**Problem Chosen**
B

**2024**
**ShuWei Cup**
**Summary Sheet**

**Team Control Number**
2024101029505

# Research on the Collaborative Estimation Method for Spatial Variables

## Summary

Based on the complexity and diversity of spatial data, this paper focuses on the issue of spatial variable estimation accuracy and employs an optimized Kriging-Machine Learning estimation method using intelligent optimization algorithms.

**For problem 1**,the data is first preprocessed and organized into a 266×266 format. Subsequently, three-dimensional surface plots and pseudo-color maps are generated for each target variable and auxiliary variable in Appendices 1 and 2. Next, a random uniform resampling model and a Kriging interpolation model based on the Particle Swarm Optimization (PSO) algorithm are established. Interpolation is performed using datasets with different sampling rates, and evaluation metrics such as RSS, MAE, MSE, RMSE, and $R^2$ are utilized to construct an evaluation model. The results are then visualized using line graphs. The findings indicate that as the sampling rate increases, the model's goodness-of-fit improves and the error decreases. The Kriging interpolation model exhibits excellent performance when the sampling rate is 70% or higher.

**For problem 2**,this paper establishes a decision tree model based on a genetic algorithm with elitist preservation strategy, and then calculates the Pearson correlation scores between features and the target variable. Combined with this model, the correlation results are ranked in descending order: variable1, variable3, variable2, and variable4. Finally, the correlation results are visualized.

**For problem 3**,three models are established for target variable prediction: a hybrid Kriging-MLP neural network model, a hybrid Kriging-Random Forest algorithm, and a nonlinear regression model based on the Differential Evolution algorithm. Experiments are conducted using datasets with different sampling rates, and the results indicate that the hybrid Kriging-MLP neural network model performs best across various error metrics, making it the optimal choice for predicting the target variable.

**For problem 4**,the Kriging-MLP model is employed to perform interpolation estimation on the F2 target variable, and the results are visualized to showcase the trend characteristics of the F2 target variable.

During the model testing phase, cross-validation methods are used to evaluate model performance, further validating the effectiveness of the hybrid Kriging-MLP neural network model. Sensitivity analysis reveals that sampling rate and model parameters have significant impacts on model performance.

In summary, this paper optimizes the variogram function using intelligent optimization algorithms to obtain the optimal parameters for the Kriging variogram, thereby improving computational efficiency and reducing Kriging interpolation errors. By establishing multiple models and comparing their results through experimental methods, it is ultimately concluded that the intelligent optimization algorithm-optimized Kriging-MLP model achieves the best performance.

**Keywords: Intelligent optimization algorithm Kriging interpolation Machine Learning Hybrid Kriging-MLP neural network**

# Contents

# 1. Introduction

## 1.1 Background

Spatial data often exhibit complexity and diversity. For instance, in the geospatial environment, various natural phenomena and socio-economic activities are influenced by spatial locations, resulting in data that frequently possess spatial correlation. In practical applications, there are high requirements for the estimation accuracy of spatial variables. Due to the complexity and diversity of spatial data, single-variable estimation methods often fail to achieve the desired accuracy. Therefore, it is necessary to adopt co-estimation methods that fully leverage the spatial correlation among multiple variables to enhance estimation accuracy. The cokriging method is a commonly used co-estimation approach for spatial variables. This method categorizes geological data for spatial estimation into primary variables and secondary variables, and cokriging utilizes the complementary information from these two types of variables to improve estimation accuracy, achieving good results in practical applications.

## 1.2 Work

**Problem 1:** Using the data from Appendix 1, investigate the variation pattern of the target variable (F1). This includes randomly and uniformly resampling the target variable, estimating values at unsampled locations, and exploring the relationship between sample size and estimation error.

**Problem 2:** Analyze the correlation between the target variable and other covariables in Appendix 1, and select two covariables as estimation covariables for the target variable.

**Problem 3:** Based on the analysis in Problem 2, select one or two covariables to study the variation pattern of the target variable (F1). Conduct random and uniform resampling, estimate spatial variable values at unsampled locations, and compare different methods.

**Problem 4:** For the target variable (F2) in Appendix 2, due to insufficient sampled data, choose the optimal method from Problem 3 to estimate the trend of the target variable and present the results.

# 2. Problem analysis

## 2.1 Data analysis

### 2.1.1 Data cleaning

First, the data in Attachment 1 and Attachment 2 was cleaned to detect any possible outliers in the data. After the data was cleaned, no outliers or missing values were found in Attachment 1 or Attachment 2.

### 2.1.2 Data Visualization Analysis

Next, the data was organized, and since the data provided in Attachment 1 and Attachment 2 was not in the form of 266x266, we organized it. In order to visually present the distribution and trend of the data, we performed visualization processing on the organized data. The distribution and trend of the F1 target variable are shown in Figure 1 below:

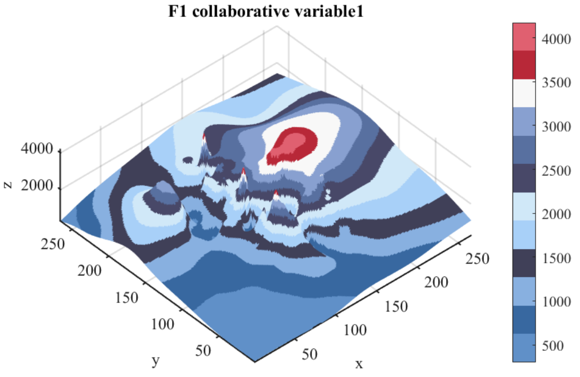


Figure 1: 3D contour plot of F1 target variable

In Figure 1, we can see that the F1 target variable has multiple local extremum points in the three-dimensional space, which are visually represented by the depth of color. In addition, the density of contour lines also reflects the trend of the F1 target variable value in space. The dense areas indicate that the variable value changes rapidly, while the sparse areas indicate a more gradual change. The overall trend is roughly an increase from bottom to top.

Next, to enhance the details and contrast of the data, we used a pseudocolor map to further display the details of the F1 target variable.

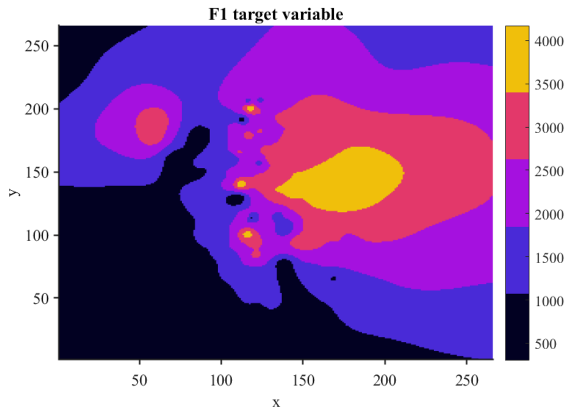


Figure 2: Pseudocolor map of F1 target variable

Looking at Figure 2, we can see that the graph has four extremum regions (yellow areas), one of which is the largest and is located towards the right side of the graph, while the other three are smaller and are generally located towards the left side of the center of the graph.

Figure 3 below shows the distribution and trend of the other four cooperative variables:

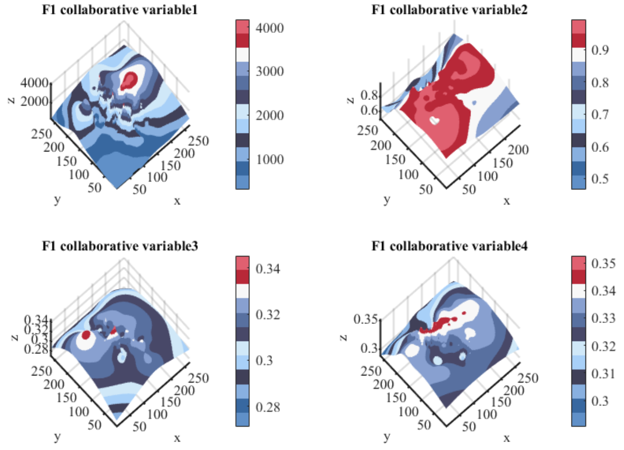


Figure 3: shows a 3D surface plot of collaborative variable1 to collaborative variable4 for F1

Collaborative variable1 exhibits a complex distribution pattern, with high values (red) mainly concentrated in the upper left and lower right corners, and low values (blue) scattered in other areas, showing a clear non-uniformity.

Collaborative variable2 shows a clear partitioning pattern, with high-value regions (red) clearly leaning to the right side of the graph, and low-value regions (blue) concentrated on the left side, showing that this variable has significant differences between two different regions.

Collaborative variable3 has a relatively smooth distribution, with uniform color gradation, with high values (red) concentrated in the upper left corner and low values (blue) located in the lower right corner, showing the continuous change characteristics of this variable.

Collaborative variable4 has a more complex distribution than other variables, with high-value (red) and low-value (blue) regions interlaced, with red regions mainly concentrated in the upper right and lower left corners, and blue regions distributed on the left and lower right corners of the graph, showing that this variable has complex interactions between different regions.

