2022 S.T. Yau High School Science Award (Asia)

Research Report

The Team

Registration Number: math-153, comp-113

Name of team member: Hou Yibo
School: Victoria Junior College
Country: Singapore

Name of team member:
School:
Country:

Name of supervising teacher: Yang Zejun
Job Title: Research Fellow
School: National University of Singapore
Country: Singapore

Title of Research Report

An optimized prediction model of RON loss in gasoline refining process

Date

31 August 2022

Commitments on Academic Honesty and Integrity

We hereby declare that we

1. are fully committed to the principle of honesty, integrity and fair play throughout the competition.
2. actually perform the research work ourselves and thus truly understand the content of the work.
3. observe the common standard of academic integrity adopted by most journals and degree theses.
4. have declared all the assistance and contribution we have received from any personnel, agency, institution, etc. for the research work.
5. undertake to avoid getting in touch with assessment panel members in a way that may lead to direct or indirect conflict of interest.
6. undertake to avoid any interaction with assessment panel members that would undermine the neutrality of the panel member and fairness of the assessment process.
7. observe the safety regulations of the laboratory(ies) where the we conduct the experiment(s), if applicable.
8. observe all rules and regulations of the competition.
9. agree that the decision of YHSA(Asia) is final in all matters related to the competition.

We understand and agree that failure to honour the above commitments may lead to disqualification from the competition and/or removal of reward, if applicable; that any unethical deeds, if found, will be disclosed to the school principal of team member(s) and relevant parties if deemed necessary; and that the decision of YHSA(Asia) is final and no appeal will be accepted.

(Signatures of full team below)

X Hou Yibo
Name of team member: Hou Yibo

X
Name of team member:

X
Name of team member:

X Zejun Yang
Name of supervising teacher: Zejun Yang

Noted and endorsed by

Ms Grace Lee
Vice-Principal
(signature) Victoria Junior College 30/8/2022

Name of school principal:

An optimized prediction model of RON loss in gasoline refining process

Hou Yibo

Abstract

Exhaust gas from cars has caused environmental pollution around the globe. An increasing number of oil refining companies and governments are setting higher emission standards (NEA, *Air Pollution Regulations* 2022) resulting in an increased demand for patrols with higher quality. Hence, desulfurizing fuel during the refinement process has become a popular trend around the globe. However, in the refining process, conventional prediction of research octane number (RON) in the industrial refining process lacks the ability to predict the characteristics of fuels outside the training data set and can result in a major RON loss when petrol is desulfurized. In addition, the loss in RON can create a large financial loss for companies and consumers. The less the RON is reduced, the higher the economic benefits the company can receive. In the consideration of economic benefits, petrochemical companies maintain the RON loss in the desired range (0.5-1) (Lu et al., 2021). Hence, to fulfil the need for the better-quality petrol and reduce the RON loss, it is important to create a new model to predict RON loss in the refinement process while ensuring that petrol is desulfurized.

These are the main areas that we have worked on from existing industrial data

1. We processed industrial data, eliminating data with excess null values while completing the data with few null values. We conducted the normalization of data and used 3σ rule to eliminate outliers. Through data processing, we eliminated 1966 abnormal data to build the foundation for further analysis
2. We used Grey Relational Analysis, K-means Clustering and Random Forest Model to conduct dimension reduction to decrease complexity to intrinsic dimensional variables that are correlated with RON value. By simplifying variables, we find 16 variables that will be investigated as main variables.
3. Since different factors have different significance to affect RON loss, we introduced a neural network prediction model embedded in SE-NET, and a weighted Loss function is constructed according to our optimization objective to complete the neural network prediction model of octane Loss and sulfur content, the prediction results are verified with the original data values of samples.
4. Accounting the main variables optimisation, we proposed an optimization method of operation variables based on three prediction model and genetic algorithm. We use prediction model to predict the variation of main variables for octane number loss and sulfur content to obtain fitness, then according to the fitness training iteration times to find the main operation variables on the optimal solution. The optimized operation variables can reduce the octane loss by 38.3% on average, and the optimization rate is 66.5%.

Key words: RON Loss, Grey Relational Analysis, K-means Clustering, Random Forest Model, Artificial neural network, Genetic algorithm

Acknowledgement

1. Dr Zejun Yang, NUS Department of Design and Engineering for providing support and guidance

Contents

1. Introduction

Exhaust gas from cars has caused environmental pollution around the globe. An increasing number of oil refining companies and governments are setting higher emission standards (Singapore, Euro IV), resulting in an increased demand for patrols with higher quality. Hence, desulfurizing fuel during the refinement process has become a popular trend around the globe. However, in the refining process, conventional prediction of research octane number (RON) in the industrial refining process lacks the ability to predict characteristics of fuels outside the training data set and can result in a major RON loss when petrol is desulfurized。 In addition, the loss in RON can create a large financial loss for companies and consumers. The less the RON is reduced, the higher the economic benefits the company can receive. In the consideration of economic benefit, petrochemical companies actively maintain the RON loss in the desired range (0.5-1). Hence, to fulfil the need for the better-quality petrol and reduce the RON loss, it is important to create a new model to predict RON loss in the refinement process while ensuring that petrol is desulfurized.

I obtained 2 raw sets of data from a petrol chemical enterprise. However, certain variables in data sets are not available all times and some variables are partially or fully null values. Hence, data analysis is needed.

I want to work out a RON loss prediction model based on the data set. I made related assumptions and defined variables based on the data given.

Assumption 1: The measured value of RON is the comprehensive effect of the operation within two hours before the measurement time, and the average value of the operation variable within two hours corresponds to the measured value of octane number.

Assumption 2: If there are more than 4 values empty in a variable sequence, it is considered to have too many incomplete values

Assumption 3: The operation range of the data variable ranges from the minimum to the maximum value of the variable in Annex 1.

Assumption 4: The range that a manipulating variable can change will be within the +- 20% of the average of the variable, $\overline{X_i}$

Symbol	Instructions

X_i	Operational variable sequence
X_0	RON loss sequence
$\overline{X_i}$	Average of a operational sequence
σ	Standard Error
Ntree	Total number of decision trees
errorOOB1	out-of-bag error
errorOOB2	Add noise outside error
IMP	Importance of operating variable
S	Sulfur content
R	RON loss
S_pred	Predicted sulfur content
RON_pred	Predicted RON loss
R_best	Optimum
fitness	The minimum current state in the model optimization process

Chart 1.1 definition of variables

2. Raw data processing

2.1 Preprocessing

Data pre-processing refers to manipulation or dropping of data before it is used in order to ensure or enhance performance (Wikipedia, *Data pre-processing*). It simplifies and improves the reliability of the data. From analysis and observation in data sample 285 and 313(Excel files), I first deleted some of the incomplete data to make it simpler for sorting out main variables, improving the efficiency of data mining. Then I mended the data to improve the quality of raw data, laying the foundation for constructing reliable RON loss model. The process is showed below(Fig 2.1)

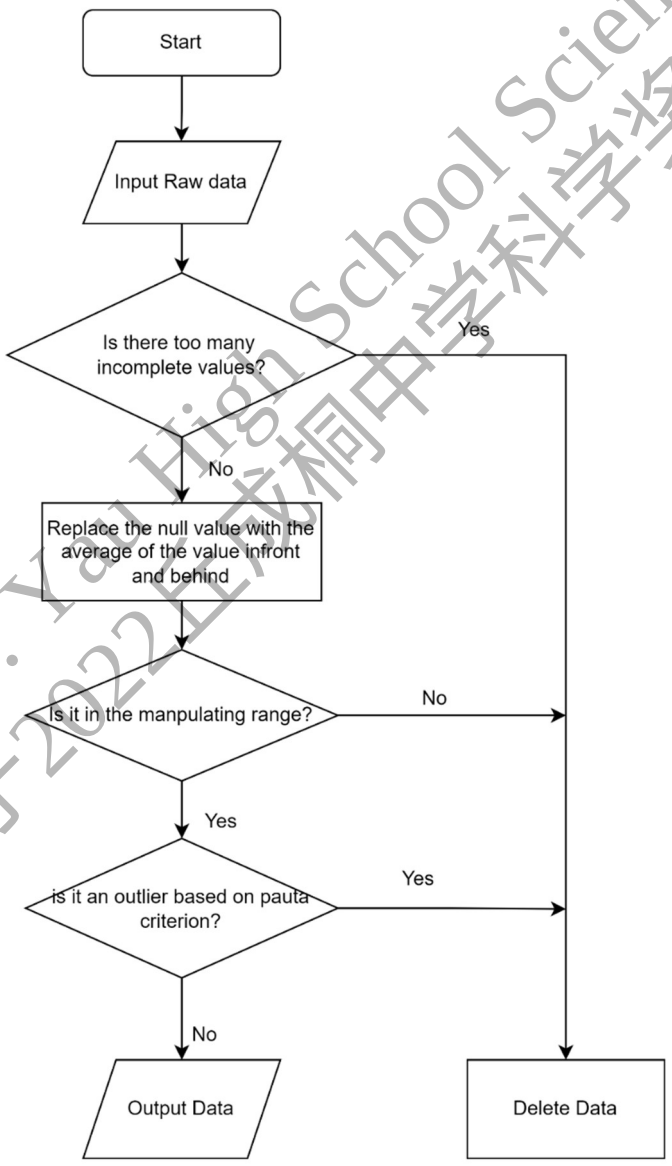


Fig2.1 Flow chart for data preprocessing

2.2 Introduction about raw data and standard of data processing

Raw data are collected from a live data base(HoneywellPHD) and LIMS data base of a petrol chemical enterprise. The operating variables are from the live data base while data about raw material, products and catalytic agent from LIMS data base. In the raw data, the majority of variable data are normal. However, there are certain problemetic data points for different instrstial devices: certain data points does not show up all the time, certain data points' values are fully or partially null values.

