



---

*Research article*

## Machine learning-enabled multi-objective optimization design of process parameters for a medium-entropy alloy

Lihong Zhao, Yuyang Zhu and Cun Chen\*

School of Mathematics and Statistics, Zhengzhou University, Zhengzhou 450001, China

\* **Correspondence:** Email: [chencun@zzu.edu.cn](mailto:chencun@zzu.edu.cn).

**Abstract:** In this article, a machine learning strategy is designed to improve both the tensile strength and the elongation of a medium-entropy alloy using multi-objective optimization. In order to overcome the problem of small data sets, the linear interpolation method and Gaussian mixture model (GMM) sample generation method are integrated to augment the data. Based on the enhanced data set, the machine learning model is optimized by the sparrow search algorithm (SSA), and the multi-objective particle swarm optimization (MOPSO) algorithm is used to search the alloy process parameters with high tensile strength and elongation. Finally, 15 process parameter combinations superior to existing alloys are presented, among which three combinations, as predicted by gradient boosting regression (GBR), achieve a tensile strength exceeding 2000 MPa and an elongation of about 11%, significantly outperforming existing alloys.

**Keywords:** alloy design; machine learning; virtual data generation; performance prediction; multi-objective optimization; interpretability analysis

---

### 1. Introduction

A medium-entropy alloy (MEA) exhibits good corrosion resistance, wear resistance, and high strength. 3J22G is designed as a new medium-entropy alloy, which contains Co (26.01%), Cr (26.92%), Ni (23.03%), and Fe (15.76%), along with small amounts of Mo (3.48%), W (3.72%), and C (0.108%). The process usually includes raw material selection, alloy melting, heat treatment, and forming steps. In the melting process, the aging temperature, time, and melting temperature are strictly controlled to ensure the microstructure and property stability of the obtained alloy. The main characteristics of an

MEA focus on its excellent magnetic properties, thermal stability, and mechanical properties [1–3]. In aerospace, automotive manufacturing, the oil and gas industry, and medical devices, there is a growing demand for high strength and ductility [4,5]. The excellent properties of an MEA make it an ideal material choice in these fields. However, although an MEA has a wide range of application potential in many fields, its properties and preparation process still need further research and optimization [6]. There is a challenging problem: the mutual constraint between tensile strength and elongation [7,8]. In general, increasing the tensile strength of the alloy tends to lead to a decrease in its elongation, while increasing the elongation may reduce the tensile strength. Solving this problem is of great significance for the further application of alloys. Considering the huge parameter space, a tough scientific question is how to design appropriate process parameters to improve the two mutual restricted properties. At present, to achieve the improvement of the comprehensive properties of the material, the traditional method mainly designs process parameters by trial and error [9–13], which is time and resource consuming. Meanwhile, the trial-and-error method cannot accurately achieve the multi-objective optimization problem. In order to solve this problem, data-driven machine learning (ML) methods are introduced to optimize process parameters.

In the field of materials science, ML is an effective approach to explore the complex nonlinear relationship between alloy components, processing processes, and properties based on experimental data. Several researchers have applied ML to the material design for target properties. Wang et al. [14] proposed a high-performance high/medium entropy alloys design strategy that combines ML with a precipitation strengthening mechanisms. They synthesized the alloy Al<sub>5</sub>Ti<sub>8</sub>Co<sub>40</sub>Fe<sub>7</sub>Ni<sub>40</sub> with a yield strength close to 1.15 GPa. In 2021, Liu et al. [15] predicted the alloy hardness using ML and the CALPHAD auxiliary experiment for the system of Ti-Zr-Nb-Ta high-entropy alloys (HEAs). Zhu et al. [16] used neural networks and first-principles calculations to reveal the relationship between hardness and element descriptors in multi-component alloys, and predicted the hardness of Al-Cr-Fe-Ni systems. In 2022, Wang et al. [17] developed an ML alloy design system aimed at efficiently designing alloys with components to increase hardness. They used the SVR-RBF regression model, which achieved a prediction accuracy of 0.94. In 2023, Zhang et al. [18] used a non-dominated sorting genetic ML algorithm to optimize the chemical composition for both strength and corrosion resistance. A series of good performance Al-Mg-Si alloys were developed with excellent mechanical properties. In addition, ML has been widely used in high-entropy alloy phase prediction [19–21], accelerating the discovery of new materials and performance optimization.

In recent years, the strategy of combining ML with optimization has also been used to accelerate the search for materials with a single target property [22–26]. However, with increasing application requirements, it is urgent to explore materials with superior performances in various properties. In 2022, Li et al. [27] combined forward and reverse models to design the composition and heat treatment parameters of Reduced Activation Ferritic/Martensitic (RAFM) steel with the desired tensile property and a comparable elongation. In 2023, Xie et al. [28] proposed an efficient alloy material design method based on data transfer learning for a rapid design of new alloys, which can significantly improve the strength and plasticity of E2 alloys. Xie et al. [29] used a strategy combining Bayesian optimization ML with experimental iteration to rapidly design the secondary deformation-aging parameters of a novel copper alloy, while improving both the mechanical and electrical properties.

At present, 3J22G is a newly developed material whose two properties cannot meet the application requirements at the same time, especially its poor toughness. At present, there is no data-driven method for multi-objective parameter optimization design research. Moreover, considering the

small datasets, training an ML model to achieve multi-objective optimization of is difficult. In this paper, a process parameter optimization strategy for the multi-objective performance of alloy 3J22G is designed using virtual sample generation and ML. By considering the extreme shortage of experimental data, two virtual sample generation methods are introduced for data enhancement. Based on the enhanced dataset, several ML models are trained to estimate the mapping between process parameters and target properties, namely the tensile strength and the elongation. A multi-objective optimization algorithm is applied to search the Pareto frontier of the tensile strength and elongation. Fifteen process parameter combinations are recommended, with the tensile strength reaching up to over 2000 MPa. Meanwhile, the elongation of these recommended alloys is increased from 1% up to 11%, which can meet the application requirements.

In summary, a new process parameter optimization strategy for the multi-objective performance improvement of 3J22G is proposed. This strategy abandons the traditional method of trial and error, and achieves simultaneous improvement of the tensile strength and the elongation of alloys. Its novelty is related to two factors: (1) the linear interpolation method and virtual sample generation method based on the GMM model are integrated to enhance the data, so as to more accurately predict the visible domain and effectively explore the invisible domain; and (2) for traditional ML models, it tends to fall into local optimal solutions. To solve this problem, we optimized the ML model by the sparrow search algorithm, and the improved model has a higher accuracy and better robustness.

## 2. Result and discussion

The strategy to guide the alloy design using virtual sample generation technologies and multi-objective optimization is shown in Figure 1. It consists of four parts: sample expansion, model construction, multi-objective optimization, and machine model interpretation. First, because the real dataset is very small, with only 13 data, 181 virtual samples are generated using the mixed interpolation method of linear interpolation and GMM interpolation; then, the generated 181 virtual samples are used to train the ML model to predict the alloy elongation and tensile strength, while 13 real raw data are used for model testing. The sparrow search algorithm (SSA) is incorporated into the ML model to optimize its hyperparameters, thus improving the prediction accuracy of the model. Then, the multi objective particle swarm optimization (MOPSO) is employed to establish the Pareto frontier. Based on the Pareto frontier, the design scheme of the alloy process parameters with a high tensile strength and elongation is determined.

### 2.1. Data preparation

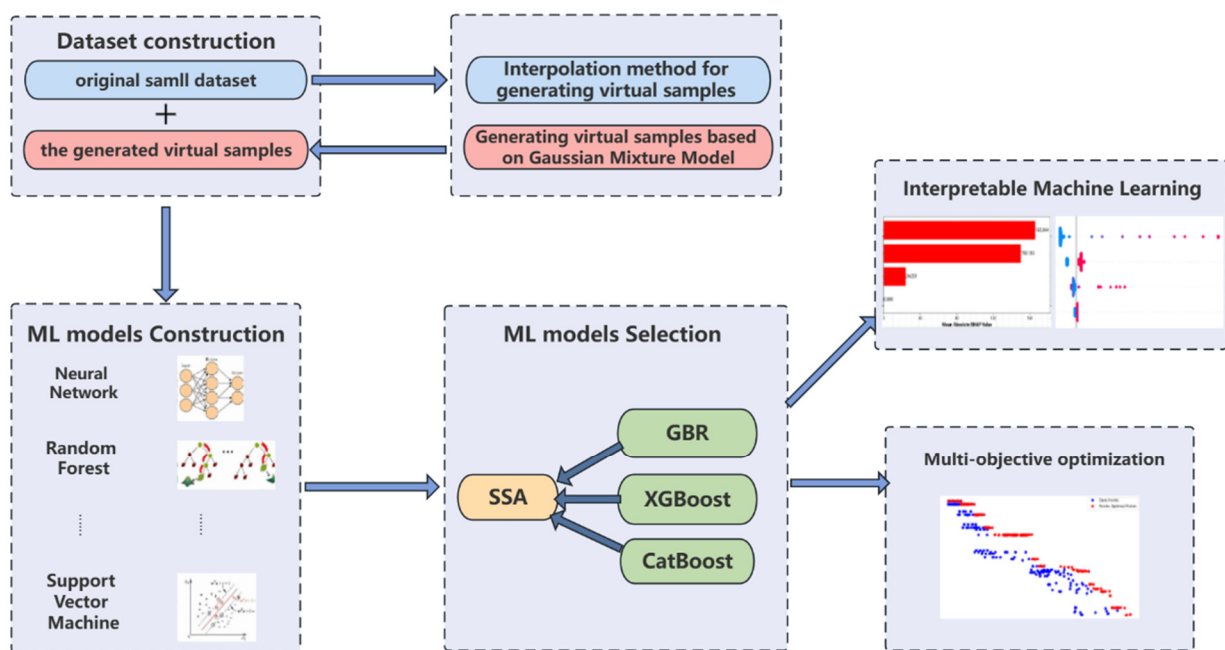
The data used in this work are from experiments. The data set includes six characteristics: cold rolling thickness (mm), aging temperature (°C), tensile test temperature (°C), tensile test shape, tensile strength (MPa), and elongation (%). The first four characteristics are the process parameters of the alloy, and the last two characteristics are the properties of the alloy. There are two types of tensile test shapes: I-shaped sample (marked as 1), and straight-bar sample (marked as 0).

The tensile strength and elongation are two important indexes of mechanical properties of materials, which are often used to describe the mechanical behavior and plastic deformation ability of materials. The tensile strength refers to the ability of a material to resist fracture under tensile loading. It is the ratio of the maximum tension that the material can withstand during the tensile process to the real cross-

sectional area. A higher tensile strength indicates that the material has a stronger resistance when it is stressed and is not easy to break. The elongation refers to the degree of tensile deformation that a material can produce before breaking under tensile loading. It refers to the percentage between the length of growth before the material breaks and the real length. A higher elongation indicates that the material can have a greater degree of plastic deformation when it is stressed and has better toughness.