Attachment 1 shows the details and contrast of the other 4 collaborative variables as follows:

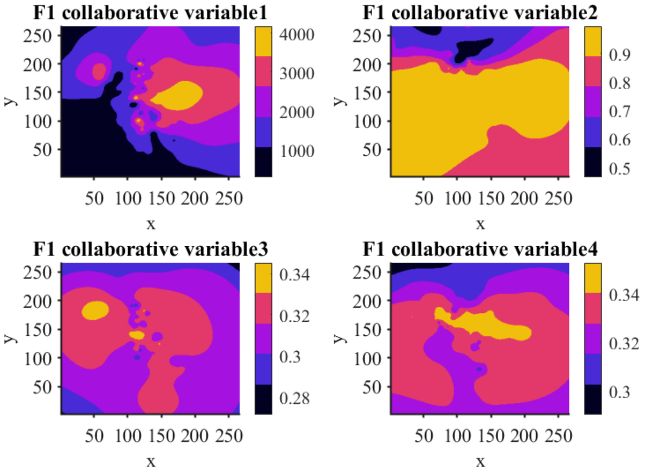


Figure 4: Trend and Distribution Exhibition of F2 Collaborative Variable1 to Collaborative Variable4

In the collaborative variable 1 of F1, the numerical range is from 0 to 4000, and the colors gradually change from black (representing low values) to yellow (representing high values). It can be observed that the left and middle parts of the figure are densely populated with high-value points (yellow areas), while the right and bottom are mainly low-value points (black areas). This distribution reveals the concentration trend of data in different regions.

The collaborative variable 2 map of F1 shows the distribution of numerical values ranging from 0.5 to 0.9. Most of the areas in the figure display yellow, indicating that these areas have higher values on average; while the black areas are less, mainly concentrated in some edge positions of the figure, which shows that the high-value data accounts for a larger proportion, while the low-value data is relatively sparse.

For the collaborative variables 3 and 4 of F1, the color bars represent the numerical ranges of 0.28 to 0.34 and 0.3 to 0.34, respectively. In both figures, the colors gradually change from blue (low values) to yellow (high values). It is worth noting that most of the areas in the figures display red and pink, indicating that these areas have relatively higher values; while the yellow areas are relatively sparse, located in the upper left corner, lower right corner, and middle part of the figures, these yellow areas represent the highest value points in the data.

Similarly, the distribution and trend of the data of the four auxiliary variables in Attachment 2 are shown in Figure 5 below:

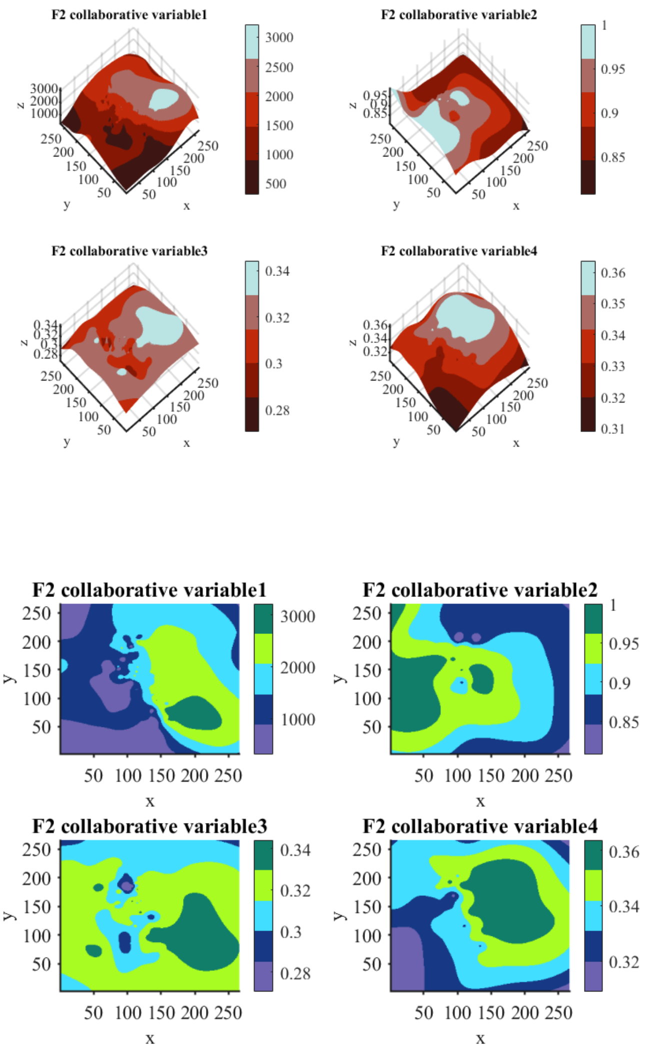


Figure 5: Is a pseudocolor map of the collaborative variables 1-4 in F2

In the upper half, the three-dimensional plots of each variable intuitively depict their distribution states in three-dimensional space. The three-dimensional plots of F2 Collaborative Variable1 and F2 Collaborative Variable2 utilize a red-to-blue gradient color bar, where red areas represent high values and blue areas represent low values, showing that the high-value regions of these two variables are relatively concentrated. In contrast, the three-dimensional plots of F2 Collaborative Variable3 and F2 Collaborative Variable4 adopt a purple-to-blue gradient color, with their high-value regions appearing more dispersed, indicating a broader distribution for these two variables.

The lower half further exhibits the distribution of these variables on a two-dimensional plane through two-dimensional plots. Each variable's two-dimensional plot also employs a color gradient from dark blue to light green, clearly presenting the distribution characteristics of different value ranges. These two-dimensional plots not only echo the three-dimensional plots in the upper half but also provide a more concise and intuitive perspective, facilitating the observation and analysis of the specific distribution of the variables.

## 2.2 Analysis of question 1

Problem one requires studying the variation pattern of the spatial variable (F1_target) in Annex One. We will first perform random, uniform resampling of F1_target, selecting spatially even samples from the original data. Then, we'll use these resampled values to estimate the variable at unsampled locations, often employing spatial interpolation techniques like Kriging. The results will be visually presented as contour maps to clearly show the distribution and trends of the spatial variable.

Secondly, we need to vary the sample size to explore the relationship between sample size and estimation error. By increasing or decreasing the number of samples and repeating the estimation process, we can observe how the estimation error changes with the sample size. This helps us understand the sample size required to achieve acceptable estimation accuracy, thereby providing guidance for actual sampling strategies.

## 2.3 Analysis of question 2

Problem two requires studying the correlation between the target variable and auxiliary variables, and selecting two covariables as the covariables for the target variable. Firstly, we need to conduct a correlation analysis on all variables in Appendix 1, which can be achieved by calculating correlation coefficients such as Pearson's correlation coefficient. Through correlation analysis, we can identify auxiliary variables that are highly correlated with the target variable (F1_target).

Next, from these highly correlated auxiliary variables, we need to select two as covariables. The selection criteria can include the strength of their correlation with the target variable, their spatial distribution characteristics, and their ease of measurement and acquisition, among other factors. After selecting the covariables, we will use them together with the target variable in subsequent spatial estimation and cokriging studies.

## 2.4 Analysis of question 3

Problem three requires utilizing the selected spatial or covariable(s) to study the variation pattern of the target variable (F1_target). Firstly, we need to perform random and uniform resampling of the target and covariable(s), and use these resampled values to estimate the spatial variable values at unsampled locations. Similar to Problem one, we will present the estimation results in the form of contour maps to visually observe the distribution and trends of the spatial variables.

Secondly, we need to vary the sample size and explore the relationship between sample size and estimation error. This helps us understand the accuracy and reliability of cokriging estimates at different sample sizes. Finally, we need to select at least two methods for cokriging estimation and compare their results. This can assist us in evaluating the strengths and weaknesses of different methods, thereby selecting the most suitable estimation method for the current data and problem.

## 2.5 Analysis of question 4

Problem four requires using the optimal method selected in Problem three to estimate the trend of the target variable (F2_target) in Appendix two, and to present the results as a contour map. Firstly, we need to preprocess the data in Appendix two, including data cleaning and handling of missing values. Next, we will apply the optimal cokriging estimation method determined in Problem three, along with the selected covariables, to spatially estimate F2_target.

During the estimation process, we should be mindful that the sampling data for the target variable in Appendix two may be insufficient. Therefore, it may be necessary to leverage additional auxiliary information or employ more complex estimation techniques to enhance the accuracy and reliability of the estimates. Finally, we will present the estimation results in the form of a contour map, allowing for a visual observation of the spatial distribution and trends of F2_target, and providing valuable support for subsequent decision-making and analysis.

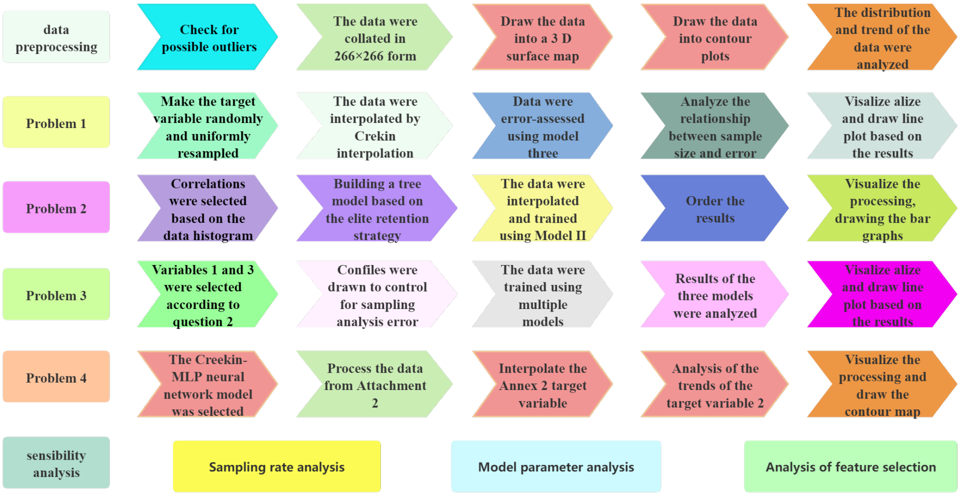

data preprocessing
Check for possible outliers
The data were collated in 266×266 form
Draw the data into a 3 D surface map
Draw the data into contour plots
The distribution and trend of the data were analyzed
Problem 1
Make the target variable randomly and uniformly resampled
The data were interpolated by Crekin interpolation
Data were error-assessed using model three
Analyze the relationship between sample size and error
Visalize alize and draw line plot based on the results
Problem 2
Correlations were selected based on the data histogram
Building a tree model based on the elite retention strategy
The data were interpolated and trained using Model II
Order the results
Visualize the processing, drawing the bar graphs
Problem 3
Variables 1 and 3 were selected according to question 2
Confiles were drawn to control for sampling analysis error
The data were trained using multiple models
Results of the three models were analyzed
Visalize alize and draw line plot based on the results
Problem 4
The Creekin-MLP neural network model was selected
Process the data from Attachment 2
Interpolate the Annex 2 target variable
Analysis of the trends of the target variable 2
Visualize the processing and draw the contour map
sensibility analysis
Sampling rate analysis
Model parameter analysis
Analysis of feature selection