Requirement for data processing :

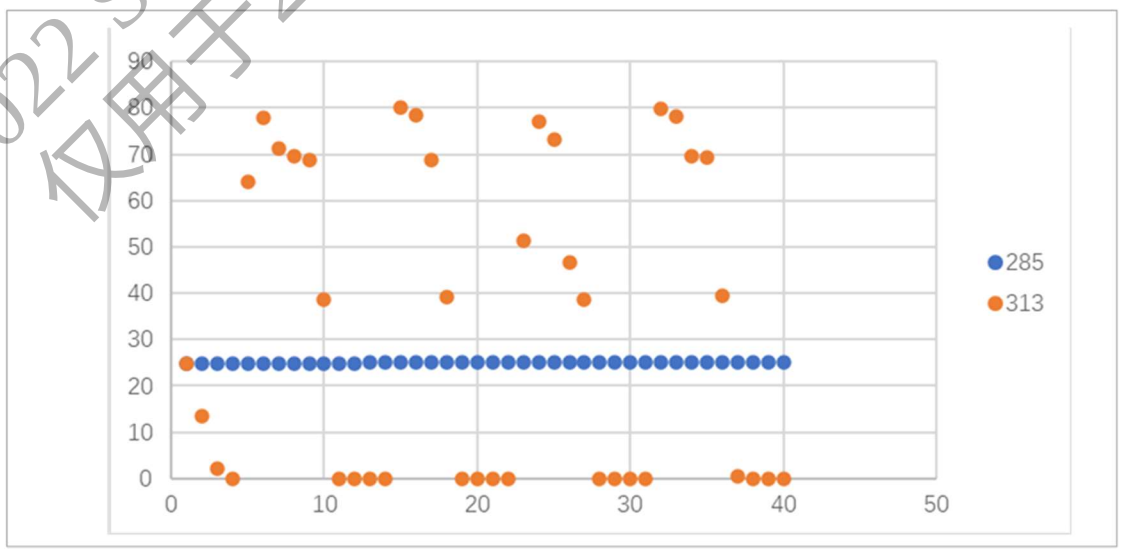
1. For data that are not available full time, if there are too many incomplete data points which make it hard to be amended, I will delete the data points
2. Delete data that only contain null values in 325 sample data
3. For data points that are null value, replace it with the average of value of +-2 hours
4. Combining operating requirements to decide the manipulation range of the variables in raw data set, and set the maximum and minimum range and delete data points that are not in the range.
5. Use 3σ rule to delete abnormal values

2.3 Data processing

2.3.1 Null value processing

For example, in Fig 2.2, I illustrated D-106 hot nitrogen gas flow data in 285 and 313 data sample. It is every evident that D-106 contain lots of null values in 313 which makes it hard for us to find the trend. However, in 285, the data is clean and there is a very evident trend. Thus, I decided that D-106 in 313 are abnormal data, these data points are deleted.

Fig 2.2 D-106 hot nitrogen gas flow data in both 313 and 285



Considering that the devices to collect data can be malfunctioned and in maintenance, causing null value in data points, it is important to decide the standard to delete data with too many null values. If there are more than 4 null value, it is deemed as unamendable and these data points will be deleted. If there are less than 4 null values, I will use the average value of data collected from +-2hours to replace the null value. The process is as shown in Chart 2.1

Type of null value	How to decide	Operating procedure
Many null values	Number of Null values ≥ 4	Delete these data points
Few null values	Number of Null values ≤ 3	Replace it with average value

Chart2.1

2.3.2 Range for manipulation

Since the range for manipulation of variables is decided from operating requirements, I use assumption 3 to decide the range for manipulation of variables. MATLAB is used to iterate data in 285 and 313 samples (x_1, x_2,……, x_n) to calculate the average x and create function f(xi) to decide the range for x_i, f(x) is shown in equation 2-1

$$f(x_i) = \begin{cases} null & x_i < 0.8x \text{ or } x_i > 1.2x \\ x_i & 0.8x \leq x_i \leq 1.2x \end{cases} \quad (2\text{-}1)$$

2.3.3 Data processing for outliers

There are outliers in the raw data, we will use 3σ rule to determine if the variable is an outlier. 3σ criterion：Calculate the arithmetic mean value x according to the equal precision measurement values of the variables x_1, x_2,……, x_n, and the residual error $v_i = x_i - x$ ，then the standard error σ is obtained by the Bessel formula，if there is a variable to residual error v_b of the measured value v_a, make

$$|v_b| = |x_b - x| > 3\sigma$$

X_b is considered as an outlier. The Bessel formula is shown in 2-2

$$\left[\frac{1}{n-1}\sum_{i=1}^{n} v_i^2\right]^{1/2} = \left\{\left[\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2 / 2\right] / (n-1)\right\}^{1/2} \quad (2\text{-}2)$$

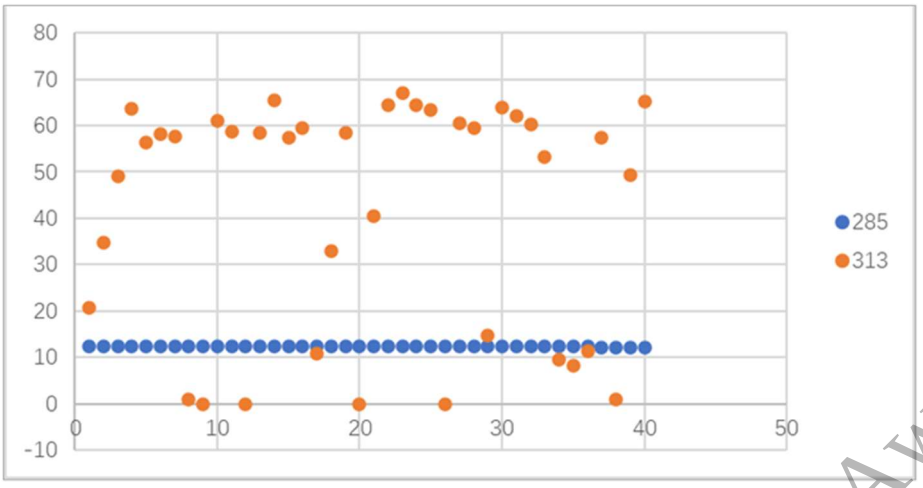


Fig 2.3 285 and 313 lock hopper liquid level raw data

For example, in Fig 2.3, lock hopper liquid level data in 313 sample data are fluctuating violently and there are negative values. These values do not comply 3σ rule. Compared to data in 215 data sample, these outliers make it difficult to analyse the effect of this manipulating variable on RON loss. Hence, these outliers are deleted (Fig 2.4)

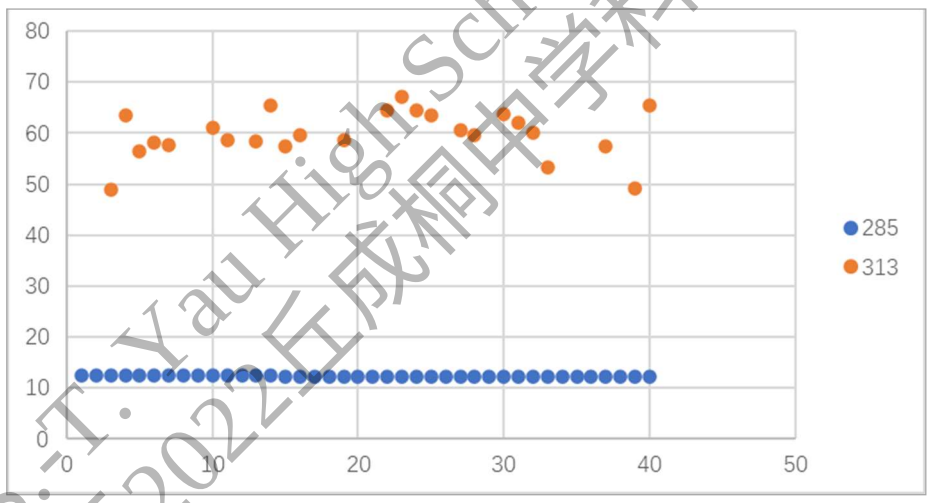


Fig 2.4, 285 and 313 lock hopper liquid level processed data

Using these steps, I amended different manipulating variables at different timing, the processed 285 and 313 sample data are appended in Appendix 1 excel file.