There is usually an inverse relationship between the tensile strength and the elongation, that is, the higher the tensile strength of the material, the lower its elongation tends to be, and vice versa. This is because during the tensile process, when the tensile strength of the material is higher, it is more likely to break, so the elongation is relatively low. On the contrary, when the elongation of the material is higher, it is able to undergo a greater degree of plastic deformation, so the tensile strength is relatively low.

All available data shown in Table 1.



**Figure 1.** The design framework for searching for HEAs with high tensile strength and elongation.

**Table 1.** Available data.

| No | Cold-rolled thickness | Aging temperature | Tensile test temperatures | Shape               | Rm (tensile strength) MPa | A (Elongation)% |
|----|-----------------------|-------------------|---------------------------|---------------------|---------------------------|-----------------|
| #1 | 0.15                  | 425               | 20                        | I-shaped sample     | 1824                      | 2               |
| #2 | 0.15                  | 425               | 700                       | I-shaped sample     | 648                       | 26              |
| #3 | 0.15                  | 425               | 700                       | I-shaped sample     | 624                       | 29.5            |
| #4 | 0.15                  | 500               | 20                        | I-shaped sample     | 2092                      | 1               |
| #5 | 0.15                  | 800               | 20                        | I-shaped sample     | 1298                      | 13.5            |
| #6 | 0.4                   | 425               | 20                        | Straight-bar sample | 1520                      | 0.24            |
| #7 | 0.4                   | 425               | 20                        | Straight-bar sample | 1662                      | 0.32            |

*Continued on next page*

| No  | Cold-rolled thickness | Aging temperature | Tensile test temperatures | Shape               | Rm (tensile strength) MPa | A (Elongation)% |
|-----|-----------------------|-------------------|---------------------------|---------------------|---------------------------|-----------------|
| #8  | 0.4                   | 425               | 20                        | I-shaped sample     | 1816                      | 3               |
| #9  | 0.4                   | 480               | 20                        | I-shaped sample     | 1908                      | 3               |
| #10 | 0.4                   | 500               | 20                        | I-shaped sample     | 1926                      | 4               |
| #11 | 0.4                   | 520               | 20                        | I-shaped sample     | 1768                      | 1.5             |
| #12 | 0.15                  | 425               | 20                        | Straight-bar sample | 1795                      | 0.5             |
| #13 | 0.15                  | 425               | 20                        | Straight-bar sample | 1422                      | 0.1             |

## 2.2. Virtual sample generation

Due to the small number of available alloy samples, it is a challenge to build a robust and accurate prediction model through data-driven methods. In order to solve the problem of small datasets, the virtual sample generation (VSG) method can be used to generate new virtual samples according to the trend of a real small dataset, which can also improve the modeling performance.

In regression problems, if data points are sparse or the data coverage is limited, interpolation and extrapolation methods can be used to generate new data points. Interpolation methods estimate data points based on the relationship between the existing data points, while extrapolation methods predict new data points based on the trends and rules of the existing data points. These methods can help fill data gaps, and expand the range of the dataset. Therefore, in this paper, the interpolation method is used to generate 171 samples between real data points, and the GMM is used to generate 10 samples with high tensile strengths, so as to achieve an extrapolation based on the real data.

### 2.2.1. The interpolation method generates data

The existing samples are grouped by controlling different variables. For example, samples with a cold-rolled thickness of 0.4 mm, I-shaped sample, and a temperature of 20 °C for the tensile test are defined as group A1, for which only the aging temperature changes. There are four samples in group A1: #8, #9, #10, and #11. Nine samples are inserted between each two samples in A1. The combination number of the four samples in group A1 is  $C_4^2 = 6$ ; therefore,  $6 \times 9 = 54$  samples are inserted for group A1. A total of 171 samples were inserted in the same way. The equations used for interpolation between two samples are as follows:

$$x_{vs} = x_i + t(x_{i+1} - x_i), \quad (1)$$

$$y_{vs} = y_i + t(y_{i+1} - y_i), \quad (2)$$

where  $x_i$  and  $x_{i+1}$  are the  $i$  and  $i + 1$  samples, respectively, that is, interpolation is performed between  $x_i$  and  $x_{i+1}$ . The parameter  $t$  is used to determine the number of virtual samples inserted between two data points. Set  $t = 0.1$ , where 9 samples are inserted between each two samples.

The remaining groups A2 to A5 are interpolated as described above. One hundred and seventy-one virtual samples were obtained, as shown in Table 2.

**Table 2.** In interpolation-based virtual sample generation, controlling variable conditions.

| Quantitative |                                                                              | Variables                   | Real samples used | Number of generated samples |
|--------------|------------------------------------------------------------------------------|-----------------------------|-------------------|-----------------------------|
| A1           | Room temperature tensile, I-shaped sample, Cold-rolled thickness 0.4 mm      | Aging heat temperature      | #8, #9, #10, #11  | $6 \times 9 = 54$           |
| A2           | Room temperature tensile, I-shaped sample, Cold-rolled thickness 0.15 mm     | Aging heat temperature      | #1, #4, #5        | $3 \times 9 = 27$           |
| A3           | Room temperature tensile, I-shaped sample, Aging heat temperature 500 °C     | Cold-rolled thickness       | #10, #4           | $1 \times 9 = 9$            |
| A4           | Room temperature tensile, Straight-bar sample, Aging heat temperature 425 °C | Cold-rolled thickness       | #6, #7, #12, #13  | $6 \times 9 = 54$           |
| A5           | Aging heat temperature 425 °C, Cold-rolled thickness 0.15 mm                 | Tensile testing temperature | #1, #2, #3        | $3 \times 9 = 27$           |

### 2.2.2. Sample generation method based on GMM model

The sampling process of the GMM model is based on the fitted Gaussian mixture distribution. The following is the detailed principle of sampling using the GMM model. Use the existing 13 samples to create a GMM model, and specify the number of sub-models, covariance type, and other parameters. Training data is used to fit the GMM model, and the parameters of the model are estimated by the expectation-maximization (EM) algorithm. In the fitting process, the model learns the weight, mean, and covariance matrix of each sub-model to capture the distribution characteristics of the data.

Sample the well-fitted GMM model, where one of the sub-model components of the model is selected according to the weight of the sub-model. The greater the weight, the higher the probability the sub-model is selected. Sample a data point from the Gaussian distribution of the selected component, that is, generate a random number based on the Gaussian distribution as the sample value. Multiple sample data points can be generated by repeating the above steps to select the sub-model according to the weights and sampling data points from the Gaussian distribution of the selected sub-model. The data points obtained by sampling reflect the characteristics of the data distribution learned by the model.

### 2.3. Virtual sample evaluation

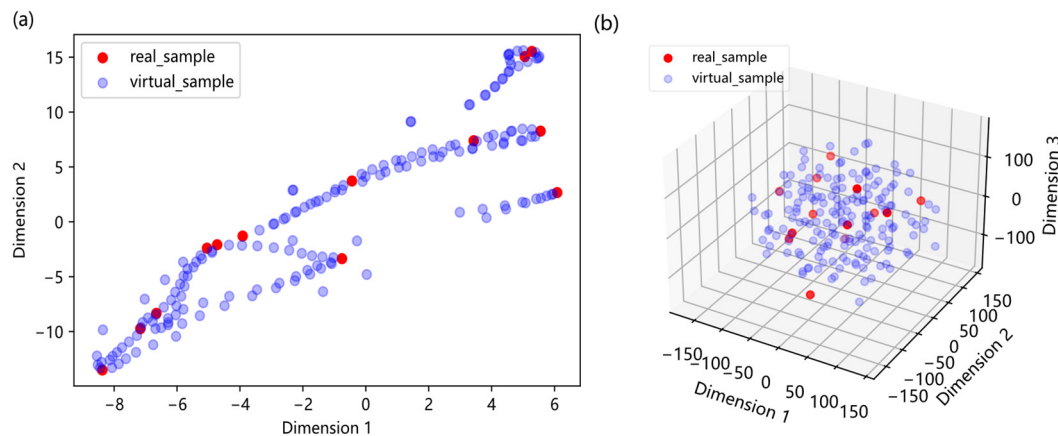
In order to evaluate the quality of the virtual sample inserted above, the following three aspects of virtual sample evaluation are probability density estimation, visual observation, and result comparison.

#### 2.3.1. Visualize the generated virtual sample

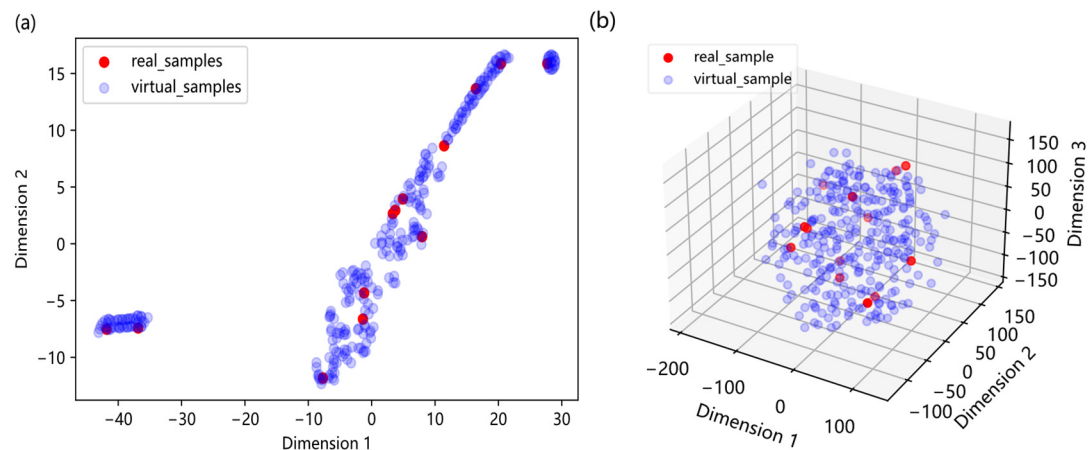
In order to further understand the distribution of the generated virtual samples, the t-distributed stochastic neighbor embedding (t-SNE) dimensionality reduction is applied to reduce the dataset of 6 dimensions to 2 and 3 dimensions. Then scatter plots on a 2D plane and a 3D space are plotted, as shown in Figure 2. After dimensionality reduction, the real samples are evenly distributed among the generated virtual samples. It can be seen that the distribution of the generated sample is roughly the same as the real sample.

### 2.3.2. Probability density estimation

In addition, the probability density histogram is a method used to represent the distribution of samples in the feature space. The probability density curves of real samples and generated samples are drawn in Figure 4. It can be observed that in different feature dimensions, the probability density curves of virtual samples and real samples are similar in shape, and the peak value and trend are consistent. It means that the generated samples can simulate the characteristics of real samples well. Therefore, comparing and analyzing the probability density curves of real and generated samples shows that the generated samples can effectively capture the distribution and characteristics of real samples. It concludes that the quality of the generated samples is commendable.