Figure 6: The overall flowchart

# 3. Symbol and Assumptions

## 3.1 Symbol Description

| Serial Number | Symbol | Explanation |
|---|---|---|
| 1 | MAE | Mean Absolute Error |
| 2 | MSE | Mean Squared Error |
| 3 | RMSE | Root Mean Squared Error |
| 4 | $R^2$ | Coefficient of Determination |
| 5 | RSS | Sum of Squared Residuals |
| 6 | $\hat{Z}(u)$ | the estimated value for the prediction location $u$ |
| 7 | $Z(u_i)$ | the measured value of the $i$ sample point |
| 8 | $\lambda_i$ | the weight value of the $i$ sample point |
| 9 | $\gamma$ | the variogram |
| 10 | $w$ | the weight |
| 11 | $c_1$ ,$c_2$ | acceleration constants |
| 11 | $r_1$ , $r_2$ | random factors between [0,1] |
| 13 | $v_j(k)$ | the velocity and position of the $j$ particle at time $k$ |
| 14 | $T_j$ | the nodes that utilize feature |
| 15 | $Impurity$ | a purity metric |

## 3.2 Fundamental assumptions

Spatial variables exhibit values that are interrelated in space, displaying a characteristic of dependence on spatial structure, where values at neighboring sampling points are often closely correlated. This premise is rooted in the principles of spatial statistics, emphasizing the existence of non-independence and spatial autocorrelation among sampling points.

Despite differences arising from using various methods to measure the same physical quantity, these measurements still retain a certain pattern of correlation in space. This implies that, even with differing measurement techniques, the spatial correlation structure among the data can still be identified and effectively utilized through cokriging techniques to achieve data integration.

Furthermore, selecting covariables (i.e., other spatial variables) that are correlated with the target variable, which have linear or nonlinear relationships with the target variable, greatly benefits the accurate estimation of the spatial distribution of the target variable. By carefully selecting covariables, the accuracy of spatial predictions for the target variable can be significantly enhanced.

# 4. Model

## 4.1 Question 1: Model Establishment

**Model 1: Random Uniform Resampling Model**

The Random Uniform Resampling Model is a technique commonly used in statistics and machine learning. It generates a new dataset by randomly and uniformly sampling from the original dataset of spatial variables, enabling collaborative estimation of these spatial variables.

**Step1:** Begin by obtaining the data dimensions.

**Step2:** Use size(data) to obtain the number of rows and columns in the data matrix data, storing them in rows and cols, respectively.

**Step3:** Calculate the proportion of data elements to be retained and then determine the total number of elements to be set to NaN.

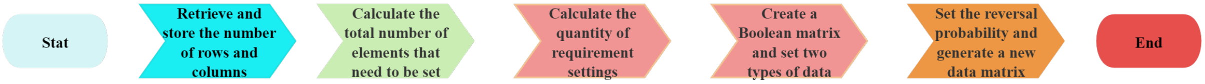
**Step4:** Calculate the approximate number of NaNs to be set per row and determine the remaining number of elements to be set to NaN.

**Step5:** Create a boolean matrix of the same size as data, initially setting all elements to True, indicating that these positions will be set to NaN.

**Step6:** Randomly select the positions of data to be retained within each row. For each row, set a reversal probability and set the corresponding positions in the original data matrix data that are True to NaN, generating a new data matrix.

**Step7:** End by achieving data sampling.

The process flow of its model is illustrated in the figure below:


Stat
Retrieve and store the number of rows and columns
Calculate the total number of elements that need to be set
Calculate the quantity of requirement settings
Create a Boolean matrix and set two types of data
Set the reversal probability and generate a new data matrix
End

**Model 2: Kriging Interpolation Model Based on Particle Swarm Optimization Algorithm**

Ordinary Kriging Algorithm: The Kriging algorithm is currently the most widely used spatial interpolation method. It is an interpolation method for unbiased optimal estimation of regional variation values within a limited area based on structural analysis and variogram theory. Its basic principle is as follows:

$$\begin{cases} \hat{Z}(u) = \sum_{i=1}^{n} \lambda_i Z(u_i) \\ \sum_{i=1}^{n} \lambda_i = 1 \end{cases} \tag{1}$$

In the equation,$\hat{Z}(u)$ is the estimated value for the prediction location $u$, and $Z(u_i)$ is the measured value of the $i$ sample point. $\lambda_i$denotes the weight value of the $i$ sample point, and $n$ represents the total number of sample points. The sample semivariogram value is:

$$R(h) = \frac{1}{n}N(h)\sum_i \left[z(x_i) - z(x_i + h)\right]^2 \tag{2}$$

From Equation 1, it can be observed that the value of the prediction point is estimated through the weighted average of the surrounding measured points. Therefore, the determination of weights is crucial. The calculation of weights relies on the variogram, which reflects the spatial correlation between different points, as shown in Equation 2.

$$\begin{bmatrix} \gamma(x_1 - x_1) & \cdots & \gamma(x_1 - x_n) \\ \gamma(x_2 - x_1) & \cdots & \gamma(x_2 - x_n) \\ \vdots & \ddots & \vdots \\ \gamma(x_n - x_1) & \cdots & \gamma(x_n - x_n) \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \vdots \\ \gamma_n \end{bmatrix} = \begin{bmatrix} \gamma(x_1 - x_0) \\ \vdots \\ \gamma(x_n - x_0) \end{bmatrix} \tag{3}$$

In the equation, $\gamma$ represents the variogram; $x_0$ denotes the location of the prediction point. It can be seen that in the Kriging algorithm, the variogram is a core component. The variogram is used to calculate the covariance or semivariance between each pair of points, which describes the degree of variation between any two points in spatial data. The choice of the variogram affects the accuracy and reliability of the Kriging interpolation results. Currently, commonly used variogram models include the exponential model, Gaussian model, and spherical model. However, regardless of the type of variogram, it involves three fitting parameters: the nugget value ($C_0$), the partial sill ($C$), and the range ($a$). Currently, the determination of these three parameters mainly relies on empirical judgment and trial-and-error methods, leading to a significant subjective influence in constructing the variogram model. This may result in the selected model not being optimal, thereby affecting the accuracy of the interpolation results.

Particle Swarm Optimization (PSO): Particle Swarm Optimization is a stochastic and parallel optimization algorithm. To prevent the algorithm from falling into a local optimum, weights are added to the basic Particle Swarm Optimization for improvement and optimization, thereby enhancing its search capability for various problems. The updated formula for the improved Particle Swarm Optimization is as follows:

$$v_j(k+1) = wv_j(k) + c_1r_1\left(pt_j(k) - x_j(k)\right) + c_2r_2\left(gt(k) - x_j(k)\right) \tag{4}$$

$$x_j(k+1) = x_j(k) + v_j(k+1) \tag{5}$$

In the equation, $w$ represents the weight; $c_1$ and $c_2$ are acceleration constants; $r_1$ and $r_2$ denote random factors between [0,1]; $pt$ is the individual best position;$gt$ is the global best position; and $v_j(k)$,$x_j(k)$, and $v_j(k+1)$,$x_j(k+1)$ respectively represent the velocity and position of the $j$ particle at time $k$ and $k+1$.

Optimizing the Kriging Interpolation Algorithm: To address the issue of subjectivity in selecting the parameters of the variogram in traditional Kriging interpolation algorithms, this paper adopts the Gaussian model as the variogram and determines the three parameters of the variogram model—the nugget value($C_0$), the partial sill ($C$), and the range ($a$)—based on the characteristic of the Particle Swarm Optimization (PSO) algorithm, which can quickly find the global optimal solution. The specific process is shown in Figure 7.