2.4 The result of data pre-processing

Through these steps, I pre-processed the data, the number of amended data is as shown below.

Sample number	The number of incomplete data	The number of data deleted by f(xi)	The number of data deleted by 3σ rule
285	440	0	0
313	362	1147	17

3 Methods to identify main variables for RON loss model

3.1 Analysis

RON loss is affected by both the raw material as well as conditions of machines and devices. It is also affected by Adsorption properties to be generated and Regenerating adsorbability. Hence, the variables that contribute to RON loss is complex and excessive number of manipulating variables will affect the complexity of the model. It is also not suitable for the optimisation of the model. Hence, I will simplify the variables and focus on those that have more significance to affect RON loss. Manipulating variables' dimension will be reduced to get the main variables and factors that affect RON loss. The number of main variables will be less than 30. To make the model more simplified and concise, I aim to find independent and representative variables. "Independent" means that the way the manipulating variables affect RON loss should be distinct from other variables and "representative" means that the variables need to be the most significant factor that contribute to RON loss among similar variables that affect RON loss the same way.

3.2 Plan to decipher main variables

In the RON loss modelling process, it is crucial to obtain main variables and factors to create an accurate RON loss prediction model. Hence, I used 3 methods to select variables and compare with each other to obtain a relatively better main variables selection method. The selection process is as shown in Fig 3.1

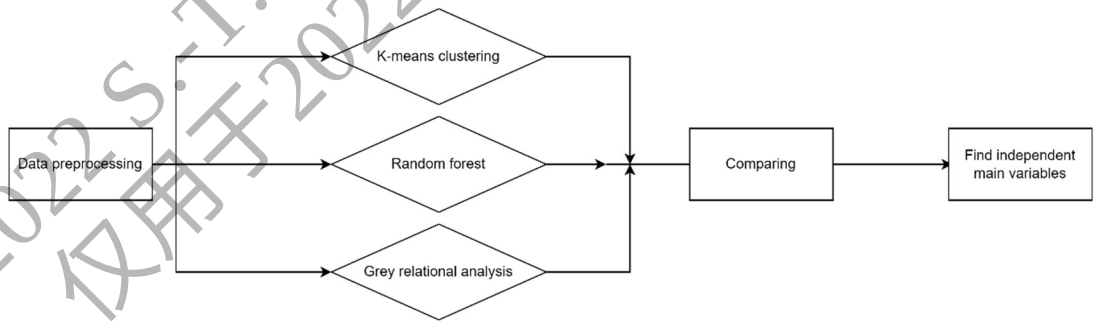


Fig 3.1

1st Method: Grey relational analysis(Kuo et al., 2008): RON loss as Mother sequence and manipulating variables as subsequence to compare and analyse will enable me to get the degree of correlation of each variable with RON loss. Yet, there are some limitations: The degree of correlation obtained by grey correlation analysis may be very close, and

representative data cannot be selected. The important variables obtained by grey correlation analysis are not independent, and the manipulating variables have similar characteristics.

2nd method: K-means clustering (MacQueen, 1967): It is widely used in variable selection and feature extraction, but it has excellent clustering effect. However, in actual operation, it also has certain defects: the number k of clusters is difficult to determine. It is very sensitive to noise and abnormal points, and the processing effect is not good for data with large difference in class size.

3rd method: Random forest model(By: IBM Cloud Education): The random forest model is a bagged ensemble machine learning method based on decision tree, which has higher generalization ability than the single classifier model. The random forest is suitable for processing samples with a large number of features. Whether the useful features can be automatically selected is very consistent with the given data set. Therefore, the random forest can be used to filter the main variables.

3.2.1 Grey relational analysis

Grey relational analysis is a method of calculating grey relational degree and determining the contribution measure of the main behaviour of the system or the influence degree between the system factors. The measure of correlation between two factors or between two systems is called grey correlation degree. The change trend, size and speed of grey correlation degree reflect the relative change of factors in the process of system development. In the process of development, when the relative changes of two factors or systems have basically the same trend of change, the two factors have a greater degree of grey correlation; otherwise, they have a smaller degree of grey correlation (Wu, 2002). In the data processing, through the combination of quantitative and qualitative analysis, the original data is directly used for calculation, which is more reliable, the evaluation results are more objective and accurate, and it also has a good performance in solving related problems such as the difficulty of accurate quantification and statistics of evaluation standards.

In this scheme, the grey correlation analysis method is used to judge the correlation degree between the operating variable and the octane number loss by comparing the similarity degree of the curve geometry for the operating variable sequence and the octane number loss sequence. The correlation degree is an objective existence between the variables, but not a strictly corresponding dependency degree in quantity.

Steps:

1. Let RON loss sequence $X_0 = [x_0(1), \ldots\ldots, x_0(n)]$ as the mother sequence and manipulating variables' sequence as subsequence $(i \in [1, 367])$. We need to eliminate the different dimensions of the original data and convert them into data that can be compared with each other, as shown in 3-1:

$$x_0^0 = \left[x_0^0(1), \ldots, x_0^0(n)\right] = \left[x_0(1)d, \ldots, x_0(n)d\right]$$
$$x_1^0 = \left[x_1^0(1), \ldots, x_1^0(n)\right] = \left[x_1(1)d, \ldots, x_1(n)d\right] \quad (3\text{-}1)$$

of which $x_i^0(k) = x_i(k)d = x_i(k) - x_i(1)$

2. The grey correlation between the sequence for RON loss and manipulating variables is shown in equation 3-2

$$\gamma\left(X_0^0, X_1^0\right) = \frac{1 + |s_0| + |s_1|}{1 + |s_0| + |s_1| + |s_1 - s_0|} \quad (3\text{-}2)$$

In the equation:

$$|s_0| = \left|\sum_{k=2}^{n-1} x_0^0(k) + \frac{1}{2}x_0^0(n)\right|,$$
$$|s_1| = \left|\sum_{k=2}^{n-1} x_1^0(k) + \frac{1}{2}x_1^0(n)\right|,$$
$$|s_1 - s_0| = \left|\sum_{k=2}^{n-1}\left(x_1^0(k) - x_0^0(k)\right) + \frac{1}{2}\left(x_1^0(n) - x_0^0(n)\right)\right|.$$

When simplified, we get equation 3-3:

$$\gamma\left(X_0^0, X_1^0\right) = \frac{1}{1 + |s_1 - s_0|} \quad (3\text{-}3)$$

3.2.2 K-means clustering

K-means clustering is a kind of unsupervised learning. It aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean. It is important to decide an appropriate k value. In most cases, with the increase of the value of K, the distortion function will continue to decrease, and there is no obvious inflection point. Therefore, the selection of K value needs to be based on practical experience and the purpose of cluster analysis. After determining the value of the cluster number k, K classes are randomly selected as the initial centers for clustering, the distance between each class and the initial center is calculated, and it is classified into the nearest initial center to form K clusters, and the centroids of these K clusters are recalculated. Through multiple iterations, when the

centroid positions meet certain conditions, the iteration ends, and the classification is completed.

Steps:

1. Use Principal component analysis(PCA) to get an appropriate k value
2. Use k number of random manipulating variables as the initial centroids for centroid based clustering
3. For the rest of the manipulating variables, calculate their distances with different centroids and assign them to the closest clusters. Euclidean distance is usually applied to obtain the distance between operating variables to centroid, as shown in equation 3-4

$$D(x_i, x_j) = \left\| x_i, x_j \right\| = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \quad (3\text{-}4)$$

4. For clustered manipulating variables, recalculate the centroids for each cluster
5. Repeat steps above until centroids no longer change
6. Choose the manipulating variables that are closest to centroids in each cluster as the main variables.

3.2.3 Choosing variables for random forest

The random forest model is a bagged ensemble machine learning method based on decision tree, which has higher generalization ability than the single classifier model. The steps required for random forest to generate main variables are shown in Figure 3-2.