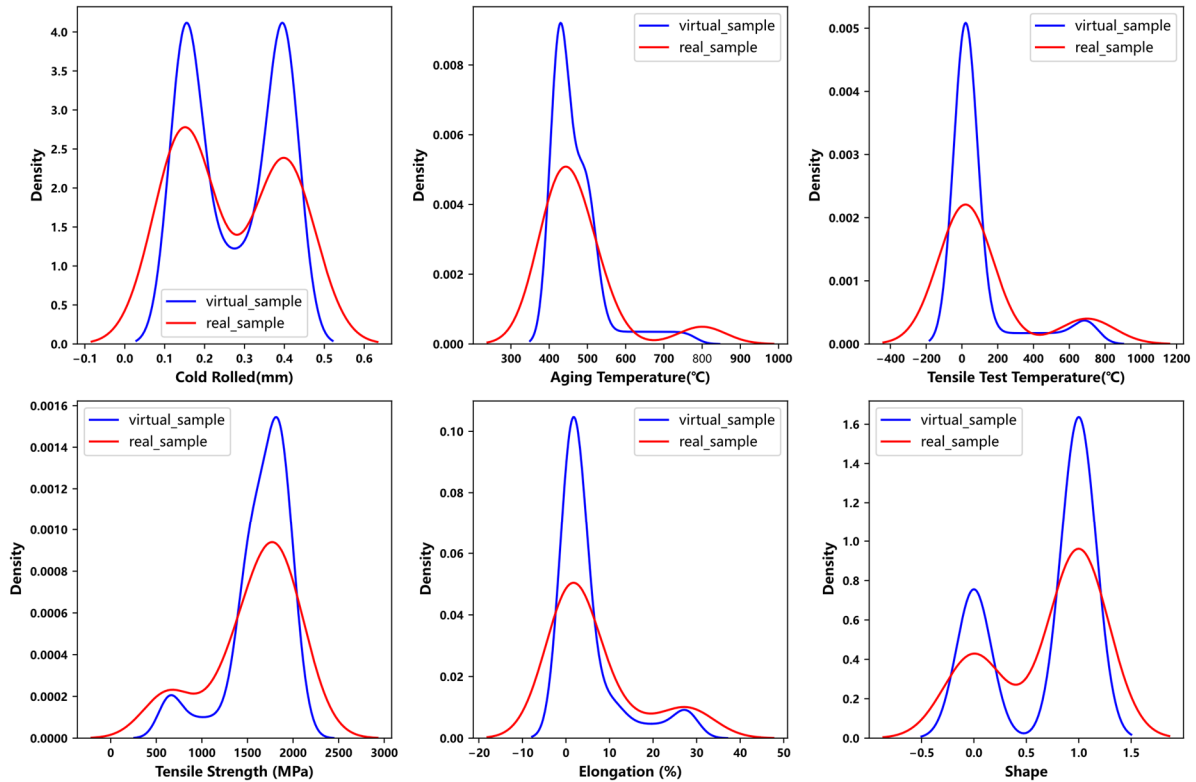
**Figure 2.** t-SNE dimensionality reduction scatter plot of virtual sample (blue dot) and real sample (red dot) generated by interpolation. (a) The two-dimensional scatter plot. (b) The three-dimensional scatter plot.



**Figure 3.** t-SNE reduced-dimension scatter plot of virtual samples (blue dots) and real samples (red dots) generated by the GMM. (a) The two-dimensional scatter plot. (b) The three-dimensional scatter plot.

### 2.3.3. Comparison of results

To further evaluate the quality of the generated virtual samples, a comparative experiment is conducted using the Random Forest model based on two distinct datasets: one solely consisting of the real samples, and the other containing both the virtual and real samples. The experimental results show that the model trained with virtual samples exhibited a superior performance in terms of the prediction. This not only reflects the high quality of the generated virtual samples but also demonstrates the effectiveness of the training strategy that integrates real samples with virtual samples. The addition of virtual samples can effectively improve the performance of the regression model.



**Figure 4.** Probability density histogram of the real and virtual samples, where blue is the virtual sample and red is the real sample.

Initially, a baseline regression model is constructed using the real samples, and its performance is evaluated on the test set. The evaluation metrics, namely the mean absolute error (MAE) and the mean absolute percentage error (MAPE), are calculated by the following:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|, \quad (3)$$

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{|y_i|}, \quad (4)$$

where  $y_i$  and  $\hat{y}_i$  represent the real value and predicted value of the  $i$ -th sample, respectively, and  $n$  is the size of the training or test set.

Subsequently, the generated virtual samples are combined with the real samples for model training, and the model is reevaluated on the test set. The evaluation metrics are calculated. Due to the

sparsity of the material data, a comprehensive evaluation of the model is necessary. In this paper, leave-one cross-validation (LOOCV) is an effective method to evaluate the extrapolation and prediction ability of models trained based on small datasets, which can make full use of datasets for training during the evaluation process. LOOCV is a special case of k-fold, where k is equal to the number of samples in the dataset. The MAE and MAPE results are shown in Table 3. By comparing the results of the model trained by adding virtual samples with the model trained only using real samples, it is observed that the model has significant improvements in all evaluation indexes after adding virtual samples for training. The results reflect the high quality of the generated virtual samples.

**Table 3.** The comparison of MAE and MAPE for the model trained based on two datasets: one is the real samples and the other is the union of real samples and the generated samples.

|                              | MAE of Rm | MAPE of Rm | MAE of A | MAPE of A |
|------------------------------|-----------|------------|----------|-----------|
| Real samples                 | 196.56    | 0.18       | 3.54     | 1.12      |
| Real samples+virtual samples | 111.29    | 0.06       | 0.91     | 0.51      |

In summary, virtual sample generation techniques can produce high-quality samples and make up for the problem of data scarcity. It expands the size and diversity of datasets, which helps the model better learn the patterns and relationships of the data, and improves the generalization ability and prediction accuracy of the model. Subsequent research will be built upon the foundation of the virtual data.

#### 2.4. Machine learning model performance evaluation

Based on the virtual samples generated by the interpolation method and the GMM, nine basic machine learning models are trained: random forest (RF), support vector regression (SVR), artificial neural network (ANN), gradient boosting regression (GBR), extreme gradient boosting (XGBoost), categorical boosting (CatBoost), ridge regression, lasso regression, and elastic net regression (Elastic Net). For different ML models, the two evaluation indexes of decision coefficient  $R^2$  and root mean square error (RMSE) are evaluated. The formulas are shown as Eqs (5) and (6). The evaluation metric results for each model are shown in Tables 4 and 5.

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y^{(l)})^2}{\sum_{i=1}^n (\bar{y} - y_i)^2}, \quad (5)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}, \quad (6)$$

where  $n$  is the sample size,  $\hat{y}_i$  is the predicted value,  $y_i$  is the real value, and  $\bar{y}$  is the average value.

**Table 4.** Evaluation results of different ML models for the tensile strength Rm.

|        | $R^2$ | RMSE    | MAE     |
|--------|-------|---------|---------|
| RF_Rm  | 0.929 | 123.886 | 104.777 |
| SVR_Rm | 0.924 | 122.757 | 95.474  |

*Continued on next page*

|                | $R^2$ | RMSE    | MAE     |
|----------------|-------|---------|---------|
| ANN_Rm         | 0.934 | 105.058 | 83.970  |
| XGBoost_Rm     | 0.914 | 130.295 | 104.523 |
| GBR_Rm         | 0.931 | 117.146 | 102.679 |
| CatBoost_Rm    | 0.905 | 136.821 | 97.515  |
| Ridge_Rm       | 0.918 | 127.603 | 109.054 |
| Lasso_Rm       | 0.918 | 127.41  | 108.376 |
| Elastic_Net_Rm | 0.915 | 129.897 | 109.183 |

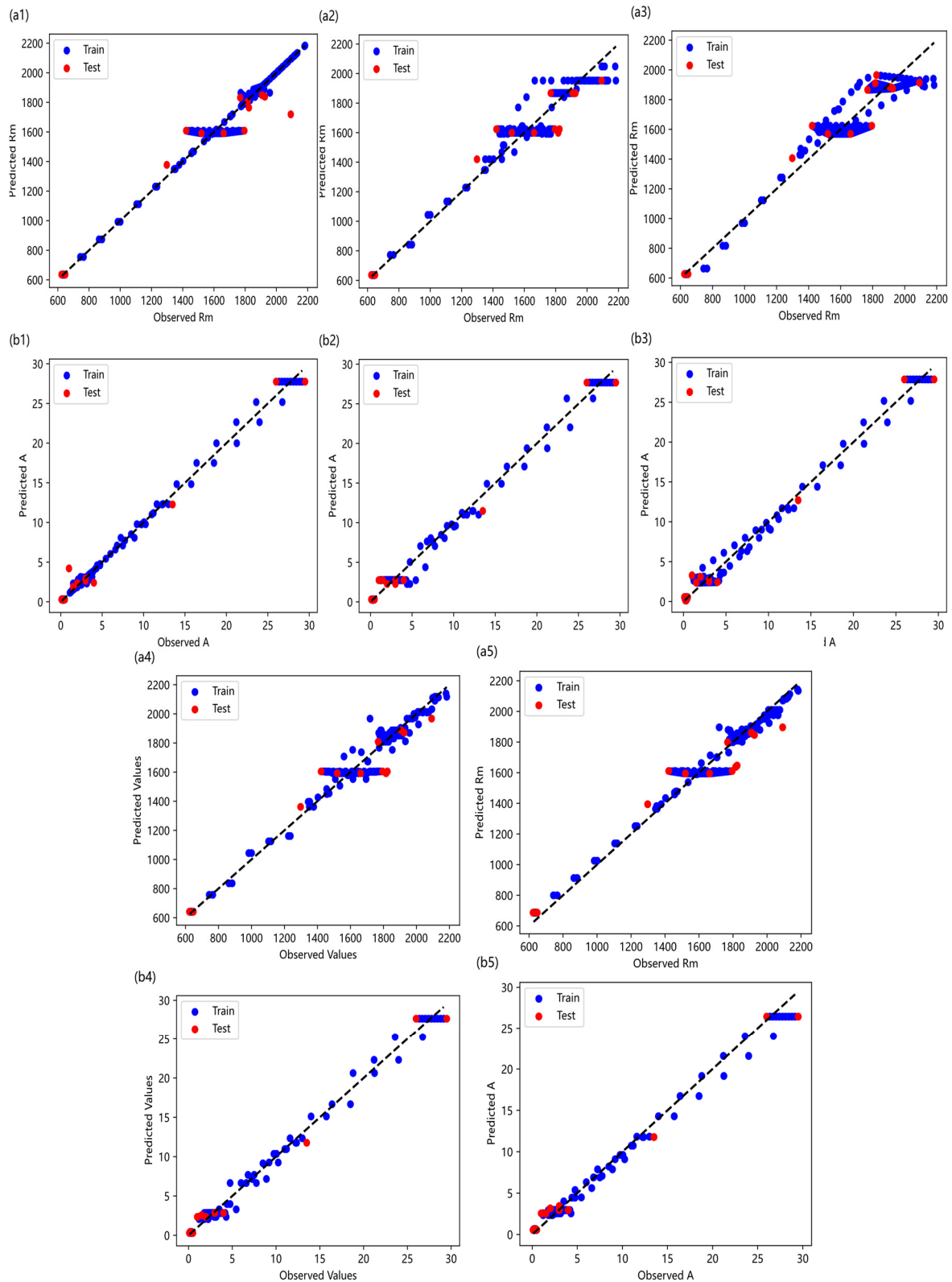
**Table 5.** Evaluation results of different ML models of elongation A%.