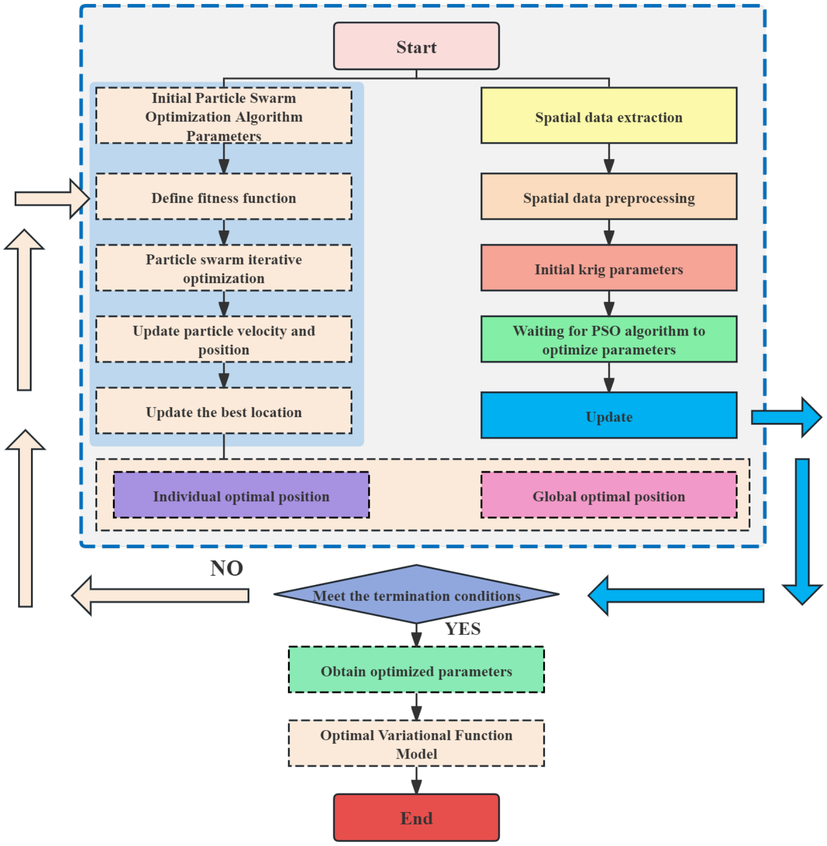

Start
Initial Particle Swarm Optimization Algorithm Parameters
Define fitness function
Particle swarm iterative optimization
Update particle velocity and position
Update the best location
Spatial data extraction
Spatial data preprocessing
Initial krig parameters
Waiting for PSO algorithm to optimize parameters
Update
Individual optimal position
Global optimal position
NO
Meet the termination conditions
YES
Obtain optimized parameters
Optimal Variational Function Model
End

Figure 7: Kriging Interpolation Process Optimized Based on Particle Swarm Optimization Algorithm

### Model 3: Evaluation Model

To quantify the performance of an algorithm, five metrics are employed for assessment, namely Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R-squared ($R^2$), and Residual Sum of Squares (RSS).

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{6}$$

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{7}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{8}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{9}$$

$$RSS = \sum(y_i - \hat{y}_i)^2 \tag{10}$$

## 4.2 Question 1: Model Solution

Firstly, we utilize a random uniform resampling model to sample the original data according to the specified process. Here, we set the sampling rates ranging from 10% to 90% respectively. The randomly uniformly resampled images are shown in Figure 8 below:

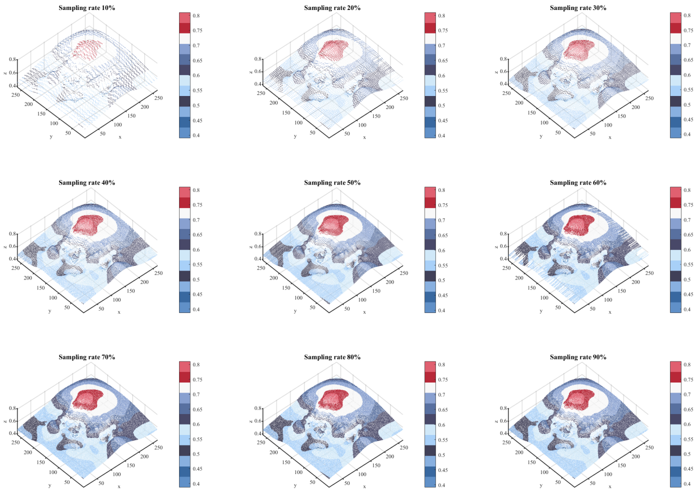


Figure 8: Illustrations of Different Sampling Rates

Observing Figure 8, it is evident that as the sampling rate increases, both the number and density of data points rise significantly. The data distribution remains uniform without abnormal variations such as substantial deviations, demonstrating the effectiveness of the random uniform resampling model established in this paper.

Next, we interpolate and fill the nine different samples extracted at 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90% sampling rates. Here, we employ the kriging interpolation algorithm model optimized based on the particle swarm algorithm that we established earlier.

The calculation steps are as follows:

**Step1:**Input the original data, which are the sampling points.

**Step2:**Grid the area by selecting the range and size of the grid.

**Step3:**Conduct data inspection and analysis to check whether the sampling values align with reality and remove the obviously discrepant points.

**Step4:**Calculate the histogram to decide whether preprocessing of the original data is necessary. In this step, we selected the sample data at rates of 30%, 60%, and 90% as representatives, calculated, and plotted the histograms for these three samples as shown in Figure 9 below:

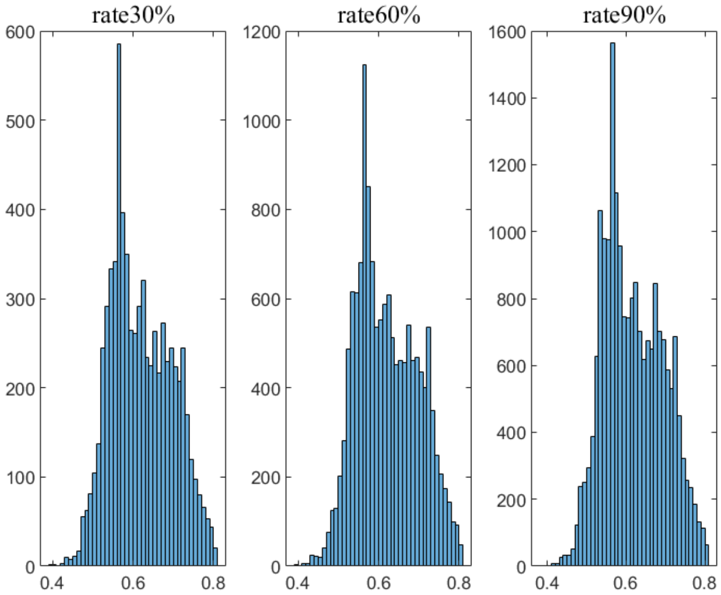


Figure 9: Sample Histograms

Based on the calculations, the sample data generally conform to a normal distribution, but there are peaks and outliers present. These anomalous data may affect the accuracy and stability of the interpolation. Therefore, we have performed data smoothing on the peaks and outliers in the nine sampled values to improve the accuracy and reliability of the Kriging interpolation.

**Step5:**Obtain the optimal variogram using Particle Swarm Optimization (PSO).

**Step6:**Calculate the variogram using the principles of variogram theory to understand the spatial structure of the variables.

**Step7:**Perform kriging interpolation estimation.

Through the above calculation steps, we interpolated the samples at 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90% sampling rates, respectively. We also conducted a visual comparison of the results before and after interpolation, as shown specifically in Figure 10 below:

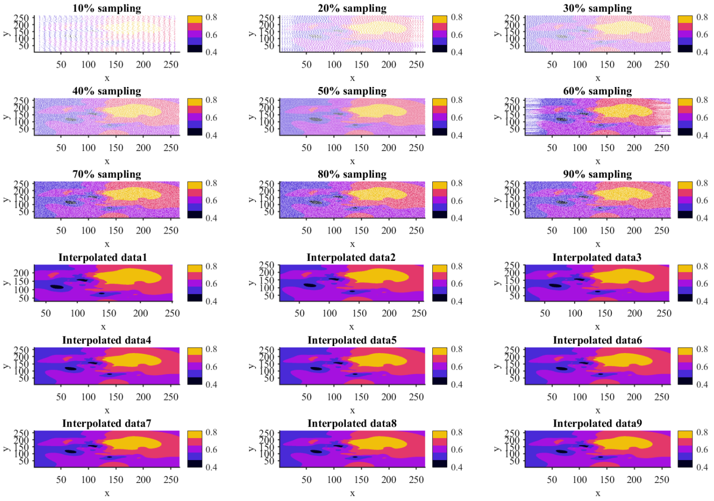


Figure 10: Comparison Charts of Each Sample Before and After Interpolation

Finally, we calculated the relationship between sample size and estimation error using the evaluation model we established. The specific results are shown in Table 1 below:

**Table 1 The table of correlation between sample size and estimation error**

| sampling rate | RSS($10^{-3}$) | MAE($10^{-3}$) | MSE | RMSE($10^{-3}$) | $R^2$ |
|---|---|---|---|---|---|
| 10% | 0.24 | 0.7 | 0 | 1.8 | 0.9993 |
| 20% | 0.23 | 0.4 | 0 | 1.5 | 0.9995 |
| 30% | 0.21 | 0.2 | 0 | 1.0 | 0.9998 |
| 40% | 0.19 | 0.2 | 0 | 0.7 | 0.9997 |
| 50% | 0.05 | 0.3 | 0 | 0.5 | 0.9998 |
| 60% | 0.01 | 0.1 | 0 | 0.5 | 0.9999 |
| 70% | 0.01 | 0.1 | 0 | 0.3 | 1 |
| 80% | 0.01 | 0.1 | 0 | 0.3 | 0.9999 |
| 90% | 0.01 | 0.1 | 0 | 0.1 | 1 |

The RSS gradually decreases as the sampling rate increases, indicating that as the sample size grows, the model's fit to the data improves, and the residuals diminish.At lower sampling rates (e.g., 10%), the RSS is relatively high, suggesting a lower model fit; whereas when the sampling rate reaches 70% or above, the RSS decreases significantly, indicating a very high model fit.