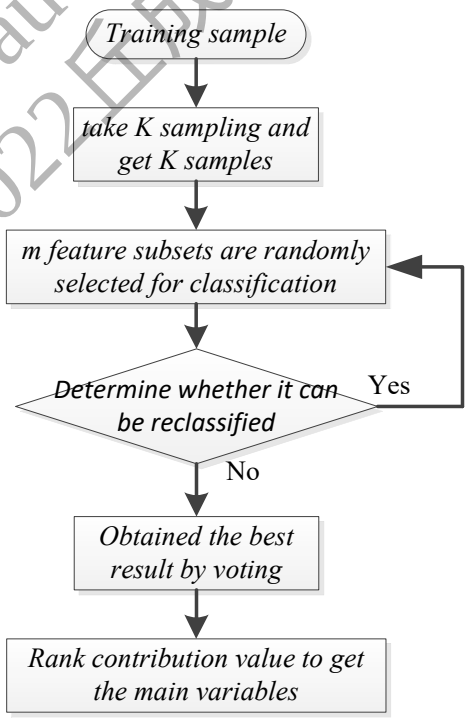


Fig 3.2

Steps:

Step 1: set the number of generated classification regression trees in the random forest as K value, and the sampling without return will lead to the difference of training samples. Therefore, we use the bagging algorithm to sample with return and generate the self-help samples of the operation variable set.

Step 2: each classification and regression tree is generated based on a self-help sample. The original data set is the set of operation vectors obtained after gray-scale analysis and screening. M suitable operation variables are selected from the operation variables. The value of M is positively related to the correlation and classification ability of the decision tree, and the correlation of the decision tree is proportional to the forest classification error rate, However, the classification ability of the decision tree is inversely proportional to the error rate. Therefore, how to weigh the most appropriate m value and select the feature node with the most classification ability from the M features to split.

Step 3: repeat step 2 until the decision tree can no longer split into a forest (each decision tree grows as much as possible), and get the total number of trees in the random forest nTree.

Step 4: K trees in the model of K classification regression trees vote to get the final result, and the classification with the largest number of votes is the final prediction.

Step 5: the relative importance of the prediction of the target operating variables can be evaluated by the depth of the decision nodes in the tree. That is, the operational variables at the top of the decision tree make a greater contribution to the final prediction decision of the sample (the proportion of the contributed samples); Therefore, the contribution value of each operating variable to the final prediction can be used to evaluate the importance of the operating variable and sort to obtain the main variables for modeling.

In the raw data, there are some data sample that are not used during bootstrapping. These data samples are called Out-of-band data(OOB).The out of bag error as errorOOB1, can be calculated using the corresponding OOB. At the same time, noise interference is randomly added to the out of bag data OOB, and the out of bag error is calculated again, which is recorded as errOOB2. When noise is added to an operation variable, the out of bag accuracy is greatly reduced, which proves that the operation variable has a great impact on the

classification result, the importance IMP (importance) value of the operation variable is high. IMP's equation are shown in equation 3-5:

$$IMP = \frac{1}{Ntree}\sum(errOOB_2 - errOOB_1) \quad (3\text{-}5)$$

3.3 Results of different models

3.3.1 Grey relational analysis

By using grey relational analysis, we are able to get rid of certain manipulating variables' sequence. From the line chart, we can see a curve that is relatively parallel to x-axis. By analysing parts of the curve that deviate greatly from the straight line, we are able to get main variables and make our model less complex. However, as mentioned, it reflects the relationship between manipulating variables and mother sequence, but it fails to consider the correlation between manipulating variables. Hence, some important data points may be wrongly neglected or manipulating variables with similar features are kept. Thus, grey relational analysis is unable to produce distinct and useful main variables that affect RON loss from 354 manipulating variables.

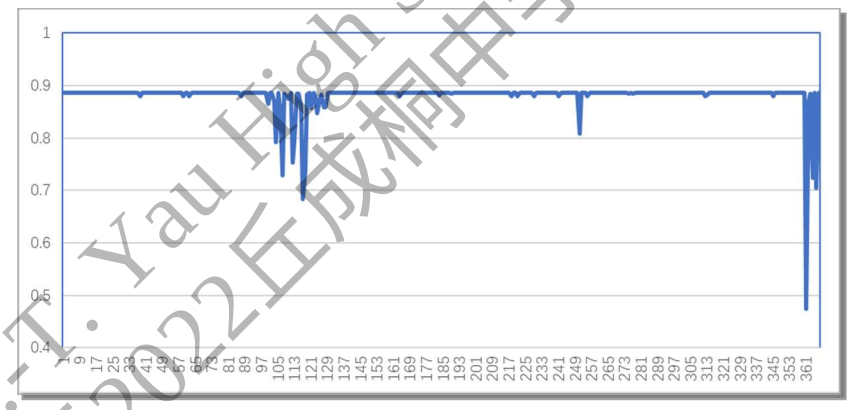


Fig 3.3

3.3.2 K-means clustering

K-means clustering enables us to separate 354 manipulating variables to 20 relatively distinct clusters. As shown in Fig3.4

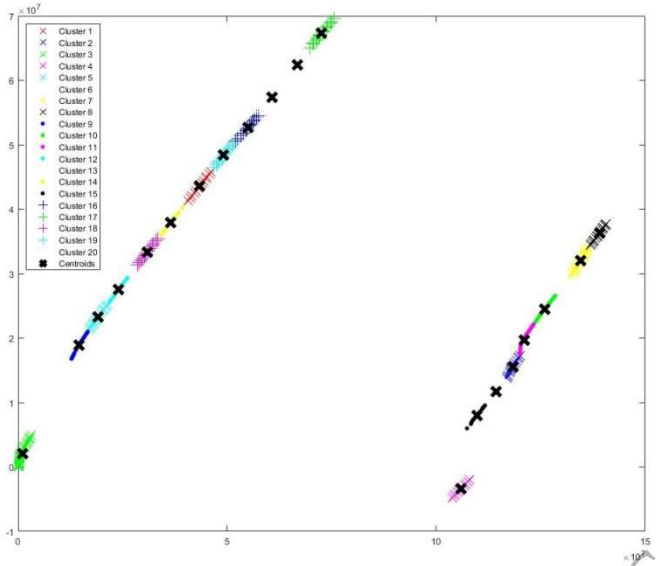


Fig3.4

Representative manipulating variables are chosen from each cluster as main variables, as shown in Chart 3.1

Serial number of manipulating variables in Annex A	Manipulating variables' name
78	Reactor inlet temperature
174	D-107bottom pressure
294	K-101B intake pressure
132	E-101shellside exit temperature
219	air heater exit temperature
320	E-205 shellside exit temperature
261	P-105A/B outlet header flow
22	Bottom reactor temperature
37	Flow of refined petrol
240	R-102 Bottom pressure
92	Reducer temperature
7	aromatic hydrocarbons, v%
280	K-103A inflow pressure
52	Flow of recycled water entering the device
197	Reproducer bottom

	temperature
65	% level of raw material in the column
152	D-123 steam gas flow at the exit

Chart 3.1

Because of the large number of features of the original data samples and the large difference in the category size, the K-means clustering effect is not so well.

3.3.3 Random Forest modelling

Using random forest calculation, we obtain 15 independent and representative manipulating variables as shown in Fig 4.5

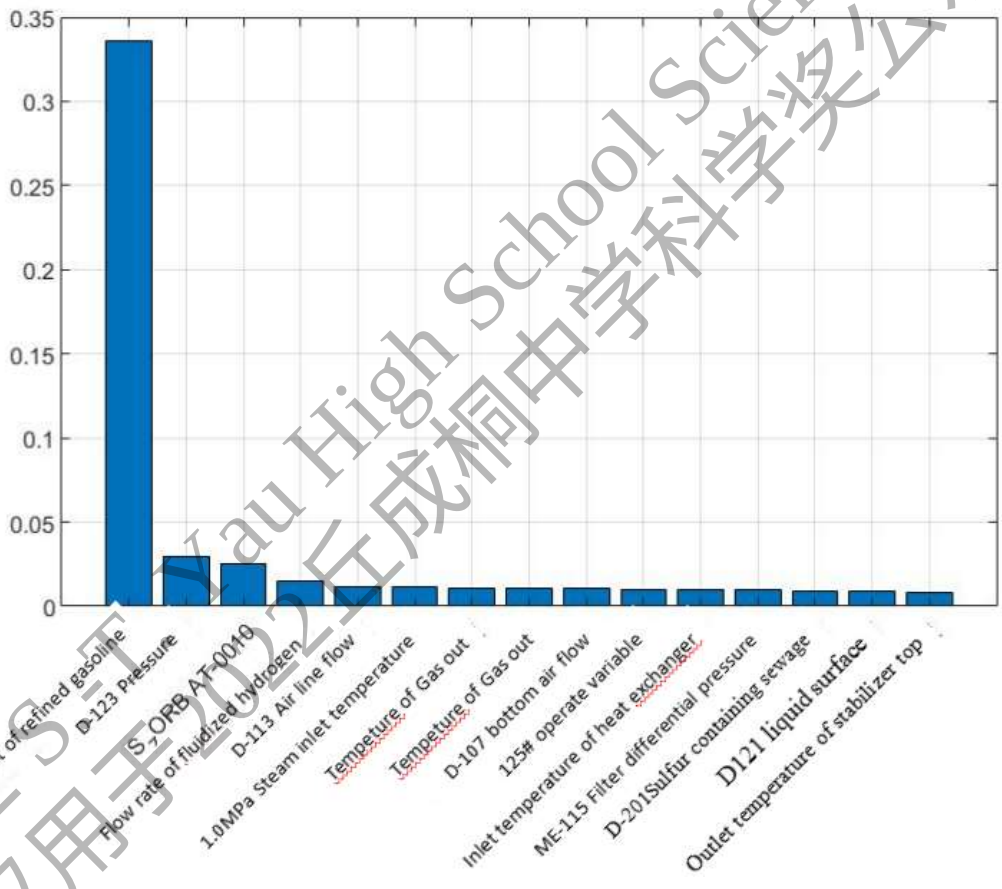


Fig 4.5

3.3.4 Results analysis

Since main variables obtained from grey relational analysis is not representative, K-means clustering effect is poor due to the large number of features and the large difference of category scale. Hence, main variables are obtained from random forest modeling. They are shown below in Chart 3.2