|               | $R^2$ | RMSE  | MAE   |
|---------------|-------|-------|-------|
| RF_A          | 0.984 | 1.359 | 1.081 |
| SVR_A         | 0.972 | 1.618 | 1.161 |
| ANN_A         | 0.985 | 1.176 | 0.930 |
| XGBoost_A     | 0.988 | 1.060 | 0.813 |
| GBR_A         | 0.983 | 1.250 | 0.947 |
| CatBoost_A    | 0.982 | 1.296 | 0.875 |
| Ridge_A       | 0.980 | 1.375 | 1.086 |
| Lasso_A       | 0.979 | 1.396 | 1.200 |
| Elastic_Net_A | 0.980 | 1.376 | 1.130 |

10-fold cross validation is a reliable way to evaluate the performance of ML models. By using different combinations of training sets and validation sets, it can more accurately evaluate the performance of the model on different data subsets, reduce the risk of overfitting or underfitting specific datasets, and provide more reliable model evaluation results. The results of the 10-fold cross-validation of the above models are shown in Table 6. The results show that RF, SVR, XGBoost, GBR, and CatBoost have better performances.  $R^2$  reaches 0.9 in the prediction tasks of the two properties. By comparing the predicted results of these five models with the real samples (Figure 5), it can be observed that the CatBoost, XGBoost, and GBR models performed better. Therefore, we further optimize these three models and search for the process parameter with a high tensile strength and a high elongation.

**Table 6.**  $R^2$  of different models on the test set after ten-fold cross-validation.

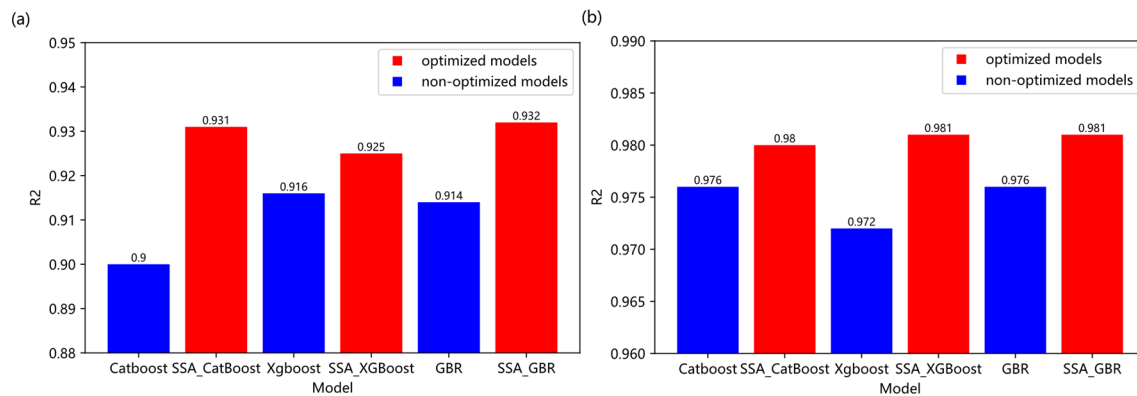
| Model       | Rm_ $R^2$ | A_ $R^2$ |
|-------------|-----------|----------|
| RF          | 0.906     | 0.981    |
| SVR         | 0.850     | 0.958    |
| ANN         | 0.913     | 0.986    |
| XGboost     | 0.916     | 0.972    |
| GBR         | 0.914     | 0.976    |
| Catboost    | 0.900     | 0.976    |
| Ridge       | 0.887     | 0.978    |
| Lasso       | 0.888     | 0.975    |
| Elastic_net | 0.884     | 0.977    |



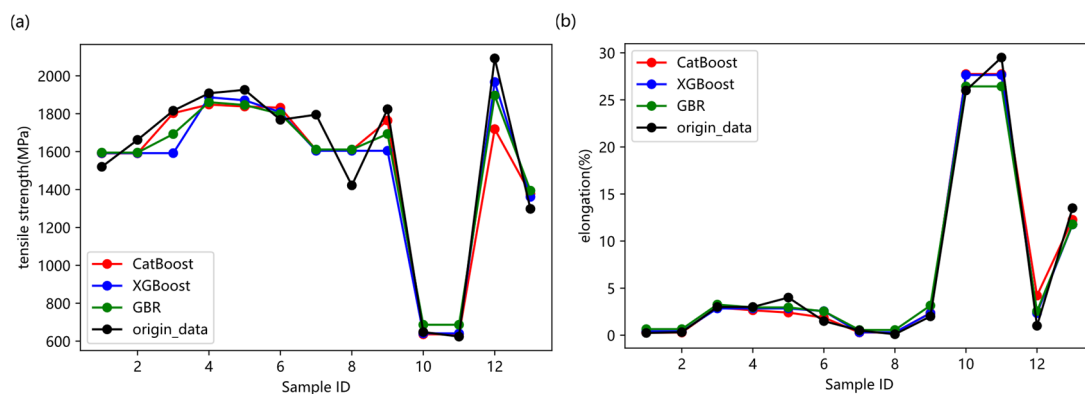
**Figure 5.** The virtual data serves as the training set, and different models predict the results of the real data. A1- a5 represent CatBoost, RF, SVR, XGBoost, and GBR prediction results on tensile strength, represent. b1-b5 represent CatBoost, RF, SVR, XGBoost, and GBR prediction results on elongation, respectively.

Based on the above results, three algorithms with better performances are selected, namely CatBoost, XGBoost, and GBR. However, there are still some challenges in ML. For example, the large parameter space of the model leads to time-consuming hyperparameter tuning. The introduction of the intelligent optimization algorithm can overcome the limitations of traditional optimization methods.

As a global optimization algorithm, the SSA can avoid falling into the local optimal solution, the iteration speed is fast, and the parameter optimization effect is good. The SSA has a strong local search ability and effectively optimizes the parameters of the ML model. Therefore, the SSA is introduced to iteratively search for the best parameter combination, which can further improve the performance of the model. The  $R^2$  of the ML model with the SSA algorithm and the basic ML model is shown in Figure 6 using 10-Fold Cross-validation. We can observe the performance improvement of the model with the SSA. The optimized parameter combination makes the model achieve better results in the prediction task. These results prove the effectiveness of the SSA algorithm in parameter tuning of ML models. The prediction results of the tensile strength and the elongation for the optimized ML models are shown in Figure 7. It suggests that the predicted result of the established model is consistent with the real value.



**Figure 6.** Comparison of  $R^2$  between the SSA optimized model and the no-optimized model.



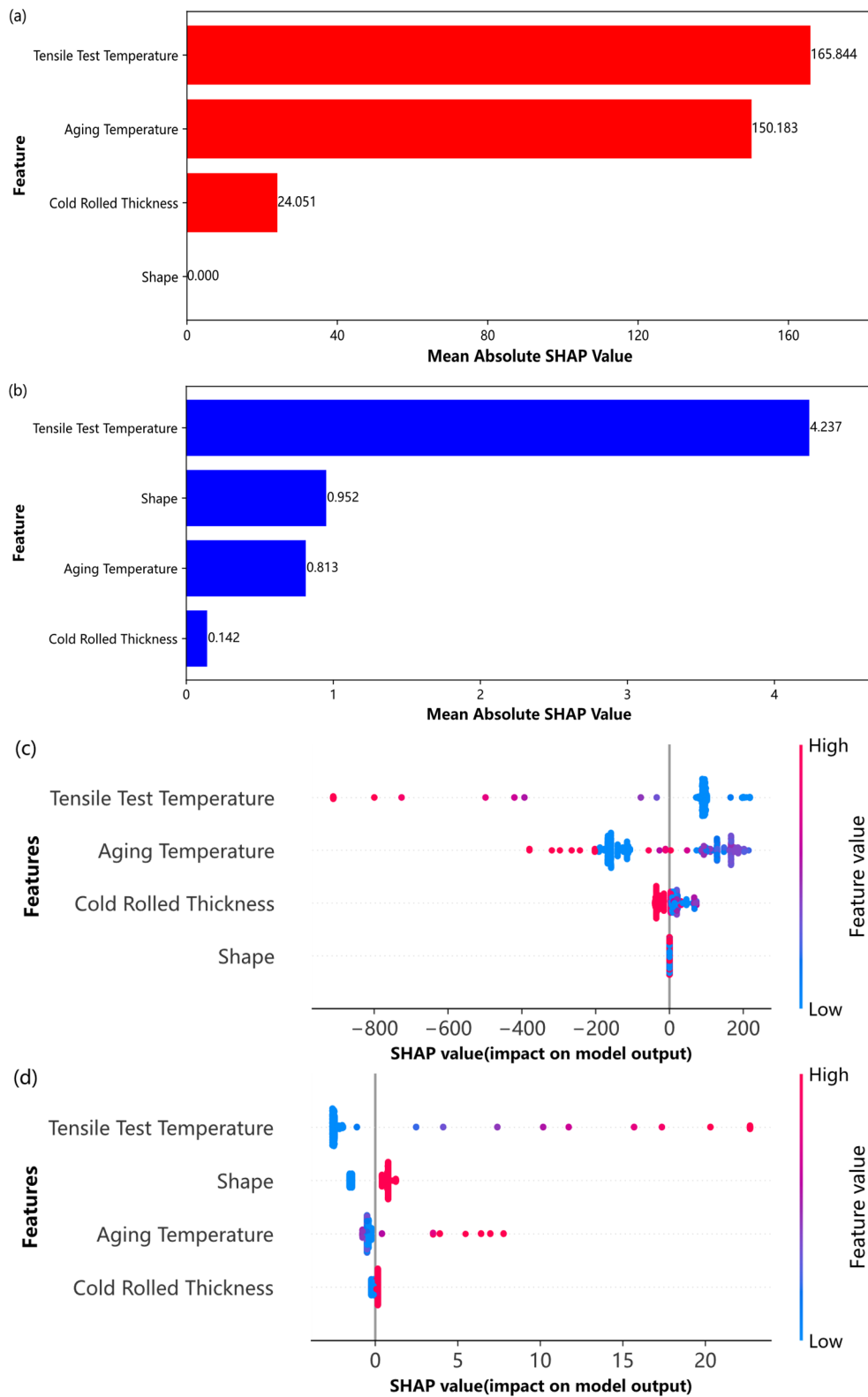
**Figure 7.** Comparison of prediction results between different models and real samples. (a) Comparison of prediction results of tensile strength. (b) Comparison of prediction results of elongation.

## 2.5. Interpretable machine learning

As a black box model, it is difficult to explain and understand the complex prediction mechanism behind the ML models, let alone know how the process parameters affect the tensile strength and elongation of the alloy. By improving the interpretability of the model, we can further optimize the process parameters and illustrate the mechanism of how the process parameter affects the target performance.