The MAE also decreases with increasing sampling rate, but the change is not as significant as that of the RSS.

At lower sampling rates, the MAE is relatively high, indicating a larger average deviation between the predicted and actual values; as the sampling rate increases, this deviation gradually diminishes.

Notably, there is a slight increase in MAE at a sampling rate of 50%, which may be due to the randomness of the samples.

The MSE is 0 at all sampling rates, which may imply that the model can predict the data very accurately at these sampling rates or that the data itself is very stable with no fluctuations.

This situation is relatively rare in practical applications and may require further examination of the data or model's validity.

RMSE, similar to MSE, decreases with increasing sampling rate, and the decreasing trend is more pronounced.

Since MSE is 0, RMSE is correspondingly 0 or close to 0, indicating that the error between the predicted and actual values is very small.

The $R^2$ value gradually approaches 1 as the sampling rate increases, indicating that the model's explanatory power gradually strengthens.

At lower sampling rates, the $R^2$ value is slightly lower, suggesting limited explanatory power of the model for the data; whereas when the sampling rate reaches 70% or above, the $R^2$ value is 1 or close to 1, indicating that the model almost completely explains the variation in the data.

The relationship between sample size and error is shown in Figure 11:

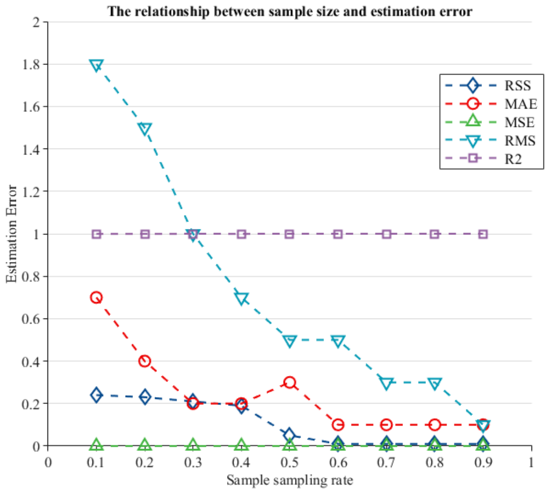


Figure 11: As can be seen from the relationship between sample size and error

The overall error trend decreases as the sample size increases, leading to higher estimation accuracy.

## 4.3 Question 2: Model Establishment

**Model 4: Tree Model Based on Genetic Algorithm with Elite Preservation Strategy**

Elite preservation strategy is a crucial approach in genetic algorithms, designed to ensure that outstanding individuals are not lost during the evolution process, enabling the accurate identification of key data among the covariates in Annex 1. Specifically, in each generation of evolution, the individuals with the highest fitness are selected and directly replicated to the next generation, bypassing subsequent crossover and mutation operations. This approach guarantees that the algorithm consistently retains the optimal or near-optimal solutions throughout the evolution process. The crossover schematic is illustrated in Figure 12 below:

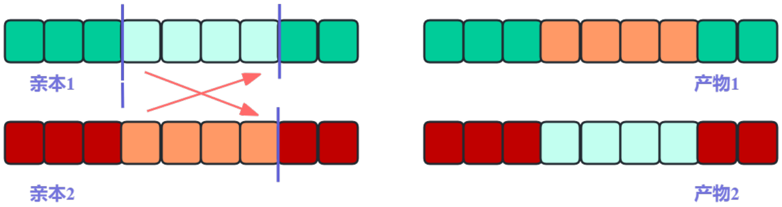


Figure 12: Two-Point Crossover Schematic

Tree models recursively partition datasets, progressively selecting features from the feature space that contribute most to predicting the target variable. A notable characteristic of tree models is their ability to naturally estimate feature importance. At each node of the tree model, the dataset is split into two subsets based on a particular feature, with the goal of increasing the purity of the resulting subsets (or reducing the value of the loss function) with respect to the target variable. By integrating the global search capabilities of genetic algorithms, we can automatically explore the feature space to find the optimal or near-optimal feature combinations. Simultaneously, utilizing the elite preservation strategy ensures that excellent feature combinations are not lost during the evolution process, thereby enhancing the performance and accuracy of the tree model.

Tree models refine the most influential features for predicting the target variable by recursively partitioning the dataset within the feature space. Their core advantage lies in their natural ability to assess the importance of each feature. At each node of the tree, the model splits the data based on a specific feature, aiming to enhance the purity of the subsets (or optimize the loss function value). By combining genetic algorithms, we can globally explore the feature space in search of the best or nearly best feature combinations. Concurrently, the elite preservation strategy is employed to ensure that outstanding feature combinations are retained throughout the iterations, thus improving the efficiency and accuracy of the tree model.

In the split nodes of a decision tree, the model partitions the data based on features, with the objective of maximizing purity gain (i.e., minimizing impurity). The importance of a feature is measured by its cumulative contribution to purity gain across the entire tree.

$$\text{Importance}(f_j) = \sum_{t\in T_j} \left(\text{Impurity}_{\text{parent}(t)} - \text{Impurity}_{\text{left}(t)} - \text{Impurity}_{\text{right}(t)}\right) \tag{11}$$

Where $T_j$ denotes all the nodes that utilize feature $f_j$ for splitting, and $Impurity$ represents a purity metric (such as Gini index or information gain).

Assuming there are $n$ features $f_1, f_2, ..., f_n$, the model is constructed through $m$ trees $T_1, T_2, ..., T_m$,For each feature$f_j$,its importance score is calculated as:

$$\text{Importance}(f_j) = \frac{1}{m}\sum_{i=1}^{m}\sum_{t\in T_i} \delta(f_j, t) \cdot \text{Impurity}(t) \tag{12}$$

Where $\delta(f_j, t)$ is an indicator function that denotes whether node $t$ uses a feature for splitting, and $Impurity(t)$ represents the impurity of that node.

Based on the data analysis, both the target and four covariate variables are standard normally distributed. Thus, the model computes Pearson correlation scores for all features, sorts them descendingly, and selects the higher-scoring ones for further modeling.

## 4.4 Question 2: Model Solution

Firstly, shuffle the data and split it into a training set. The genetic algorithm with elitist preservation strategy is described in the following pseudocode:

**Genetic Algorithm with Elitist Preservation Strategy for Optimizing Tree Models**

```
Set genetic algorithm parameters:
numGenerations = 50
populationSize = 20
eliteCount = 2
crossoverProbability = 0.8
Define the fitness function fitnessFunction(params, pn, tn, ps_input, ps_output):
Inputs:
    params: Parameters including the number of decision trees and the minimum leaf size
    pn: Predictor variables
    tn: Target variable
    ps_input, ps_output: (Not directly used in the code, possibly additional parameters)
```

```
Process:
    1. Extract and round the number of decision trees (trees) and the minimum leaf size (leaf)
    2. Construct a regression tree model using TreeBagger, setting OOBPredictorImportance to 'on', Method to 'regression', OOBPrediction to 'on', and MinLeafSize to leaf
    3. Use the model to predict pn and calculate the mean squared error between the predicted values and the actual values (tn) as the cost
Output:
    Return the cost
Set optimization options:
options = Set genetic algorithm options, including:
    - PopulationSize: populationSize
    - MaxGenerations: numGenerations
    - EliteCount: eliteCount
    - CrossoverFraction: crossoverProbability
    - Display: 'iter'
    - PlotFcn: Use gaplotbestf function to plot the change in the best value
Define variable bounds:
lb = [10, 1]  Lower bounds
ub = [2000, 50]  Upper bounds
Use the genetic algorithm for optimization:
Inputs:
    Fitness function: fitnessFunction
    Number of variables: 2
    Linear inequality constraints: None
    Linear equality constraints: None
    Bounds: lb, ub
    Nonlinear constraints: None
    Integer constraints: None
    Optimization options: options
Outputs:
    Optimal parameters x
    Optimal cost value fval
Construct the final model based on the optimal parameters:
1. Extract and round the optimal number of decision trees (optimalTrees) and the optimal leaf size (optimalLeaf)
2. Construct the final regression tree model using TreeBagger with the optimal parameters, setting OOBPredictorImportance to 'on', Method to 'regression', OOBPrediction to 'on', and MinLeafSize to optimalLeaf
End
```

The algorithm initializes parameters, defines a fitness function, sets optimization options, and defines variable bounds. It then constructs the final model based on the optimal parameters and calculates the results using the Pearson correlation coefficient, outputting the final results as shown in Figure 13 below:

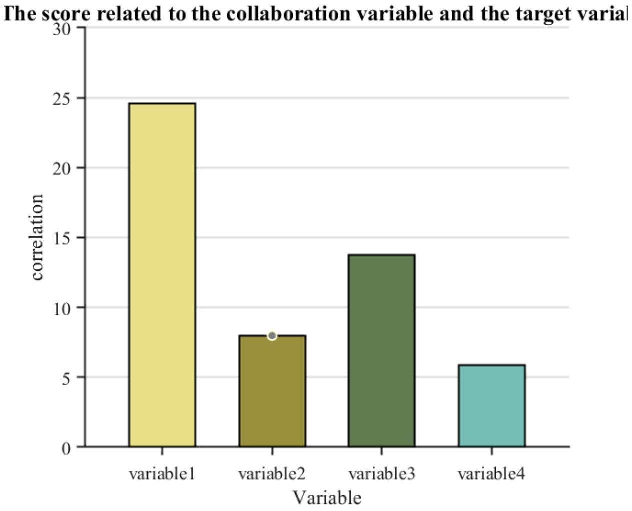


Figure 13: Correlation Analysis Result Diagram

This bar chart displays the correlation scores between variable1 to variable4 and the target variable. As clearly shown in the diagram:

variable1 has the highest correlation, nearing a score of 25, indicating an extremely strong association with the target variable.

variable3 follows closely, with a correlation of approximately 15, also demonstrating a high level of association.

variable2 has a slightly lower correlation, around 8.

variable4 has the lowest correlation, at just 5, showing the weakest association with the target variable.