Serial number of manipulating variables in Annex A	Manipulating variables' name	Significance
38	Sulfur content in refined petrol	0.3361
153	D-123 pressure	0.0292
337	S_ZORB AT-0010	0.0255
20	Flow rate of fluidized hydrogen	0.0146
160	D-113 Air line flow	0.0116
50	1.0MPa steam gas inlet temperature	0.0112
206	Temperature of Gas out	0.0111
170	D-107 bottom air flow	0.0108
125	125# operate variable	0.0107
71	Inletemperature of heat exchanger	0.0101
264	ME-115 filter differential pressure	0.0098
348	D-201sulfur containing sewage	0.0096
88	D121 liquid surface	0.0088
185	Outlet temperature of stabilizer top	0.0087
13	Coke,wt%	0.0083
11	RON value	0.0081

Chart 3.2

4 Building RON loss model

4.1 Question analysis

The model was designed to lower the RON loss while ensuring the effectiveness of desulfurization process. Hence, we want to use manipulating variables to predict RON values and calculate RON loss to optimise the industrial process of fuel refinement. This model not only predicts RON loss but also calculates the amount of sulfur content present.

After data pre-processing, determined main variables are the input of the model. We will use improved Squeeze and Excitation Network(SE-Net) combined with the prediction network to predict RON value and calculate RON loss, then verify the predicted results to achieve a relatively effective prediction model.

4.2 Constructing the Model

From part 2's results from random forest modelling, we can see that selected main variables have different significance in determining RON value. Hence, when building our RON prediction, we need to modify the SE-Net and the prediction network as our data is one dimensional. Then we need to first construct a SE-Net to weigh the significance of main variables to determine significance of the effect each main variable has on the predicted result. Lastly, weighted variables are input into the prediction network to get the predicted sulfur content and RON value, and build the loss function between the prediction result and the original sulfur content and octane number of the sample, and optimize the model through back propagation. The Neural network structure is as shown below in Fig 4.1

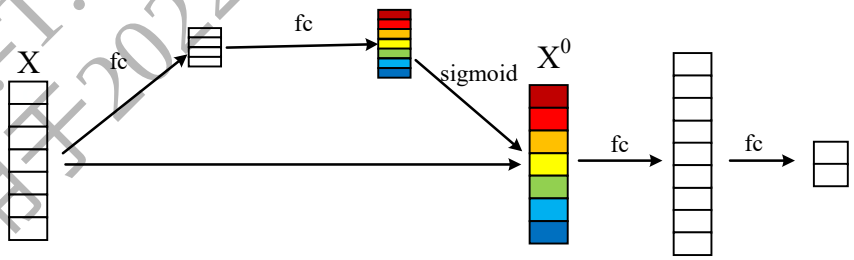


Fig4.1

4.2.1 SE-Net

SE-Net (Hu et al., 2018) is a convolutional neural network, comprising of Squeeze, Excitation and Reweight

Squeeze operate: the squeeze operation was to compress the features in the spatial dimension to reduce the spatial dimension. However, this study is a one-dimensional data, so our model

deletes the original se net pooling layer, that is, does not perform the squeeze operation, so as to better apply it to this one-dimensional data model.

Excitation operate: It is similar to the mechanism of the middle gate of the cyclic neural network: first, the model uses two full connection layers to form the bottleneck structure to construct the correlation between the operating variables. Then, we reduce the length of the operating variables from 14 to 1 / 2. After passing through a relu activation function, we pass through a full connection layer, which makes our model have better nonlinear fitting and greatly reduces the number of operating variables and the amount of calculation.

Reweight operate: after the Excitation operation, a sigmoid activation function generates a normalized weight W between 0 and 1 for each operation variable, and finally, the normalized weight is weighted to each corresponding operation variable through the scale operation. The structure of SE-Net is as shown below in Fig4.2:

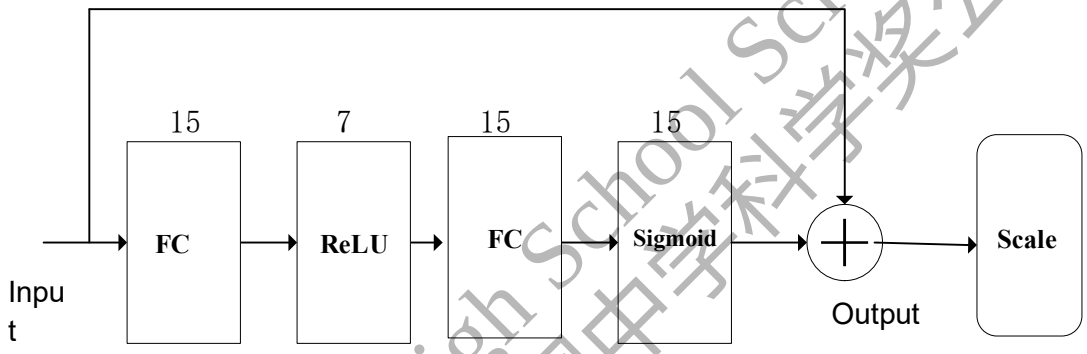


Fig4.2

4.2.2 Prediction Network

Prediction network comprises of a input layer, a hidden layer and a output layer, which can also be referred to as two fully connected layers. In a fully connected layer, every node is connected to all neural nodes in the previous layer as well as the layer that comes after. Theoratically, if there are many neurons, a fully connected layer with only one hidden layer can also fit any function.

The forward propagation of the full connection layer is a simple linear weighted summation process. The output of each node of the full connection layer can be regarded as the product of the output of each node of the previous layer and a weight W plus an offset B, as shown in Figure 4.3.

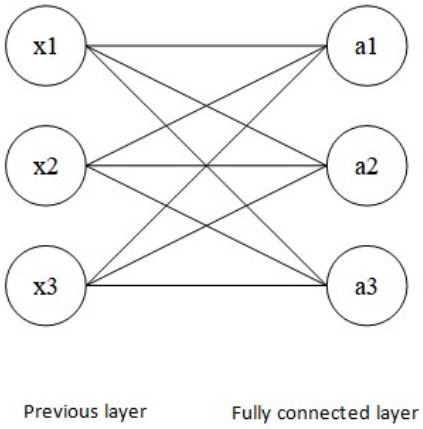


Fig 4.3

The output a and input x has the following relationship as shown in equation 4-1

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} * \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (4\text{-}1)$$

The back propagation of full connection is to determine a loss function LOSS, solve the partial derivative of LOSS to other variables to obtain the gradient, and then forward a back propagate from the output to update the weight and offset parameters, so that the predicted output can better match the actual value, and the prediction effect of the model can be more optimized.

The input of the full neural network is the operating variable parameters recalibrated by the se net neural network. The output values are predicted sulfur content

$S\_pred = (Sp_1, Sp_2 \ldots\ldots, Sp_n)$ and predicted RON loss value

$RON\_pred = (Rp_1, Rp_2, \ldots\ldots Rp_n)$

4.3 Model solving and analysis

We need to use 325 data sample's main variables as the model's input and use model to calculate predicted RON loss and sulfur content. Then we optimize the network model parameters by making the loss as small as possible to make the prediction results more accurate after multiple rounds of training.