In this paper, the shapley additive explanation (SHAP) is applied to explain the ML model. As a method of model post-interpretation, the core idea of the SHAP is to calculate the marginal contribution of features to the model output and analyze the degree of influence of each feature on the target performance and whether it promotes or inhibits the effect. Figure 8(a) shows the features importance ranking of the tensile strength  $R_m$ . The result shows that the tensile test temperature and the aging temperature are the first and second important factors for the tensile strength, and the cold rolling thickness also has some influence on it, while the shape has almost no influence on it. Figure 8(b) shows the features importance ranking of elongation  $A$ . The analysis reveals that tensile test temperature exerts the most significant influence on the elongation. Subsequently, aging temperature demonstrates a notable impact, followed by shape, while cold rolling thickness's effect remains comparatively modest.

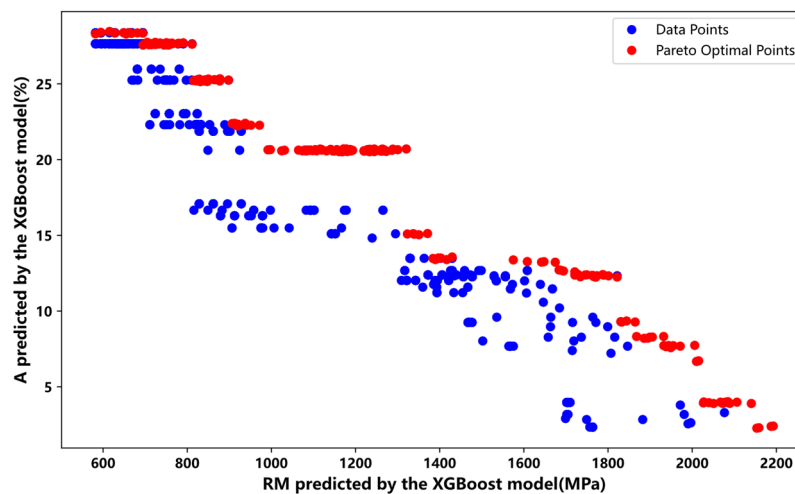
For the SHAP analysis results, there is a distribution of feature SHAP values about the tensile strength. Each sampling point in Figure 8(c) represents a tensile strength data. The vertical axis ranks the features according to the sum of the SHAP values of all samples, and the horizontal axis represents the SHAP value of each sample. The larger the SHAP value, the greater the influence of the feature on the tensile strength, the larger the coverage of the feature scattering, and the greater the influence of the tensile strength. For the tensile test temperature, it can be observed that high temperatures (red) have a negative effect on the predicted results, while lower temperatures (blue) have a positive effect on the predicted results. This means that decreasing the tensile test temperature may result in a higher tensile strength. For the aging temperature, both too high and too low temperatures seem to have a negative effect on the tensile strength, which implies that it is necessary to select a moderate aging temperature to achieve a high tensile strength. Finally, reducing the cold-rolled thickness may play a positive role in improving the tensile strength. In the same way, the influence of the process parameters on the elongation can be obtained in Figure 8(d). For the tensile test temperature and the aging temperature, a high temperature (red) has a positive influence on the elongation, while a low temperature (blue) has a negative influence on the elongation. Therefore, increasing the tensile test temperature and the aging temperature may improve the elongation of the material. Increasing the cold-rolled thickness has little effect on improving the elongation. The shape has no effect on the tensile strength of the alloy, but it significantly influences the elongation; the I-shaped sample demonstrates a noticeably better elongation compared to the straight-bar sample.



**Figure 8.** Feature analysis of SHAP on the XGBoost Model. (a) Features importance ranking of the tensile strength  $R_m$ . (b) Features importance ranking of elongation  $A$ . (c) The SHAP value distribution of each process parameter of alloy about the tensile strength. (d) The SHAP value distribution of each process parameter about the elongation of alloy.

## 2.6. Optimization of process parameters for multi-objective properties

Because of the mutually exclusive relationship between the tensile strength and the elongation, it is difficult to find the alloy manufacturing parameters with a high tensile strength and a high elongation. The MOPSO algorithm, known for its attributes of easy comprehension, implementation, parameter configuration, and rapid convergence, is employed to search the Pareto front for the tensile strength (Rm) and elongation (A), thereby obtaining optimal process parameters corresponding to good combinations of Rm and A. Pareto optimal graphs selected by the XGBoost model prediction (the red dots) are shown in Figure 9, where each red dot corresponds to a process parameter. In the Pareto frontier, 15 candidate combinations of process parameters with good tensile strengths and elongations are selected. These 15 groups of candidate process parameter combinations are based on the XGBoost model search, CatBoost, and GBR model auxiliary prediction; their prediction results on these three models are relatively consistent, and have quite good performances. Due to the high performance of the above models after strict cross-validation, we can tend to think that they are better than other alloys with high performances predicted by only one model, which improves the confidence of the recommended alloys to a certain extent. Recommended alloys are shown in Table 7. Figure 10 shows the alloy performance predicted by the ML models for 15 selected process parameter combinations and the performances of existing alloys. The graph illustrates a significant improvement in both the elongation and the tensile strength of the alloy relative to the original dataset. Based on this observation, we can conclude that the proposed framework effectively optimizes process parameters for multi-objective properties.



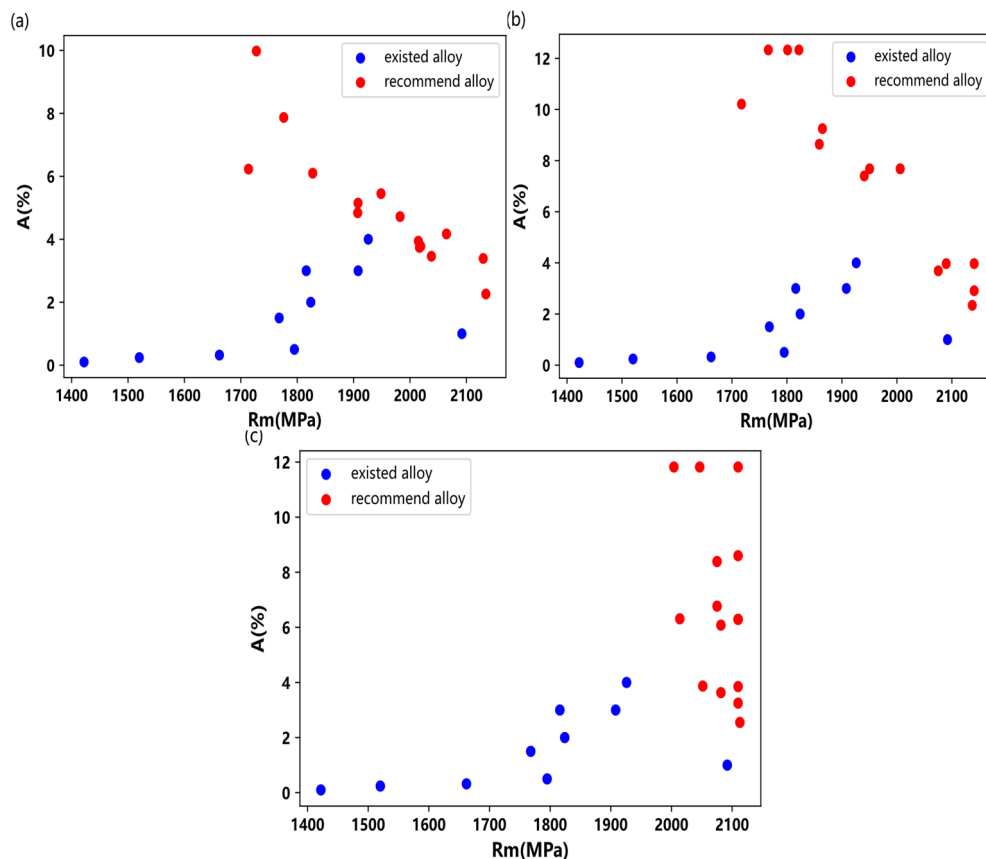
**Figure 9.** Pareto optimal graph selected by the XGBoost model prediction.

**Table 7.** Fifteen candidate alloys are presented.

| NO. | Cold-rolled thickness | Aging temperature | Tensile test temperature | Shape | CatBoost_Rm | CatBoos t_A | XGBoos t_Rm | XGBo ost_A | GBR_R m | GBR_A |
|-----|-----------------------|-------------------|--------------------------|-------|-------------|-------------|-------------|------------|---------|-------|
| (1) | 0.05                  | 490               | 0                        | 1     | 2134.48     | 2.26        | 2137.02     | 2.34       | 2112.78 | 2.55  |
| (2) | 0.20                  | 470               | 30                       | 1     | 2129.78     | 3.39        | 2140.49     | 2.91       | 2109.85 | 3.25  |

*Continued on next page*

| NO.  | Cold-rolled thickness | Aging temperature | Tensile test temperature | Shape | CatBoost_<br>Rm | CatBoos<br>t_A | XGBoos<br>t_Rm | XGBo<br>ost_A | GBR_R<br>m | GBR_A |
|------|-----------------------|-------------------|--------------------------|-------|-----------------|----------------|----------------|---------------|------------|-------|
| (3)  | 0.25                  | 470               | 70                       | 1     | 2064.71         | 4.17           | 2140.49        | 3.97          | 2109.85    | 3.85  |
| (4)  | 0.20                  | 520               | 80                       | 1     | 2037.96         | 3.46           | 2075.20        | 3.69          | 2081.52    | 3.63  |
| (5)  | 0.30                  | 490               | 120                      | 1     | 2014.82         | 3.94           | 2089.63        | 3.97          | 2051.62    | 3.87  |
| (6)  | 0.20                  | 470               | 140                      | 1     | 1948.61         | 5.45           | 2005.92        | 7.68          | 2109.85    | 6.29  |
| (7)  | 0.25                  | 560               | 180                      | 1     | 2019.00         | 3.78           | 1858.78        | 8.64          | 2075.18    | 6.77  |
| (8)  | 0.30                  | 500               | 160                      | 1     | 1907.23         | 4.84           | 1950.33        | 7.68          | 2013.71    | 6.31  |
| (9)  | 0.25                  | 530               | 130                      | 1     | 2016.79         | 3.74           | 1940.63        | 7.40          | 2081.52    | 6.08  |
| (10) | 0.20                  | 490               | 140                      | 1     | 1982.40         | 4.72           | 2005.92        | 7.68          | 2109.85    | 6.29  |
| (11) | 0.20                  | 550               | 250                      | 1     | 1907.80         | 5.15           | 1717.34        | 10.21         | 2075.18    | 8.39  |
| (12) | 0.25                  | 480               | 210                      | 1     | 1827.48         | 6.10           | 1864.48        | 9.25          | 2109.85    | 8.60  |
| (13) | 0.35                  | 500               | 260                      | 1     | 1713.61         | 6.23           | 1766.12        | 12.33         | 2004.01    | 11.82 |
| (14) | 0.25                  | 490               | 260                      | 1     | 1776.10         | 7.87           | 1821.71        | 12.33         | 2109.85    | 11.82 |
| (15) | 0.20                  | 460               | 260                      | 1     | 1727.65         | 9.98           | 1801.17        | 12.33         | 2046.56    | 11.82 |



**Figure 10.** Comparison diagram of properties of recommended candidate alloys with existing alloys, where (a), (b), and (c) are comparison diagrams of properties predicted by the CatBoost model, GBR model, and XGBoost model of candidate alloys with existing alloys, respectively. The blue dots are the existing alloys, and the red dots are the recommended alloys.