**Table 2 Correlation of Variables**

| Variables | Correlation with target (rounded to two decimal places) |
|---|---|
| variable1 | 24.35 |
| variable2 | 7.89 |
| variable3 | 13.53 |
| variable4 | 5.80 |

As can be seen from the data in the table above, the two variables with the highest correlation are variable1 and variable3, so these two variables are selected as covariates.

## 4.5 Question 3: Model Establishment

**Model 5: Target Prediction Based on Combined Kriging-MLP Neural Network**

The neuron receives the $i$ input signal XI from different sources, and each signal is weighted according to its unique importance (i.e., connection weight). These weighted input signals are summed together to form the total input $\sum_{i=1}^{n} \omega_i x_i$ of the neuron. Subsequently, further adjustments are made within the neuron by subtracting a fixed threshold value $\theta$ from the total input, thereby determining the final processed input signal:

$$\sum_{i=1}^{n} \omega_i X_i - \theta \tag{13}$$

Kriging interpolation can be utilized to estimate values at unobserved locations in space. By applying Kriging interpolation, interpolation can be performed on the feature matrix $X_i$ and the target matrix $Y$ based on existing observation points, thus obtaining estimated values for each location. The general form of the Kriging interpolation model is as follows:

$$\widehat{X}_i(s) = \mu_i(s) + \sum^{m} \lambda_j K(s, s_j) \tag{14}$$

After obtaining the interpolated feature matrix and target matrix, the interpolation results are used as the input to a Multilayer Perceptron (MLP). An MLP is a feedforward neural network that consists of an input layer, hidden layers, and an output layer. Assuming the interpolated feature matrix is $\widehat{X}_i$ and the target matrix is $\widehat{Y}$, the model of the MLP can be represented as:

$$y = f(\widehat{X}, W_1, W_2, \ldots, W_k) \tag{15}$$

Through this method, appropriate parameters can be obtained through continuous training and used for actual prediction. This is the process of model solving.

**Model 6: Target Prediction Based on Combined Kriging-Random Forest Algorithm**

Similar to the Combined Kriging-MLP model, suppose there are multiple spatial feature grid matrices $X_1, X_2, ..., X_n$ and a target grid matrix $Y$, where each element of each grid matrix corresponds to the variable value at a certain location in space. The target grid matrix $Y$ represents the target values to be predicted.

Random Forest is an ensemble learning method that makes predictions based on multiple decision trees. Each decision tree is trained by randomly selecting a subset of samples and features from the data, and the final prediction is obtained through "voting" or averaging. In regression problems, the prediction formula for Random Forest is:

$$\widehat{y_i} = \frac{1}{T}\sum_{t=1}^{T} f_t(\widehat{X_i}) \tag{16}$$

**Model 7: Target Prediction Based on Differential Evolution Algorithm**

In this problem, we can establish a regression analysis model that relates two collaborative variables to the target. By interpolating the values of the collaborative variables, we can calculate the corresponding target based on the solved regression model. We can express this relationship as follows:

$$y_i = f\left(x_1^i, x_2^i, \cdots, x_j^i, \theta_1, \theta_2, \cdots, \theta_p\right) + \sigma_i \varepsilon \quad (i = 1, 2, \cdots, n) \tag{17}$$

Through experimental validation, the regression model that fits this relationship is shown below:

$$z = p1 + p2 \cdot v1 + p3 \cdot v3 + p4 \cdot v3^2 \tag{18}$$

For this problem, we choose the differential evolution algorithm, which is effective in solving global optimization problems, to optimize $\theta_1, \theta_2, ..., \theta_n$. The mutation vector is:

$$H_i(\mathrm{g}) = X_{p1}(\mathrm{g}) + F \cdot \left(X_{p2}(\mathrm{g}) - X_{p3}(\mathrm{g})\right) \tag{19}$$

The mutation operator is:

$$V_i = X_b + F_i(X_m - X_w) \tag{20}$$

The mutation strategy is:

$$\text{DE/rand/}1 : V_i(\mathrm{g}) = X_{p1}(\mathrm{g}) + F\left(X_{p2}(\mathrm{g}) - X_{p3}(\mathrm{g})\right) \tag{21}$$

## 4.6 Question 3: Model Solution

Based on the conclusion of the correlation ranking between collaborative variables and the target variable in Question 2, we additionally selected Collaborative Variable 1 and Collaborative Variable 3 in the interpolation process of Question 3 to investigate the spatial variation patterns of the variables. The interpolation results are shown in Figure 14 below:

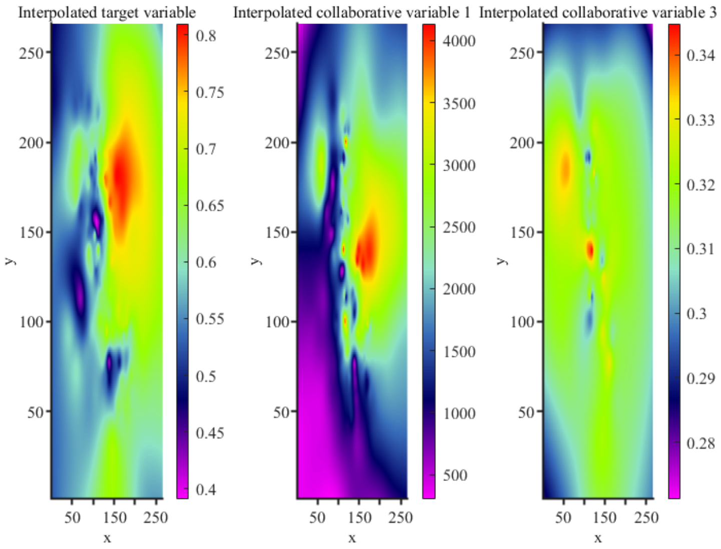


Figure 14: The Kriging interpolation results for one target variable and two collaborative variables

By comparing the interpolated contour plot in Figure 14 with the original data contour plot from the previous data analysis, it can be observed that the data distribution and trends after interpolation are generally consistent with the original data.

Subsequently, we repeated the experimental steps using samples with sampling rates of 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, and 90%, and employed Models 1, 2, and 3. The results are presented in Table 3 below:

**Table 3 Estimation Errors of the Combined Kriging-MLP Neural Network**

| sampling rate | RSS($10^{-3}$) | MAE($10^{-3}$) | MSE($10^{-6}$) | RMSE($10^{-3}$) | $R^2$ |
|---|---|---|---|---|---|
| 10% | 0.14 | 1.06 | 5.14 | 2.27 | 0.9991 |
| 20% | 0.11 | 0.93 | 4.04 | 2.01 | 0.9995 |
| 30% | 0.09 | 0.91 | 3.79 | 1.95 | 0.9996 |
| 40% | 0.04 | 0.99 | 4.66 | 2.16 | 0.9997 |
| 50% | 0.02 | 0.10 | 4.78 | 2.19 | 0.9998 |
| 60% | 0.01 | 1.14 | 5.2 | 3.19 | 0.9999 |
| 70% | 0.01 | 1,16 | 4.52 | 2.55 | 1 |
| 80% | 0.01 | 1.23 | 4.16 | 2.48 | 1 |
| 90% | 0.01 | 1.11 | 4.74 | 2.18 | 1 |

The RSS values significantly decreased when the sampling rate was between 10% and 30%, and then fluctuated slightly. However, there was a sudden increase at a 60% sampling rate, followed by another decrease. This fluctuation may reflect that at certain sampling rates, the model's fit to the data is more accurate or unstable.

The $R^2$ values gradually approached 1 as the sampling rate increased, indicating that the model's fit improved continuously. Even at lower sampling rates, the $R^2$ values were already very close to 1, suggesting that the overall performance of the model was quite good.

The trends of MAE and MSE were not entirely consistent, indicating that the distribution of large and small errors may vary at different sampling rates. Overall, the Combined Kriging-MLP Neural Network demonstrated good performance across different sampling rates, with $R^2$ values consistently close to or equal to 1.

**Table 4 Estimation Errors of the Combined Kriging-Random Forest Algorithm**

| sampling rate | RSS($10^{-3}$) | MAE($10^{-3}$) | MSE($10^{-6}$) | RMSE($10^{-3}$) | $R^2$ |
|---|---|---|---|---|---|
| 10% | 0.15 | 1.06 | 5.15 | 2.3 | 0.9992 |
| 20% | 0.16 | 0.93 | 4.04 | 2 | 0.9991 |
| 30% | 0.14 | 0.91 | 3.78 | 1.9 | 0.9992 |
| 40% | 0.12 | 0.99 | 4.67 | 2.2 | 0.9994 |
| 50% | 0.08 | 1.03 | 4.79 | 2.2 | 0.9997 |
| 60% | 0.01 | 1.14 | 5.21 | 3.2 | 1 |
| 70% | 0.01 | 1,16 | 6.52 | 2.6 | 0.9999 |
| 80% | 0.01 | 1.23 | 6.16 | 2.5 | 0.9999 |
| 90% | 0.01 | 1.12 | 4.76 | 2.2 | 0.9999 |

The Mean Absolute Error (MAE) decreased as the sampling rate increased from 10% to 30%, but then exhibited a fluctuating trend at higher sampling rates without a clear pattern of decrease or increase. This suggests that while increasing the sampling rate may help reduce the average absolute error, this relationship is not monotonic.