Steps:

1. Separation of data set: randomise the processed 325 data samples' manipulating variables, using 80% of the data as training sets and 20% of the data as test data sets

2. The model's parameters are as shown below in Chart 4.1

Parameters' name	Parameters' values
EPOCH	50
SGD Learning rate	0.01
Number of training	50

Chart 4.1

Although the model is created to ensure effective desulfurization while minimising RON value loss, the main focus of the study is still RON loss. Hence, Therefore, we set weights for the influence of the predicted sulfur content and the predicted octane number on the LOSS function. The formula of the Loss value of the neural network is as follows:

$$Loss = \sum_{i=1}^{n} 0.3*(Sp_i - S)^2 + 0.7(Rp_i - R)^2 \quad (4\text{-}2)$$

Where s and R are the actual values corresponding to the predicted sulfur content and octane number in Annex I。

3. Training model. The whole model is trained in EPOCH rounds. During each round of training, the predicted values of sulfur content and octane number are obtained by inputting the values of the training set, and the predicted values and the original values corresponding to the samples are calculated by the LOSS function. The parameters of the whole network are updated by back propagation, and the parameters of the whole network are updated with the goal of reducing the LOSS value to obtain better monthly measurement results. After each round of training, calculate the prediction results of the test set.

4. For evaluation of the model fitting, we introduced relative error δ to assess the predicted result of RON loss. Relative error is the average of the mod of total sum of predicted RON values minus the total sum of actual RON values, equation is as shown in equation 4-3:

$$\delta = \frac{1}{R_i} \sum_{i=1}^{n} | Rp_i - R_i | \quad (4\text{-}3)$$

Relative error reflects the deviation of the predicted values from the actual values. Hence, the lower the δ, the better is the model. The predicted values, actual values and relative values are as shown in Fig 4.4 and 4.5

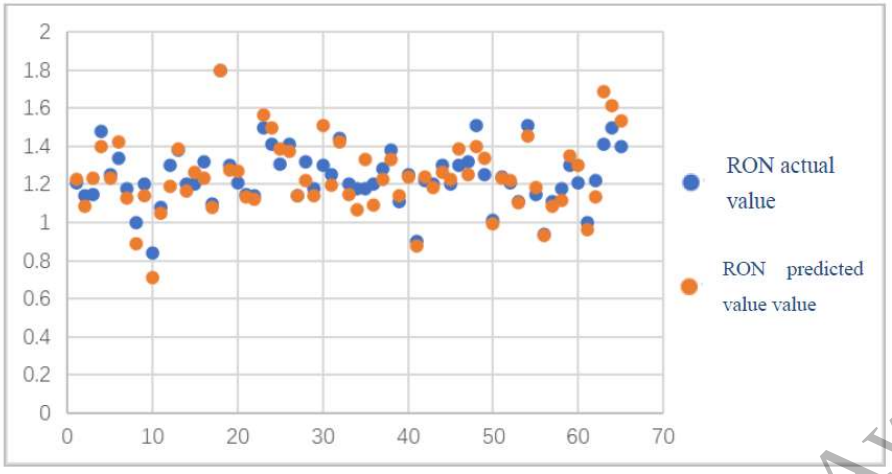


Fig4.4 comparing RON predicted values and actual values

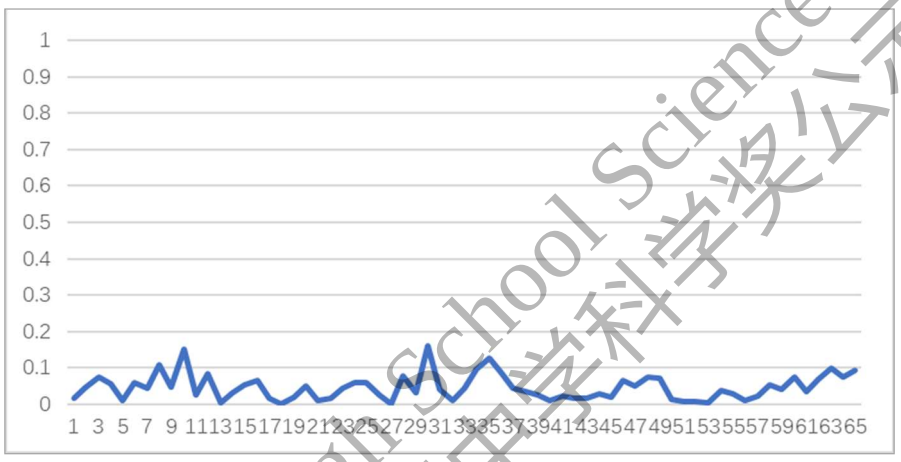


Fig4.5 RON relative error line graph

From Fig4.5 we can see that the predicted RON value deviate little from real RON value as δ in most of the time is lower than 0.1, with maximum δ being 0.19. This shows that our prediction model are able to predict RON value according to the main variables listed above.

5 Optimization of main variable operation scheme

In chapter 4 we have constructed the neural network model that is able to predict sulfur content and RON loss. In this chapter, we aim to manipulate main variables for 325 data samples to make percentage RON loss to decrease by more than 30%.

We need to modify main variables and use modified main variables to predict sulfur content and RON loss. We thus believe genetic algorithm is the best way to find optimal values for main variables to achieve our goal. Genetic algorithm used to solve optimization in the field of computer science and artificial intelligence. It uses mutation to jump out of the local optimal solution. Finally, we can optimize the main variables of the sample towards the optimal solution to improve the octane number loss.

5.1 Construction of optimizing model

Changing main variables to achieve a lower RON loss while keeping sulfur content under control is our main goal. Hence, we aim to find out which direction the main variables should be optimized. Thus we want to use genetic algorithm to find main variables' optimal values.Since the optimal solution of the main variable operation scheme is obtained, the operation optimization direction of the main variable is also obtained. The flow chart of the main variable operation scheme optimization model is shown in Fig. 5.1:

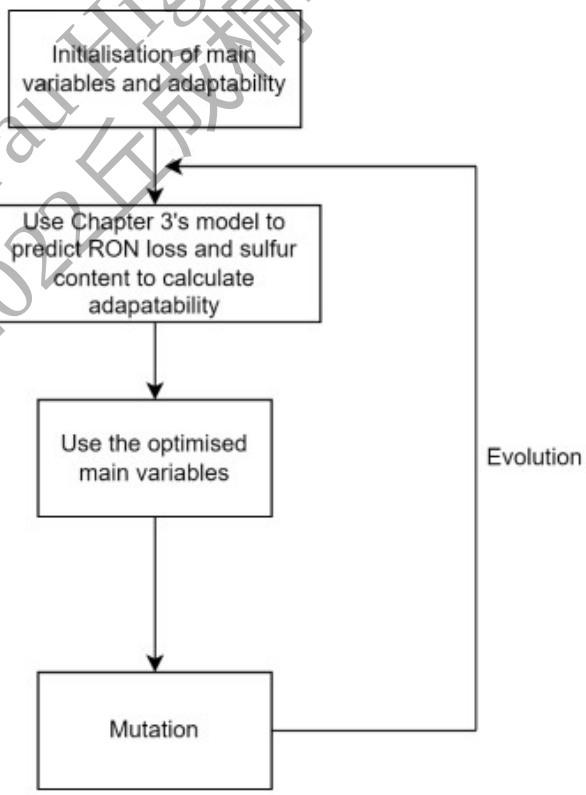


Fig5.1

Genetic algorithm is a method for solving both constrained and unconstrained optimization problems that is based on natural selection, the process that drives biological evolution. It contains crossover and mutation actions:

Crossover is a genetic operator used to combine the genetic information of two parents to generate new offspring. First, take the initial solution as the parent, and according to the principle of cross mutation and mutation mutation, that is, exchange the number of two different positions and change the number of one position, and then generate the offspring. The Order Crossover (OX) operator is used in the algorithm. The crossover process of this operator is as follows:

1. Select a chromosome from a chosen pair of chromosomes as the father and randomly generate a sequence from start to end as a child. In the example, we generate chromosomes and named one of it as Father1. Then we locate the start at the third position, end at the sixth position, Father1 generates Child1 from the start to the end, as shown in Fig 5.2

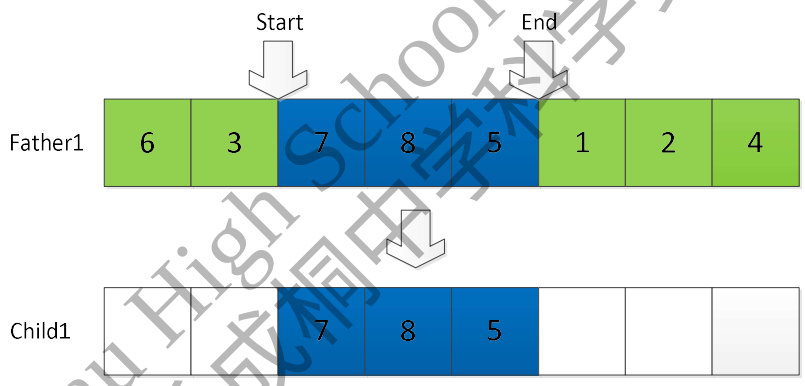


Fig5.2

2. Use another string of chromosome as the mother (Mother1), and add in the rest of the mother's genetic code according to the original sequence as shown in Fig5.3