### 3. Conclusions

In this study, based on the small sample data, we designed a strategy to optimize process parameters using ML to achieve a better elongation and tensile strength for 3J22G. This strategy combines virtual sample generation technologies, ML and optimization technology to facilitate the process parameter design.

First, the interpolation method and virtual sample generation method based on the GMM are used to expand the original data. This method can improve the predictive performance of regression model in a regression task with a small number of training samples. Second, according to the key characteristics of the data, namely cold rolling thickness, aging temperature, tensile test temperature, and tensile test shape, the ML model of the tensile strength and the elongation were established. We established nine machine learning models for multiple performance prediction: RF, SVR, ANN, Elastic network, XGBoost, CatBoost, GBR, Lasso regression, and Ridge regression. The above models were evaluated using ten-fold cross-validation, and the results showed that RF, SVR, XGBoost, GBR, and Catboost had higher  $R^2$ , with  $R^2$  above 0.9 for all five models. Furthermore, among the above five models, the prediction results of CatBoost, XGBoost, and GBR were shown to have smaller errors and better performances. Third, the hyper-parameters in the models CatBoost, XGBoost, and GBR were optimized by the SSA. Multi-objective particle swarm optimization algorithm was used to search the appropriate combination of process parameters. Using this strategy, 15 groups of process parameter combinations were quickly designed to meet the performance requirements. Three of the process parameter combinations exhibited tensile strengths above 2000 MPa and elongations of approximately 11%, markedly surpassing the performance of current alloys. This data-driven ML strategy enhances the application prospects in material performance optimization, and it can greatly reduce the time and cost of experimentation and research in material development. Finally, in order to understand the model from the black box machine learning model, the interpretable method was used to analyze the alloy strength and elongation. SHAP analysis results showed that increasing the tensile test temperature can increase the elongation, while it reduces the tensile strength. A relatively high aging temperature is beneficial to increase the elongation. However, to improve the tensile strength, the aging temperature should not be too high or too low. The present work provided a new design framework of the process parameters for the 3J22G alloy with a multi-objective performance based on ML, which can accelerate the development of the new alloys.

### Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

### Acknowledgements

This work was supported by the National Key R&D Program of China (2024YFB3813700).

### Conflict of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Lundberg SM, Lee SI, (2017) A unified approach to interpreting model predictions. *Adv Neural Inf Proces Syst* 2017: 30.
2. Rao Z, Tung PY, Xie R, Wei Y, Zhang H, Ferrari A, et al., (2022) Machine learning—enabled high-entropy alloy discovery. *Science* 378: 78–85. <https://doi.org/10.1126/science.abo4940>
3. Sutojo T, Rustad S, Akrom M, Syukur A, Shidik GF, Dipojono HK, (2023) A machine learning approach for corrosion small datasets. *NPJ Mater Degrad* 7: 18. <https://doi.org/10.1038/s41529-023-00336-7>
4. Jiang L, Wang C, Fu H, Shen J, Zhang Z, Xie J, (2022) Discovery of aluminum alloys with ultra-strength and high-toughness via a property-oriented design strategy. *J Mater Sci Technol* 98: 33–43. <https://doi.org/10.1016/j.jmst.2021.05.011>
5. Li H, Yuan R, Liang H, Wang WY, Li J, Wang J, (2022) Towards high entropy alloy with enhanced strength and ductility using domain knowledge constrained active learning. *Mater Des* 223: 11118. <https://doi.org/10.1016/j.matdes.2022.111186>
6. Zhang Y, Wen C, Wang C, Antonov S, Xue D, Bai Y, et al., (2020) Phase prediction in high entropy alloys with a rational selection of materials descriptors and machine learning models. *Acta Mater* 185: 528–539. <https://doi.org/10.1016/j.actamat.2019.11.067>
7. Ritchie RO, (2011) The conflicts between strength and toughness. *Nat Mater* 10: 817–822. <https://doi.org/10.1038/nmat3115>
8. Lu K, (2014) Making strong nanomaterials ductile with gradients. *Science* 345: 1455–1456. <https://doi.org/10.1126/science.1255940>
9. Zhang H, Fu H, Shen Y, Xie J, (2022) Rapid design of secondary deformation-aging parameters for ultra-low Co content Cu-Ni-Co-Si-X alloy via Bayesian optimization machine learning. *Int J Miner Metall Mater* 29: 1197–1205. <https://doi.org/10.1007/s12613-022-2479-3>
10. Li C, Xue Y, Hua M, Cao T, Ma L, Wang L, (2016) Microstructure and mechanical properties of  $\text{Al}_x\text{Si}_{0.2}\text{CrFeCoNiCu}_{1-x}$  high-entropy alloys. *Mater Des* 90: 601–609. <https://doi.org/10.1016/j.matdes.2015.11.013>
11. Luo LY, Chen SN, Liu Y, Ji G, Chen HD, Jia YJ, et al., (2022) Microstructure and mechanical properties of lightweight  $\text{Al}_x\text{CrNbTiV}$  ( $x = 0.2, 0.5, 0.8$ ) refractory high entropy alloys. *Int J Refract Met Hard Mater* 104: 105784. <https://doi.org/10.1016/j.ijrmhm.2022.105784>
12. Kondapalli VAS, Suresh K, Ramakrishna M, Narasaiah N, Srinivasarao B, (2023) Effect of Cu content on the microstructure and mechanical properties of  $\text{FeNiMnCu}_x\text{Al}_{0.1}\text{Ti}_{0.1}$  ( $x = 0.5, 1.0$  and  $1.5$ ) high entropy alloy system. *J Alloys Compd* 940: 168819. <https://doi.org/10.1016/j.jallcom.2023.168819>
13. Wang R, Yan Z, He J, Fan W, Li Y, Hu Y, et al., (2023) Microstructural evolution and mechanical properties of 6Cr13 martensitic stainless steel subjected to cold rolling and heat treatments. *Mater Charact* 203: 113070. <https://doi.org/10.1038/s41524-021-00609-2>
14. Li H, Wang J, Xu Q, Dai L, Liao W, Yuan R, et al., (2023) High-strength medium-entropy alloy designed by precipitation-strengthening mechanism via machine learning. *Mater Sci Eng A* 882: 145443. <https://doi.org/10.1016/j.msea.2023.145443>
15. Sun Y, Lu Z, Liu X, Du Q, Xie H, Lv J, et al., (2021) Prediction of Ti-Zr-Nb-Ta high-entropy alloys with desirable hardness by combining machine learning and experimental data. *Appl Phys Lett* 119: 20. <https://doi.org/10.1063/5.0065303>

16. Qiao L, Lai Z, Liu Y, Bao A, Zhu J, (2021) Modelling and prediction of hardness in multi-component alloys: A combined machine learning, first principles and experimental study. *J Alloys Compd* 853: 156959. <https://www.sci-hub.vg/10.1016/j.jallcom.2020.156959>
17. Yang C, Ren C, Jia Y, Wang G, Li M, Lu W, (2022) A machine learning-based alloy design system to facilitate the rational design of high entropy alloys with enhanced hardness. *Acta Mater* 222: 117431. <https://doi.org/10.1016/j.actamat.2021.117431>
18. Feng X, Wang Z, Jiang L, Zhao F, Zhang Z, (2023) Simultaneous enhancement in mechanical and corrosion properties of Al-Mg-Si alloys using machine learning. *J Mater Sci Technol* 167: 1–13. <https://doi.org/10.1016/j.jmst.2023.04.072>
19. Zhang Y, Wen C, Wang C, Antonov S, Xue D, Bai Y, et al., (2020) Phase prediction in high entropy alloys with a rational selection of materials descriptors and machine learning models. *Acta Mater* 185: 528–539. <https://doi.org/10.1016/j.actamat.2019.11.067>
20. Li Z, Nash WT, O'Brien SP, Qiu Y, Gupta RK, Birbilis N, (2022) CardiGAN: A generative adversarial network model for design and discovery of multi principal element alloys. *J Mater Sci Technol* 125: 81–96. <https://doi.org/10.1016/j.jmst.2022.03.008>
21. Guo Q, Xu X, Pei X, Duan Z, Liaw PK, Hou H, et al., (2023) Predict the phase formation of high-entropy alloys by compositions. *J Mater Res Technol* 22: 3331–3339. <https://doi.org/10.1016/j.jmrt.2022.12.143>
22. Xue D, Balachandran PV, Yuan R, Hu T, Qian X, Dougherty ER, et al., (2016) Accelerated search for BaTiO<sub>3</sub>-based piezoelectrics with vertical morphotropic phase boundary using Bayesian learning. *Proc Natl Acad Sci* 113: 13301–13306. <https://doi.org/10.1073/pnas.1607412113>
23. Zhang Y, Wen C, Wang C, Antonov S, Xue D, Bai Y, et al., (2020) Phase prediction in high entropy alloys with a rational selection of materials descriptors and machine learning models. *Acta Mater* 185: 528–539. <https://doi.org/10.1016/j.actamat.2019.11.067>
24. Ramprasad R, Batra R, Pilania G, Mannodi-Kanakkithodi A, Kim C, (2017) Machine learning in materials informatics: Recent applications and prospects. *NPJ Comput Mater* 3: 54. <https://doi.org/10.1038/s41524-017-0056-5>
25. Ramakrishna S, Zhang TY, Lu WC, Qian Q, Low JSC, Yune JHR, et al., (2019) Materials informatics. *J Intell Manuf* 30: 2307–2326. <https://doi.org/10.1007/s10845-018-1392-0>
26. Rickman JM, Lookman T, Kalinin SV, (2019) Materials informatics: From the atomic-level to the continuum. *Acta Mater* 168: 473–510. <https://doi.org/10.1016/j.actamat.2019.01.051>
27. Li X, Zheng M, Yang X, Chen P, Ding W, (2022) A property-oriented design strategy of high-strength ductile RAFM steels based on machine learning. *Mater Sci Eng A* 840: 142891. <https://doi.org/10.1016/j.msea.2022.142891>
28. Jiang L, Zhang Z, Hu H, He X, Fu H, Xie J, (2023) A rapid and effective method for alloy materials design via sample data transfer machine learning. *NPJ Comput Mater* 9: 26. <https://doi.org/10.1038/s41524-023-00979-9>
29. Kim Y, Kim Y, Yang C, Park K, Gu GX, Ryu S, (2021) Deep learning framework for material design space exploration using active transfer learning and data augmentation. *NPJ Comput Mater* 7: 140. <https://doi.org/10.1038/s41524-021-00609-2>
30. Dempster A, Laird N, Rubin D, (1977) Maximum likelihood from incomplete data via the EM algorithm. *J R Stat Soc B Stat Methodol* 39: 1–38. <https://doi.org/10.1111/j.2517-6161.1977.tb01600.x>

31. Xue J, Shen B, (2020) A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst Sci Control Eng* 8: 22c34. <https://doi.org/10.1080/21642583.2019.1708830>
32. bin Mohd Zain MZ, Kanesan J, Chuah JH, Dhanapal S, Kendall G, (2018) A multi-objective particle swarm optimization algorithm based on dynamic boundary search for constrained optimization. *Appl Soft Comput* 70: 680–700. <https://doi.org/10.1016/j.asoc.2018.06.022>
33. Li G, Wang W, Zhang W, Wang Z, Tu H, You W, (2021) Grid search based multi-population particle swarm optimization algorithm for multimodal multi-objective optimization. *Swarm Evol Comput* 62: 100843. <https://doi.org/10.1016/j.swevo.2021.100843>
34. Coello CC, Lechuga MS, (2002) MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation*, 1051–1056. <https://doi.org/10.1109/CEC.2002.1004388>

## Appendix

### A.1. Gaussian mixture model generates data

GMM is a commonly used probabilistic model for modeling and clustering data. It is a mixed model consisting of multiple Gaussian distributions (also known as normal distributions). Each Gaussian distribution represents a subpopulation in the data, and the hybrid model adapts to the distribution of the data by weighting this subpopulation together. The basic principle of GMM is as follows.