The trends of Mean Squared Error (MSE) and Root Mean Squared Error (RMS) were similar, both being higher at lower sampling rates and then decreasing as the sampling rate increased. However, there was a sudden increase at a 60% sampling rate, followed by decreases at 70% and 80% sampling rates. This reflects the complexity of the errors, which may be influenced by multiple factors.

The $R^2$ (coefficient of determination) was very close to 1, indicating that the model performed exceptionally well at all these sampling rates, almost perfectly explaining the variability in the data. This demonstrates the high efficiency and stability of the Combined Kriging-Random Forest algorithm. The Combined Kriging-Random Forest algorithm exhibited good performance across different sampling rates, but the choice of the optimal sampling rate requires comprehensive consideration of multiple error metrics as well as cost and time constraints in practical applications.

**Table 5 Estimation Errors of Nonlinear Regression**

| sampling rate | RSS($10^{-3}$) | MAE($10^{-3}$) | MSE($10^{-6}$) | RMSE($10^{-3}$) | $R^2$ |
|---|---|---|---|---|---|
| 10% | 64.21 | 37.81 | 2305.66 | 48.24 | 0.68 |
| 20% | 58.43 | 38.12 | 2319.88 | 48.23 | 0.69 |
| 30% | 54.25 | 37.15 | 1598.42 | 39.92 | 0.60 |
| 40% | 57.30 | 40.42 | 2524.41 | 50.24 | 0.67 |
| 50% | 73.29 | 40.81 | 2001.26 | 44.79 | 0.56 |
| 60% | 55.84 | 43.44 | 2749.79 | 52.46 | 0.64 |
| 70% | 69.47 | 47.32 | 2862.93 | 53.55 | 0.57 |
| 80% | 54.92 | 44.26 | 2879.65 | 53,74 | 0.63 |
| 90% | 74.32 | 42.58 | 2201.5 | 46.92 | 0.54 |

Upon observing these data, it can be seen that there is no single sampling rate that minimizes all error metrics. For instance, the RSS is minimized at a 30% sampling rate, while the MAE is relatively smaller at both 30% and 10% sampling rates. The MSE and RMS reach a relatively low point at a 30% sampling rate. According to the table, the $R^2$ value is highest at a 20% sampling rate, reaching 0.69, indicating that the model fits the data relatively well at this point. However, as the sampling rate increases, the $R^2$ value does not show a clear upward or downward trend, but fluctuates between 0.54 and 0.69.

In nonlinear regression, no definitive "optimal" sampling rate minimizes all error metrics. Different rates yield varying prediction errors and model fits, with a nonlinear relationship between sampling rate and error, as shown in Figure 15:

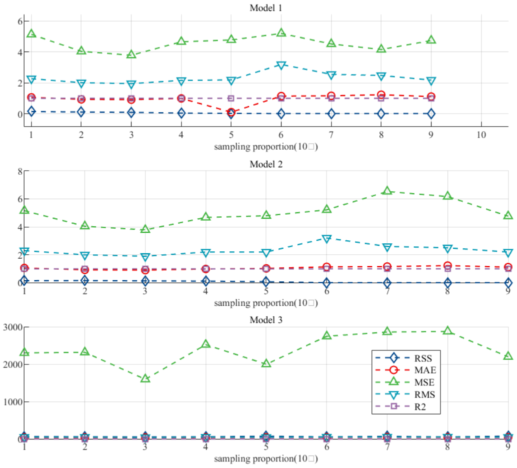


Figure 15: The Relationship between Sample Size and Error

Finally, we calculated the average values of each evaluation metric for the results of the three models and conducted a comparative analysis based on the data. The results are presented in Table 6 below:

**Table 6 Average Error Values for Three Models**

| | RSS($10^{-3}$) | MAE($10^{-3}$) | MSE($10^{-6}$) | RMSE($10^{-3}$) | $R^2$ |
|---|---|---|---|---|---|
| Kriging-Multilayer Perceptron | 0.05 | 1.03 | 4.39 | 2.26 | 0.9996 |
| Kriging-Random Forest | 0.09 | 1.06 | 4.81 | 2.37 | 0.9996 |
| Nonlinear Regression | 60.11 | 40.63 | 2405.11 | 48.77 | 0.63 |

In terms of the RSS metric, Kriging-MLP performed the best with a value of $0.05 \times 10^{-3}$, indicating the smallest difference between its predicted and actual values. Kriging-Random Forest, with a value of $0.09 \times 10^{-3}$, was slightly worse than Kriging-MLP but the difference was not significant. Nonlinear Regression had a much higher value of $60.11 \times 10^{-3}$, indicating larger prediction errors.

For the MAE metric, Kriging-MLP had the lowest value among the three models, indicating the smallest average absolute difference between its predicted and actual values. In terms of the MSE metric, Kriging-MLP also performed the best, indicating the smallest mean squared difference between its predicted and actual values. Kriging-MLP also had the lowest RMS value. For the $R^2$ metric, both Kriging-MLP and Kriging-Random Forest had values of 0.9996, which are very close to 1.

In summary, both Kriging-MLP and Kriging-Random Forest performed excellently across all error metrics, indicating that these two models can fit the data well with small prediction errors. Nonlinear Regression, on the other hand, performed poorly across all error metrics, with larger prediction errors and a weaker correlation with actual values. Therefore, among the three models, Kriging-MLP is the optimal choice.

## 4.7 Question 4: Model Solution

Through comparative experiments, this paper has identified Kriging-MLP as the optimal method for estimating the trend of the target variable in the third question. Therefore, in the fourth question, this model is used to perform interpolation estimation on the F2 target variable. The results are visualized, and to more vividly present the trend characteristics of the F2 target variable, multiple color schemes are employed to enhance the visual effect of the F2 target variable trend, as shown in Figure 16 below:

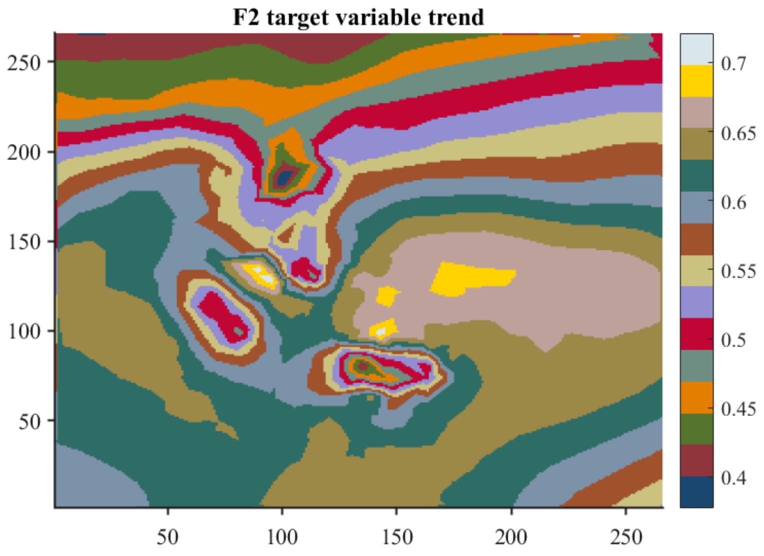


Figure 16: Contour Plot of F2 Target Variable Trend

According to the calculations, the F2 target variable has lower values in the bottom-left and bottom-right regions, higher values in the middle to right portion, and a relatively smooth transition in the central area.

# 5. Test the Models

In the model testing phase, we employed cross-validation to evaluate the generalization ability and stability of the models. Cross-validation is a statistical method used to assess and analyze the performance of machine learning models. By training and testing different subsets of the model multiple times, it can avoid overfitting and underfitting issues, thereby providing a more reliable assessment of model performance.

The specific steps are as follows:

**Step1:** Data Division: We divided the original dataset into a training set and a test set. In this study, we used 80% of the data as the training set and the remaining 20% as the test set.

**Step2:** Model Training: We trained our models using the training set data, including the Random Uniform Resampling model, the Kriging Interpolation model based on Particle Swarm Optimization, the Tree model based on Elite Retention Strategy Genetic Algorithm, the Combined Kriging-MLP Neural Network model, the Combined Kriging-Random Forest model, and the Nonlinear Regression model.

**Step3:** Model Prediction: We used the trained models to predict the test set data.

**Step4:** Performance Evaluation: We assessed the performance of the models by calculating metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and $R^2$ value on the test set.

**Step5:** Cross-Validation: We adopted the K-Fold Cross Validation method, dividing the dataset into K equal-sized subsets. Each time, K-1 subsets were used for training, and the remaining subset was used for testing. This process was repeated K times to ensure that each subset was used as the test set once. Through this approach, we obtained a more robust assessment of model performance.

Through cross-validation, we found that all models performed well on the test set, especially at higher sampling rates. The Combined Kriging-MLP Neural Network model exhibited the best performance in most metrics, validating its effectiveness in predicting the target variable.

# 6. Sensitivity Analysis

In this study, we primarily conducted sensitivity analyses on the following parameters:

(1) Sampling Rate: We analyzed the impact of different sampling rates on model prediction errors. By varying the sampling rate, we observed that as the sampling rate increases, the model's prediction error generally decreases, but this also leads to increased computational costs. Therefore, it is necessary to find a balance between prediction accuracy and computational efficiency.