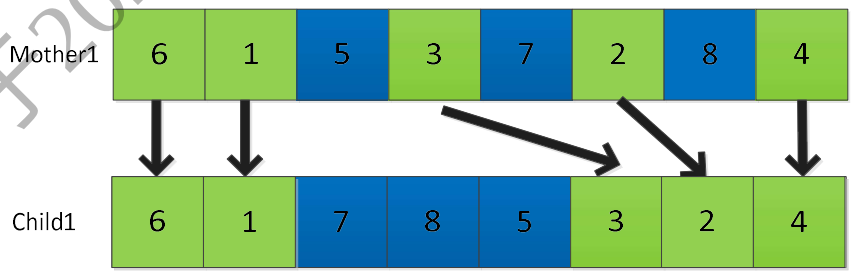


Fig5.3

3. A Child1 will be generated. Then the father becomes the mother and the mother chromosome becomes the father to generate Child2 by repeating steps above, as shown in Fig5.4

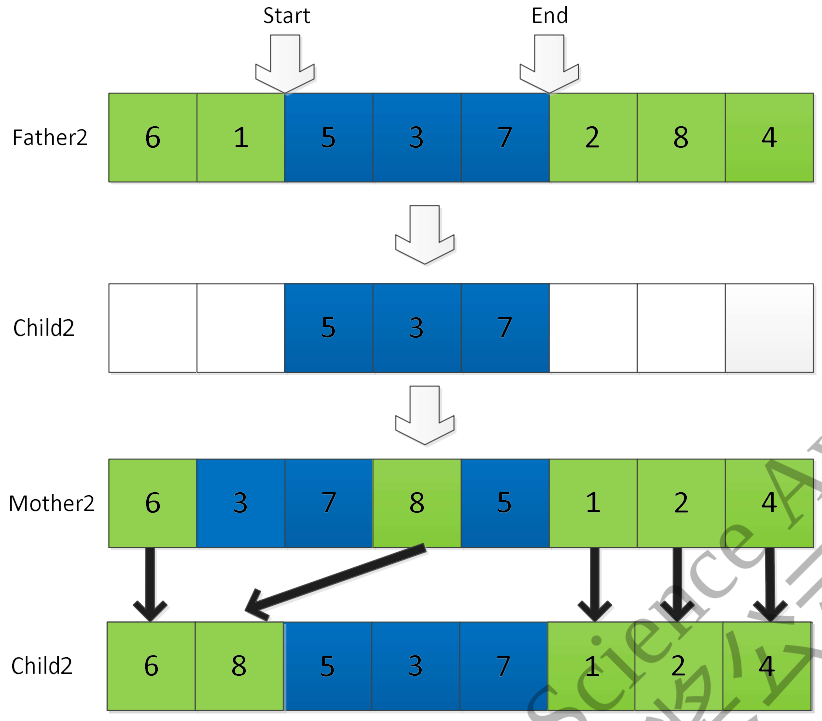


Fig 5.4

Mutation is to generate characteristics that did not exist before. However, the possibly for a mutation to occur is very low, therefore, the population is still developing in a relatively uniform direction. The basic methods of mutation includes base substitutions, deletions and insertions. We will use Position Variation Method in our algorithm, selecting two random locations and exchanging their values, as shown in Fig5.5.

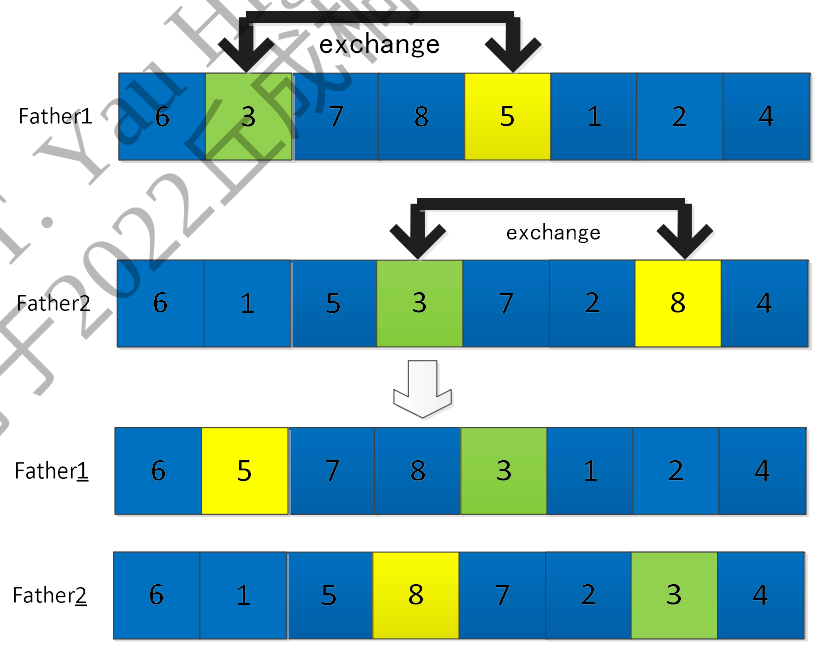


Fig 5.5

First, we need to initialize the main variable values of each individual (i.e., each sample). We need to establish a fitness to indicate whether each individual needs to survive or be eliminated. The fitness conditions are as follows:

$$\begin{cases} S\_pred \le 5\mu g / g \\ \dfrac{R - RON\_pred}{R} > 30\% \end{cases}$$

We obtain the predicted values of sulfur content and octane number loss through the prediction model established in question 3 according to the main variable values of the sample, and calculate the fitness of the sample according to the predicted values, as shown in formula 5-1:

$$fitness = \frac{RON\_pred - RON\_best}{R} \quad (5\text{-}1)$$

Where RON_best in the current state is the minimum value in the process of model optimization.

Genetic algorithm is essentially based on the theory of natural selection. Select this operation is to select individuals with excellent characteristics from a given population, and then take them as samples to lay a foundation for deriving the next generation. By comparing the fitness of each sample, we can select the optimal solution of the current main variable, the optimization direction of our main variable operation scheme.

Mutation is the way in which the main variable changes. Through different changes, we can find the operation scheme of the main variable under the conditions.

5.2 Model solving and analysis

We use algorithm to simulate evolution by making main variables as chromosomes and constantly mutate them. This helps us to obtain different directions of evolutions and find ways that are most adaptable to the environment. Our fitness is set according to the conditions in the text, hence we are able to obtain main variables that are most adaptable to the given environment via multiple trainings.

1. Set the parameters. Set the model cycle to find the best quality for 100 times. and make the possibility of mutation to be 0.5.
2. After main variables mutated, sulfur and RON loss are predicted from the prediction model. Then the adaptability is calculated. We compare the adaptability find the best main variables.
3. Iterate step 2 for 100 times and obtain the best main variables.
4.

For 325 data samples, the rate of which RON loss decreases by more than 30% is shown in the Chart 5-1

Number of data sample that meet the requirement	216
Percentage of data sample that meet the requirement	66.5%

Chart5-1

The optimised RON loss predicted values and actual values are shown in Fig 5.6 illustration

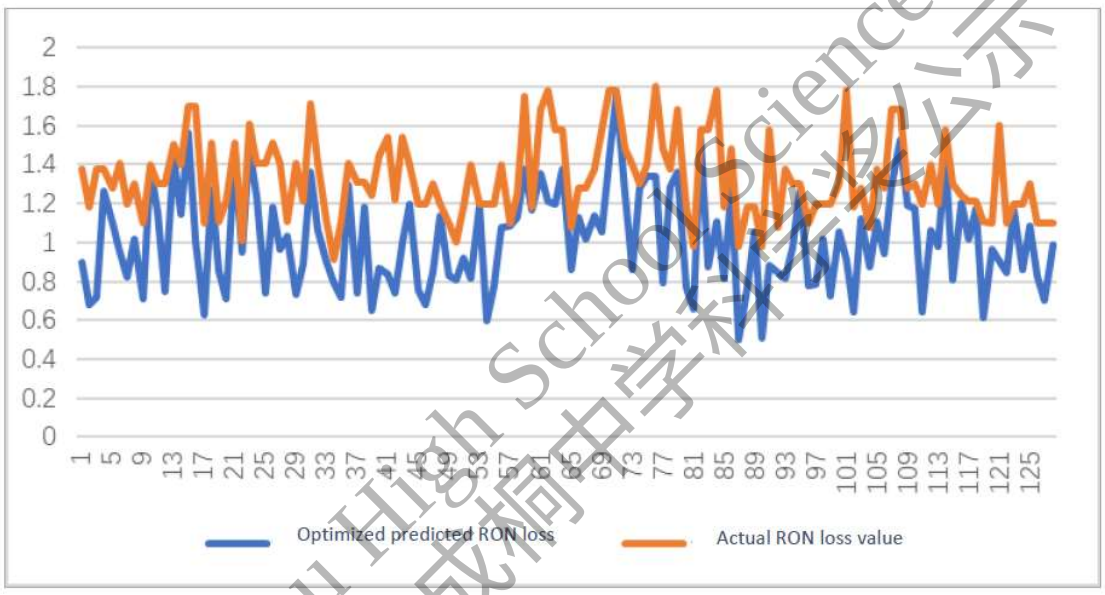


Fig 5.6 Optimised predicted RON loss compared to actual RON loss

From Fig 5.6, after manipulation of main variables with the genetic algorithm, RON loss produced are generally smaller compared to RON loss produced from original variables. This shows that our model allows us to optimise main variables effectively to lower RON loss.