**Gaussian distribution.** The Gaussian distribution is a continuous probability distribution parameterized by the mean  $\mu$  and the variance  $\sigma^2$ . Its probability dense function (PDF) can be used to describe a random variable with a symmetrical bell curve, where the one-dimensional Gaussian distribution is expressed as follows:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}. \quad (\text{A.1})$$

Let the D-dimensional random variable  $X = (X_1, X_2, \dots, X_D)^T$ , whose covariance matrix is  $\Sigma = \text{Cov}(X)$ , and the mathematical expectation vector  $\mu = (\mu_1, \mu_2, \dots, \mu_D)^T$ , then the high-dimensional Gaussian distribution describes the distribution of data with multiple parameters, including  $\mu$  and  $\Sigma$ . The expression is as follows:

$$f(x) = f(x_1, x_2, \dots, x_D) = \frac{1}{(2\pi)^{\frac{D}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right). \quad (\text{A.2})$$

**Gaussian mixture model.** The GMM combines multiple Gaussian distribution submodels into a hybrid model. Each Gaussian distribution has its own mean value ( $\mu$ ) and covariance matrix ( $\Sigma$ ), representing a component in the data. The parameters of the hybrid model include the weights of each components and the mean and covariance matrix of each subgroup. For a given data point, the GMM calculates the probability density function value for the Gaussian distribution of each subpopulation and sums the value weighted to obtain the PDF value of the entire mixed model, which can represent the probability that the data points belong to each subgroup. Assuming that  $z \in R^d$  is a sample with  $d$  features, if  $z$  comes from a mixed Gaussian model, then its probability density function can be expressed as follows:

$$p(z|\Omega) = \sum_{k=1}^K \omega_k f(z|\theta_k), \quad (\text{A.3})$$

where  $\Omega = \{\{\omega_1, \mu_1, \Sigma_1\}, \{\omega_2, \mu_2, \Sigma_2\}, \dots, \{\omega_K, \mu_K, \Sigma_K\}\}$  is the parameter of the GMM model with  $K$  components, which includes the Gaussian model parameter  $\theta_k = \{\mu_k, \Sigma_k\}$  and the mixing probability  $\omega_k (1 \leq k \leq K)$ .

## A.2. Model parameter estimation

Given  $N$  training samples  $Z = [z_1, z_2, \dots, z_N]$ , the likelihood and log-likelihood functions can be expressed as formulas (A.4) and (A.5). The goal of the GMM is to use the expectation maximization (EM) algorithm [30] to maximize the likelihood function according to Eq (A.6) and estimate the model parameters.

$$L(Z, \Omega) = \prod_{n=1}^N \sum_{k=1}^K \omega_k f(z_n | \theta_k), \quad (\text{A.4})$$

$$\log L(Z, \Omega) = \sum_{n=1}^N \sum_{k=1}^K \omega_k \log f(z_n | \theta_k), \quad (\text{A.5})$$

$$\hat{\Omega} = \arg \max_{\Omega} (\log L(Z, \Omega)). \quad (\text{A.6})$$

As an iterative optimization algorithm, the EM algorithm estimates parameters by alternating the two steps: the E step (Expectation) and the M step (Maximization).

E Step: In this step, the posterior probability that each data point belongs to each subgroup is calculated based on the current parameter estimation. These posterior probabilities, called implicit variables, thereby representing the probability that a data point belongs to each subgroup. The posterior probability that the  $j^{th}$  training sample belongs to the  $k^{th}$  Gaussian component at the  $m$  iteration is expressed as  $\gamma^{(m)}(c_k^j | z_j, \Omega^{(m)})$ , as shown in formula (A.7):

$$\gamma^{(m)}(c_k^j | z_j, \Omega^{(m)}) = \frac{\omega_k^{(m)} f(z_j | \mu_k^{(m)}, \Sigma_k^{(m)})}{\sum_{i=1}^K \omega_i^{(m)} f(z_j | \mu_i^{(m)}, \Sigma_i^{(m)})}. \quad (\text{A.7})$$

M step: In the M step, the model parameters are updated using the calculated hidden variables. In the  $m + 1$  iteration, the probability weight  $\omega_k^{(m+1)}$ , mean value  $\mu_k^{(m+1)}$ , and covariance  $\Sigma_k^{m+1}$  of the  $k^{th}$  gaussian component formula are as follows:

$$\omega_k^{(m+1)} = \frac{\sum_{j=1}^N \gamma^{(m)}(c_k^j | z_j, \Omega^{(m)})}{N}, \quad (\text{A.8})$$

$$\mu_k^{(m+1)} = \frac{\sum_{j=1}^N \gamma^{(m)}(c_k^j | z_j, \Omega^{(m)}) z_j}{\sum_{j=1}^N \gamma^{(m)}(c_k^j | z_j, \Omega^{(m)})}, \quad (\text{A.9})$$

$$\Sigma_k^{m+1} = \frac{\sum_{j=1}^N \gamma^{(m)}(c_k^j | z_j, \Omega^{(m)}) (z_j - \mu_k^{(m+1)})(z_j - \mu_k^{(m+1)})^T}{\sum_{j=1}^N \gamma^{(m)}(c_k^j | z_j, \Omega^{(m)})}. \quad (\text{A.10})$$

Repeat steps E and M until the model converges (parameters no longer change) or a predetermined number of iterations is reached.

### A.3. Sparrow search algorithm (SSA)

Based on enhanced data, several ML models are built using virtual samples to predict the tensile strength and elongation of the alloy, including RF, SVM, neural network, XGBoost, GBR, Catboost, ridge regression, lasso regression, and Elastic network. After ten-fold cross-validation evaluation and visualization of the prediction results, the Catboost, XGBoost and GBR models with good performances are selected, and the SSA is used to optimize these three models.

SSA [31] is a heuristic optimization algorithm based on sparrow foraging behavior. It simulates the behavioral strategies of sparrows when searching for food by foraging, discovering, and sharing information to iteratively search for potential solutions in the solution space. Through the interaction of position and fitness information between individuals, the algorithm gradually converges to the global optimal solution during the search process. The finder and follower are two important roles in the SSA, and the finder simulates the leader in the sparrow population, who searches for potential solutions by constantly exploring new search spaces. Followers, in turn, mimic ordinary members of the sparrow colony, thereby observing the finder's behavior and conducting their own searches based on the information they find. This formula can be achieved through foraging behavior, information sharing, and awareness of danger. In the SSA, the position update formula is used to update the sparrow's position in the search space. The specific formula is as follows.

The position of the discoverer  $X_{ij}^{t+1}$  is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^t \cdot \exp\left(\frac{-i}{\alpha \cdot iter_{max}}\right) & \text{if } R_2 < ST \\ X_{i,j}^t + Q \cdot L & \text{if } R_2 \geq ST \end{cases} \quad (\text{A.11})$$

The position of the follower  $X_{ij}^{t+1}$  is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_{worst}^t - X_{ij}^t}{i^2}\right) & \text{if } i > \frac{n}{2} \\ X_p^{t+1} + |X_{i,j}^t - X_p^{t+1}| \cdot A^+ \cdot L & \text{otherwise} \end{cases} \quad (\text{A.12})$$

The position of the aware sparrow is updated as follows:

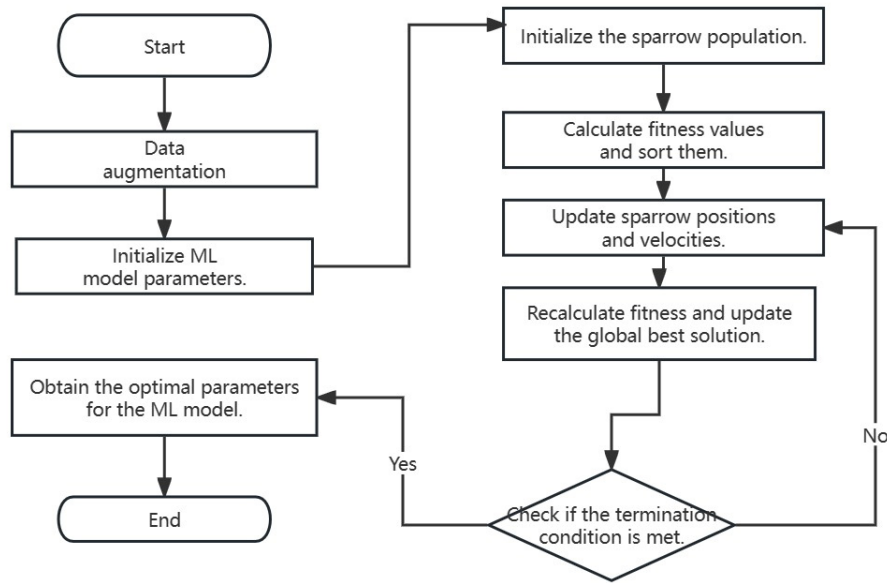
$$X_{i,j}^{t+1} = \begin{cases} X_{best}^t + \beta \cdot |X_{i,j}^t - X_{best}^t| & \text{if } f_i > f_g \\ X_{i,j}^t + K \cdot \left(\frac{|X_{i,j}^t - X_{worst}^t|}{(f_i - f_w) + \varepsilon}\right) & \text{if } f_i = f_g \end{cases} \quad (\text{A.13})$$

