tool based on artificial neural network and genetic algorithm. *Journal of Intelligent manufacturing*, *23*(3), 365–374. https://doi.org/10.1007/s10845-009-0376-5.

Wang, C., Cheung, C. F., Ho, L., Liu, M., & Lee, W. B. (2017a). A novel multi-jet polishing process and tool for high-efficiency polishing. International journal of machine tools and manufacture, 115, 60–73. https://doi.org/10.1016/j.ijmachtools.2016.12.006.

Wang, C., Cheung, C. F., & Liu, M. (2017b). Numerical modeling and experimentation of three dimensional material removal characteristics in fluid jet polishing. International Journal of Mechanical Sciences, 133, 568–577. https://doi.org/10.1016/j.ijmecsci.2017b.09.018.

Wang, Z., Lin, X., Shi, Y., Zhang, Y., & Chen, Z. (2020). Reducing roughness of freeform surface through tool orientation optimization in multi-axis polishing of blisk. *The International Journal of Advanced Manufacturing Technology*, *108*(3), 917–929. https://doi.org/10.1007/s00170-020-05433-4.

Wang, C., Zhang, Z., Cheung, C. F., Luo, W., Loh, Y. M., Lu, Y., Kong, L., & Wang, S. (2022a). Maskless fluid jet polishing of optical structured surfaces. Precision Engineering, 73, 270–283. https://doi.org/10.1016/j.precisioneng.2021.09.010.

Wang, R., Cheng, M. N., Loh, Y. M., Wang, C., & Cheung, C. F. (2022b). Ensemble learning with a genetic algorithm for surface roughness prediction in multi-jet polishing. Expert Systems with Applications, 207, 118024. https://doi.org/10.1016/j.eswa.2022b.118024 .

Wei, Y., & Wu, D. (2022). Material removal rate prediction in chemical mechanical planarization with conditional probabilistic autoencoder and stacking ensemble learning. *Journal of Intelligent Manufacturing*. https://doi.org/10.1007/s10845-022-02040-w

Xie, Y., Chang, G., Yang, J., Zhao, M., & Li, J. (2022). Process optimization of robotic polishing for Mold Steel based on response surface method. *Machines*, *10*(4), 283. https://doi.org/10.3390/machines10040283.

Xu, W., Wei, Z., Sun, J., Wei, L., & Yu, Z. (2012). Surface quality prediction and processing parameter determination in electrochemical mechanical polishing of bearing rollers. *The International Journal of Advanced Manufacturing Technology*, *63*(1), 129–136. https://doi.org/10.1007/s00170-011-3891-4.

Xu, J., Zou, P., Wang, W., & Kang, D. (2021). Study on the mechanism of surface topography evolution in melting and transition regimes of laser polishing. *Optics & Laser Technology*, *139*, 106947. https://doi.org/10.1016/j.optlastec.2021.106947.

Yuan, S., Li, T., & Wang, B. (2021). A discrete differential evolution algorithm for flow shop group scheduling problem with sequence-dependent setup and transportation times. *Journal of Intelligent Manufacturing, 32*, 427–439. https://doi.org/10.1007/s10845-020-01580-3

Zhang, H., Zhou, A., & Zhang, H. (2021). An Evolutionary Forest for Regression. *IEEE Transactions on Evolutionary Computation*. https://doi.org/10.1109/TEVC.2021.3136667.

Zhong, Z. (2021). Surface roughness of machined wood and advanced engineering materials and its prediction: A review. *Advances in Mechanical Engineering*, *13*(5), 16878140211017632. https://doi.org/10.1177/16878140211017632.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67*(2), 301–320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

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