(2) Model Parameters: For models based on optimization algorithms (such as Particle Swarm Optimization, Genetic Algorithm, etc.), we analyzed the impact of different parameter settings (such as population size, number of iterations, crossover probability, etc.) on model performance. By adjusting these parameters, we can optimize the model's search capability and convergence speed.

(3) Feature Selection: For tree models and regression models, we analyzed the impact of different feature combinations on model performance. Through feature selection, we can eliminate features that contribute less to the prediction of the target variable, thereby simplifying the model and improving prediction efficiency.

The results of the sensitivity analysis indicate that the sampling rate and model parameters have significant impacts on model performance. Through reasonable parameter selection and optimization, we can further improve the model's prediction accuracy and stability.

# 7. Strengths and Weakness

## 7.1 Strengths

(1) Multi-model Fusion: In this study, multiple models were employed for the prediction of the target variable, and the optimal model was identified through comparative analysis. This multi-model fusion approach enhanced the accuracy and robustness of the predictions.

(2) Data Preprocessing: Prior to model training, we conducted thorough data preprocessing, including data cleaning, visual analysis, and feature selection. These efforts contributed to improving the quality and generalization ability of the models.

(3) Cross-validation and Sensitivity Analysis: Through cross-validation and sensitivity analysis, we comprehensively evaluated and optimized the model performance and parameters, thereby enhancing the stability and prediction accuracy of the models.

## 7.2 Weakness

Computational Cost: Models based on optimization algorithms, such as Particle Swarm Optimization, Genetic Algorithm, etc., are computationally complex and require longer runtime. This may pose limitations in practical applications.

Parameter Tuning: The tuning of model parameters requires a certain level of experience and professional knowledge. Improper parameter settings may lead to decreased model performance or overfitting.

Data Dependence: The performance of models heavily relies on the quality and quantity of the input data. Issues such as noise or missing values in the input data may affect the prediction accuracy and stability of the models.

# Reference

[1] Ding Ziwei, Liu Jiang, Wang Xiaoyong, et. Al. Research on 3D Geological Modeling Technology Based on PSO-Kriging Algorithm [J]. Coal Engineering, 2024, 56(10): 82-89.

[2] Wei Xinpeng, Yao Zhongyang, Bao Wenli, et. Al. A Reliability Analysis Method Based on Active Learning Kriging Model and Evidence Theory [J]. Journal of Mechanical Engineering, 2024, 60(02): 356-368.

[3] Ye Shuangyi, Hu Xiaoxiang, Si Xiaosheng, et. Al. Reliability Assessment Method for Complex Systems Using Sliding Window and Kriging Interpolation Algorithm [J]. Journal of Xi'an Jiaotong University, 2023, 57(04): 171-179.

[4] Zhang Qi, Ma Yanning, Wang Xiaojun, et. Al. Tunnel Main Structural Surface Modeling Based on Optimized Kriging Interpolation Method [J]. China Civil Engineering Journal, 2022, 55(S2): 74-82. DOI: 10.15951/j.tmgcxb.2022.s2.07.

[5] Zhan Jiang, Li Zhiping, Zhao Guizhang, et. Al. Transfer Function and Regression Kriging Estimation of Saturated Hydraulic Conductivity in Different Soil Layers of the Vadose Zone Based on PCA-GWR [J]. Earth Science, 2024, 49(03): 978-991.

# Appendix

**Data Preprocessing Code 1**

```
map = TheColor('sci', 704);
figureUnits = 'centimeters';
figureWidth = 15;
figureHeight = 10;
figureHandle = figure;
set(gcf, 'Units', figureUnits, 'Position', [0 0 figureWidth figureHeight]);
Z_current = Z;
title_text = 'F2 target variable trend';
p = pcolor(X, Y, Z_current);
title(title_text);
xlabel('x');
ylabel('y');
colormap(map);
colorbar;
p.EdgeColor = 'none';
axis tight;
set(gca, 'Box', 'off', ...
'LineWidth', 1, 'GridLineStyle', '-',...
'Layer', 'top',...
'XGrid', 'off', 'YGrid', 'off', ...
'TickDir', 'out', 'TickLength', [.01 .01], ...
'XMinorTick', 'off', 'YMinorTick', 'off',...
'XColor', [.1 .1 .1], 'YColor', [.1 .1 .1]);
set(gca, 'FontName', 'Times New Roman', 'FontSize', 11);
set(get(gca, 'XLabel'), 'FontName', 'Times New Roman', 'FontSize', 11);
set(get(gca, 'YLabel'), 'FontName', 'Times New Roman', 'FontSize', 11);
set(get(gca, 'Title'), 'FontSize', 12, 'FontWeight', 'bold', 'FontName', 'Times New Roman');
set(gcf, 'Color', [1 1 1]);
figW = figureWidth;
figH = figureHeight;
set(figureHandle, 'PaperUnits', figureUnits);
set(figureHandle, 'PaperPosition', [0 0 figW figH]);
fileout = 'Pseudocolor Image';
print(figureHandle, [fileout, '.png'], '-r300', '-dpng');
```

**Data Preprocessing Code 2**

```
map = TheColor('sci', 1332);
figureUnits = 'centimeters';
figureWidth = 15;
figureHeight = 10;
```

```
figureHandle = figure;
set(gcf, 'Units', figureUnits, 'Position', [0 0 figureWidth figureHeight]);
Z_current = Z2;
title_text = 'F1 collaborative variable1';
s = surf(X, Y, Z_current, 'EdgeColor', 'none');
title(title_text);
xlabel('x');
ylabel('y');
zlabel('z');
view(-41.9, 69.5);
colormap(map);
colorbar;
axis tight;
set(gca, 'Box', 'off', ...
'LineWidth', 1, 'GridLineStyle', '-',...
'XGrid', 'on', 'YGrid', 'on', 'ZGrid', 'on',...
'TickDir', 'out', 'TickLength', [.01 .01], ...
'XColor', [.1 .1 .1], 'YColor', [.1 .1 .1], 'ZColor', [.1 .1 .1]);
set(gca, 'FontName', 'Times New Roman', 'FontSize', 11);
set(get(gca, 'XLabel'), 'FontName', 'Times New Roman', 'FontSize', 11);
set(get(gca, 'YLabel'), 'FontName', 'Times New Roman', 'FontSize', 11);
set(get(gca, 'ZLabel'), 'FontName', 'Times New Roman', 'FontSize', 11);
set(get(gca, 'Title'), 'FontSize', 12, 'FontWeight', 'bold', 'FontName', 'Times New Roman');
set(figureHandle, 'Renderer', 'opengl');
figW = figureWidth;
figH = figureHeight;
set(figureHandle, 'PaperUnits', figureUnits);
set(figureHandle, 'PaperPosition', [0 0 figW figH]);
set(figureHandle, 'PaperSize', [figW figH]);
fileout = 'Surface Plot';
print(figureHandle, [fileout, '.png'], '-r600', '-dpng');
```

**Program Code 1**

```
data = [];
for i=1:size(v1,1)
for j=1:size(v1,2)
data = [data ;[v1(i,j),v2(i,j),v3(i,j),v4(i,j),target(i,j)]];
end
end
TE= randperm(size(data,1));
PN = data(TE, 1: end-1)';
TN = data(TE, end)';
```

```
[pn, ps_input] = mapminmax(PN, 0, 1);
pn=pn';
[tn, ps_output] = mapminmax(TN, 0, 1);
tn=tn';
trees = 1250;
leaf = 5;
OOBPrediction = 'on';
OOBPredictorImportance = 'on';
Method = 'regression';
net = TreeBagger(trees, pn, tn, 'OOBPredictorImportance', OOBPredictorImportance,...
'Method', Method, 'OOBPrediction', OOBPrediction, 'minleaf', leaf);
importance = net.OOBPermutedPredictorDeltaError;
set(gca, 'XTickLabel', 'variable1', 'variable2', 'variable3', 'variable4');
xlabel('Variable');
ylabel('Correlation');
barWidth = 0.8;
set(gca, 'BarWidth', barWidth);
```

**Program Code 2**

```
function [new_set_nan] = myKriging(new_set)
[m, n] = size(new_set);
[rows, cols] = find( isnan(new_set));
valid_values = new_set(sub2ind([m, n], rows, cols));
x = cols;
y = rows;
z = valid_values;
interp = scatteredInterpolant(x, y, z, 'linear', 'none');
[rows_nan, cols_nan] = find(isnan(new_set));
new_set_nan = new_set;
for i = 1:length(rows_nan) new_set_nan(rows_nan(i), cols_nan(i)) = interp(cols_nan(i), rows_nan(i));
end
M = nanmean(new_set(:));
new_set_nan(isnan(new_set_nan)) = M;
end
```

**Program Code 3**

```
O_target=load("A1").new_set;
v1=load("A2").v1;
v3=load("A3").v3;
filename = 'A4';
mask = zeros(size(v1,1),size(v1,2));
```

```
mask(isnan(v1)) = 1;
[v1] = AKriging(v1);
[v3] = AKriging(v3);
data = [];
for i=1:size(v1,1)
for j=1:size(v1,2)
if mask(i,j)==1
data = [data ;[v1(i,j),v3(i,j),O_target(i,j)]];
end
end
end
R = randperm(size(data,1));
train_ratio=0.6;
train=data(R(1:(floor(train_ratio*size(data, 1)))),:);test=data(((R(floor(train_ratio*size(data, 1))))+1):size(data, 1),:);save(filename, 'train', 'test');
```