**Table A.1.** The meanings of symbols involved in formulas (A.9) to (A.11).

| Symbol      | Meaning                                                                         | Symbol        | Meaning                                                  |
|-------------|---------------------------------------------------------------------------------|---------------|----------------------------------------------------------|
| $X_{i,j}^t$ | The position information of the i sparrow in the j dimension in the t iteration | $iter_{max}$  | Maximum iterations                                       |
| $\alpha$    | Random number of [0,1]                                                          | $R_2$         | The random number of [0,1], indicating the warning value |
| $ST$        | A constant of [0.5,1], indicating a safe value                                  | $Q$           | Random numbers that follow a positive distribution       |
| $L$         | A 1*d matrix with element 1                                                     | $X_{worst}^t$ | The worst individual in the t iteration                  |

*Continued on next page*

| Symbol | Meaning                                                                                                                                               | Symbol  | Meaning                                                                          |
|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------|---------|----------------------------------------------------------------------------------|
| $X_p$  | The location of the current best discoverer                                                                                                           | $A$     | A 1*d matrix with randomly assigned elements of 1 or -1                          |
| $K$    | The random number of [-1,1], positive or negative indicates the direction of sparrow movement, and the size indicates the step size control parameter | $\beta$ | Follow a positive distributed random number with a mean of 0 and a variance of 1 |
| $A^+$  | $A^T(AA^T)^{-1}$                                                                                                                                      | $f_i$   | Fitness value of the current individual                                          |
| $f_g$  | Current maximum fitness                                                                                                                               | $f_w$   | The current minimum fitness value                                                |



**Figure A.1.** Flowchart of the SSA for optimizing model parameters.

## A.4. Multi-objective optimization

### A.4.1. Description of the multi-objective problem

The multi-objective optimization problem involves finding a set of solutions in the case of multiple conflicting objectives, so that the optimal tradeoff results can be obtained in each objective function. In this kind of problem, improving one goal can have an adverse effect on the others, so a set of solutions needs to be found so that no other solution can simultaneously achieve better results on all goals. The mathematical form of a multi-objective optimization problem can be described as follows:

$$\min y = F(X) = [f_1(X), f_2(X), \dots, f_n(X)], \quad (\text{A.14})$$

$$n = 1, 2, \dots, N, \quad (\text{A.15})$$

$$s. t. g_i(x) \leq 0 \quad i = 1, 2, \dots, m, \quad (\text{A.16})$$

$$h_j(x) = 0 \quad j = 1, 2, \dots, k, \quad (\text{A.17})$$

$$X = [x_1, x_2, \dots, x_d, \dots, x_D], \quad (\text{A.18})$$

$$x_{d\_min} \leq x_d \leq x_{d\_max} \quad d = 1, 2, \dots, D, \quad (A.19)$$

where  $X$  is the  $D$ -dimensional decision vector,  $y$  is the target vector, and  $N$  is the total number of optimization targets.  $g_i(x) \leq 0$  is the  $i^{th}$  inequality constraint.  $h_j(x) = 0$  is the  $j^{th}$  equality constraint.  $f_n(x)$  is the  $n^{th}$  objective function. The space composed of all possible decision vectors is the decision space, denoted  $D = \{X \in R^n \mid x_{d\_min} \leq x_d \leq x_{d\_max}, d = 1, 2, \dots, n\}$ .  $Y = \{y = F(X) \mid X \in D\}$  is called the target space.  $g_i(x) \leq 0$  and  $h_j(x) = 0$  determine the known feasible domains, where  $x_{d\_min}$  and  $x_{d\_max}$  are the upper and lower limits of the dimensions of the decision vector.

The optimal solution or non-inferior optimal solution in the multi-objective optimization problem can be defined as follows.

**Dominant relationship.** For any  $d \in [1, D]$ , if  $x_d^* \leq x_d$  and there exists  $d_0 \in [1, D]$  such that  $x_{d_0}^* < x_{d_0}$ , then vector  $X^* = [x_1^*, x_2^*, \dots, x_D^*]$  dominates vector  $X = [x_1, x_2, \dots, x_D]$ .

The dominance of  $f(X^*)$  over  $f(X)$  must satisfy the following two conditions:

$$\forall n, f_n(X^*) \leq f_n(X) \quad n = 1, 2, \dots, N, \quad (A.20)$$

$$\exists n_0, f_{n_0}(X^*) < f_{n_0}(X) \quad 1 \leq n_0 \leq N. \quad (A.21)$$

The dominance relationship of  $f(X)$  is consistent with the dominance relationship of  $X$ .

The Pareto optimal solution [32] is a solution that is not dominated by any feasible solution: if  $X^*$  is a point in the search space, then it is called a non-dominated optimal solution if and only if there does not exist  $X$  (in the feasible region of the search space) such that  $f_n(X) \leq f_n(X^*)$  holds for all  $n = 1, 2, \dots, N$ .

#### A.4.2. Multi-objective particle swarm optimization algorithm

The particle swarm optimization (PSO) algorithm was proposed in 1995 by Kennedy and Eberhart [33], which is a random search algorithm based on swarm intelligence. The PSO algorithm seeks the optimal solution by simulating the migration and swarm behavior of birds during foraging and iterative evolution. In PSO, each particle can be seen as a search entity in a  $D$ -dimensional search space with two properties: velocity and position. The velocity and position of the  $i^{th}$  particle are defined as  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$  and  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ , respectively, and the current position of the particle in the object space is a candidate solution for the corresponding optimization problem.

In addition, during evolution, each particle in the population updates its own speed and position according to the individual historical optimal position ( $p_i(t)$ ) and the global optimal position ( $p_g(t)$ ). When the  $i^{th}$  particle of generation  $t$  evolves to generation  $t+1$ , it is updated according to the following formula:

$$v_i(t+1) = w \cdot v_i(t) + c_1 \cdot r_1(t)[p_i(t) - x_i(t)] + c_2 r_2(t)[p_g(t) - x_i(t)], \quad (A.22)$$

$$x_i(t+1) = x_i(t) + v_i(t+1), \quad (A.23)$$

where,  $w$  is called the inertia weight, which can dynamically adjust the inertia of particles in flight to achieve the purpose of balancing the local search capability and the global search capability.  $c_1$  and  $c_2$  are learning factors, which are used to regulate the particle velocity;  $r_1$  and  $r_2$  are random numbers

in the interval  $[0, 1]$ , which are used to increase the randomness of the algorithm.

Multi-objective particle swarm optimization (MOPSO) was proposed in 2002 by Coello et al. [34], which introduced a Pareto dominance relationship into PSO to deal with problems with multiple objective functions. The MOPSO algorithm follows the framework of the traditional PSO algorithm. In addition, the external archive is used to store the non-dominated solutions found in each iteration search process, and the non-dominated solutions in the external archive are updated and maintained based on the grid and the density. Different from the traditional PSO algorithm, MOPSO selects the appropriate non-dominant solution for the population particles from the external archive as gbest to guide the flight of the particles, so as to find a set of optimal solutions that are evenly distributed as far as possible and close to the real Pareto optimal frontier.

The steps of the multi-objective particle swarm optimization algorithm are described as follows:

|                                                                                                    |
|----------------------------------------------------------------------------------------------------|
| Set the algorithm parameters and initializes the position and velocity of particles                |
| Initialize individual optimality (pbest) and Global optimality (gbest) for each particle           |
| Each particle is evaluated                                                                         |
| Calculate the target value for each particle                                                       |
| Set the current position of each particle to the individual optimal position pbest                 |
| Save the non-dominated solution in an external archive                                             |
| Repeat the specified number of iterations:                                                         |
| For each particle in a particle swarm:                                                             |
| Update speed and position, using particle swarm optimization formula 28.29                         |
| Limit the speed to ensure that it is within a predetermined boundary                               |
| Update Current location = Current location + speed                                                 |
| Calculates the target value for the new location                                                   |
| Update individual best (pbest) if the new location is better                                       |
| Update Global Best (gbest) if the new location is better                                           |
| For each particle in a particle swarm:                                                             |
| Pareto dominance relationship and crowding distance are calculated based on the current population |
| Update the file based on Pareto dominance relationship and congestion distance                     |
| Returns file                                                                                       |

**Table A.2.**  $R^2$  in the the train and test sets.

| Model    | Train     |          | Test      |          |
|----------|-----------|----------|-----------|----------|
|          | Rm_ $R^2$ | A_ $R^2$ | Rm_ $R^2$ | A_ $R^2$ |
| RF       | 0.959     | 0.992    | 0.929     | 0.984    |
| SVR      | 0.907     | 0.989    | 0.924     | 0.972    |
| XGboost  | 0.961     | 0.994    | 0.914     | 0.988    |
| GBR      | 0.964     | 0.989    | 0.931     | 0.983    |
| Catboost | 0.906     | 0.982    | 0.907     | 0.984    |

**Table A.3.** Hyperparameters of different ML models for the tensile strength RM.

|                |                  |                      |                      |                    |                   |
|----------------|------------------|----------------------|----------------------|--------------------|-------------------|
| RF_Rm          | Max_depth = 7    | Min_samples_leaf = 2 | N_estimators = 90    | Max_features = 0.7 | Random_state = 36 |
| SVR_Rm         | Kernel = rbf     | C = 100              | Gamma = 0.0001       | Epsilon = 0.02     |                   |
| ANN_Rm         | Epochs = 600     | Batch_size = 30      | Learning rate = 0.01 |                    |                   |
| XGBoost_Rm     | Max_depth = 5    | Learning rate = 0.5  | N_estimators = 8     | Random_state = 6   |                   |
| GBR_Rm         | Max_depth = 6    | Learning rate = 0.02 | N_estimators = 150   |                    |                   |
| CatBoost_Rm    | Iterations = 500 | Learning rate = 0.2  | Depth = 6            |                    |                   |
| Ridge_Rm       | Alpha = 2.0      |                      |                      |                    |                   |
| Lasso_Rm       | Alpha = 1.0      |                      |                      |                    |                   |
| Elastic_Net_Rm | Alpha = 0.07     | L1_ratio = 0.5       |                      |                    |                   |

**Table A.4.** Hyperparameters of different ML models for elongation A%.

|               |                  |                       |                      |                    |                   |
|---------------|------------------|-----------------------|----------------------|--------------------|-------------------|
| RF_A          | Max_depth = 7    | Min_samples_leaf = 2  | N_estimators = 90    | Max_features = 0.7 | Random_state = 36 |
| SVR_A         | Kernel = rbf     | C = 100               | Gamma = 0.001        | Epsilon = 0.02     |                   |
| ANN_A         | Epochs = 1000    | Batch_size = 30       | Learning rate = 0.01 |                    |                   |
| XGBoost_A     | Max_depth = 5    | Learning rate = 0.56  | N_estimators = 8     | Random_state = 6   |                   |
| GBR_A         | Max_depth = 6    | Learning rate = 0.017 | N_estimators = 150   |                    |                   |
| CatBoost_A    | Iterations = 500 | Learning rate = 0.2   | Depth = 6            |                    |                   |
| Ridge_A       | Alpha = 2.0      |                       |                      |                    |                   |
| Lasso_A       | Alpha = 1.0      |                       |                      |                    |                   |
| Elastic_Net_A | Alpha = 0.07     | L1_ratio = 0.5        |                      |                    |                   |



AIMS Press

© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0>)