6 Model Visualisation

We optimised sample data 133 according to the stated variables manipulation range and recorded down how RON value, sulfur content and RON loss has changed.

Variables	1	2	3	4	5	6	7
coke	2.53	2.53	2.53	2.53	2.53	2.53	2.53
Flow rate of fluidized hydrogen	648.4958	698.4958	748.4958	748.4958	748.4958	748.4958	748.4958
Sulfur content in refined gasoline	-0.15295	0.84705	1.84705	1.84705	1.84705	1.84705	1.84705
1.0MPa Steam inlet temperature	186.3764	187.3764	188.3764	189.3764	190.3764	191.3764	192.3764
Inlet temperature of heat exchanger	54.78854	55.78854	56.78854	57.78854	58.78854	59.78854	60.78854
D121 liquid surface	49.97578	49.97578	49.97578	49.97578	49.97578	49.97578	49.97578
125#Operation variable	44.75582	34.75582	34.75582	34.75582	34.75582	34.75582	34.75582
D-123 pressure	0.350085	0.350085	0.350085	0.350085	0.350085	0.350085	0.350085
D-113 air line flow	133.7167	123.7167	123.7167	123.7167	123.7167	123.7167	123.7167
D-107 bottom air flow	14.08572	19.08572	24.08572	24.08572	24.08572	24.08572	24.08572
Outlet temperature of stabilizer top	56.94665	57.94665	58.94665	59.94665	60.94665	61.94665	61.94665
Temperature of gas out	719.6724	718.6724	717.6724	716.6724	715.6724	714.6724	713.6724
ME-115 Filter differential pressure	5.748454	6.748454	7.748454	7.748454	7.748454	7.748454	7.748454
S_ZORBAT-0010	0.557532	0.657532	0.657532	0.657532	0.657532	0.657532	0.657532
D-201Sulfur containing sewage	0	10	20	30	40	50	60

By ensuring that sulfur content is lower than 5ug/g, RON value gradually increases. RON value change is shown in fig 6.1 and sulfur content value change is shown in Fig 6.2

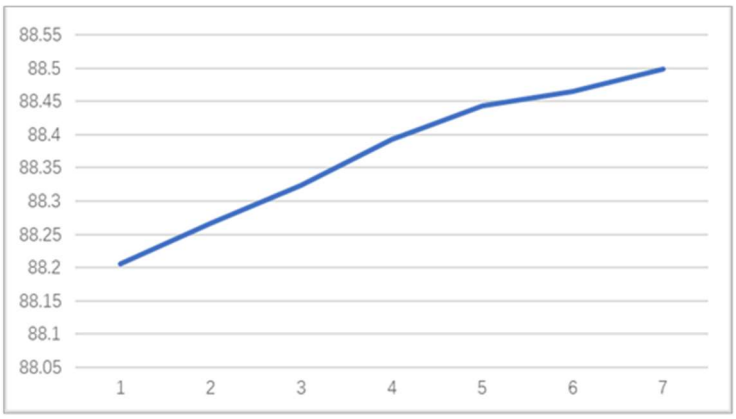


Fig 6.1

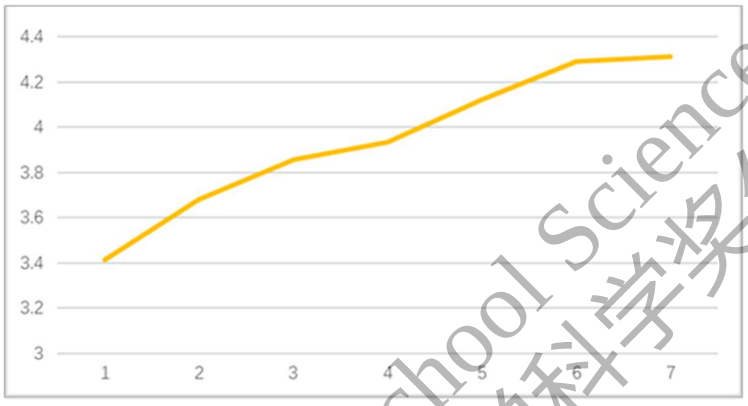


Fig 6.2

From fig 6.1 and fig 6.2, we can observe that as we optimise the main variables, both RON value and sulfur content value are increasing. Yet, sulfur content is constantly kept below 5. Through our optimisation model, we are able to ensure that RON value can be increased to economic efficiency with the given requirements.

7 Bibliography

By: IBM Cloud Education. (n.d.). *What is Random Forest?* IBM. Retrieved August 23, 2022, from https://www.ibm.com/cloud/learn/random-forest

Fang, W. (2016). *数据挖掘技术在催化裂化 MIP 工艺产品分布优化中的应用研究*. 数据挖掘技术在催化裂化 MIP 工艺产品分布优化中的应用研究--《华东理工大学》2016 年硕士论文. Retrieved August 23, 2022, from https://cdmd.cnki.com.cn/Article/CDMD-10251-1016097657.htm

Han, M. (2017). *A variable selection algorithm based on Improved Grey Relational Analysis*. A variable selection algorithm based on Improved Grey Relational Analysis--《control and decision》2017 年 09 期. Retrieved August 23, 2022, from https://en.cnki.com.cn/Article_en/CJFDTotal-KZYC201709015.htm

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/cvpr.2018.00745

Kuo, Y., Yang, T., & Huang, G.-W. (2008). The use of grey relational analysis in solving multiple attribute decision-making problems. *Computers & Industrial Engineering*, *55*(1), 80–93. https://doi.org/10.1016/j.cie.2007.12.002

Lu, X., Wang, X., Yang, Y., & Xue, J. (2021). The optimization model for reducing Ron Loss in gasoline refining process. *Geofluids*, *2021*, 1–10. https://doi.org/10.1155/2021/5520942

MacQueen, J. (1967, January 1). *Some methods for classification and analysis of Multivariate Observations*. Project Euclid. Retrieved August 23, 2022, from https://projecteuclid.org/ebooks/berkeley-symposium-on-mathematical-statistics-and-probability/Proceedings%20of%20the%20Fifth%20Berkeley%20Symposium%20on%20Mathematical%20Statistics%20and%20Probability,%20Volume%201:%20Statistics/chapter/Some%20methods%20for%20classification%20and%20analysis%20of%20multivariate%20observations/bsmsp/1200512992

Singapore Gov. (n.d.). *Air Pollution Regulations*. National Environment Agency. Retrieved August 23, 2022, from https://www.nea.gov.sg/our-services/pollution-control/air-pollution/air-pollution-regulations#:~:text=Vehicular%20Fuel%20Quality,see%20link%20for%20more%20information).

Wikimedia Foundation. (2022, July 26). *Data pre-processing*. Wikipedia. Retrieved August 23, 2022, from https://en.wikipedia.org/wiki/Data_pre-processing

Wu, H.-H. (2002). A comparative study of using Grey Relational Analysis in multiple attribute decision making problems. *Quality Engineering*, *15*(2), 209–217. https://doi.org/10.1081/qen-120015853

Yang, B. (2018). *工业过程变量间动态时延挖掘方法与应用*. 工业过程变量间动态时延挖掘方法与应用--《北京化工大学》2018 年博士论文. Retrieved August 23, 2022, from https://cdmd.cnki.com.cn/Article/CDMD-10010-1018322277.htm

Zhang, S. (2019). *CART 分层变量的选择*. Cart 分层变量的选择-手机知网. Retrieved August 23, 2022, from https://wap.cnki.net/touch/web/Dissertation/Article/10614-1020716552.nh.html

8 Appendix

8.1 Data preprocessing

```
[num313,txt313,~]=xlsread("313.xlsx");
[mm313,txtmm,~]=xlsread("maxmin.xlsx");
sumall = zeros(1,354);%sum of every row
avgall = zeros(1,354);%average of every row
numall = zeros(1,354);%number of 0 in every row
afterdeal = zeros(40,354);%processed data
sum1 = zeros(1,354);%sum after step 1
avg1 = zeros(1,354);%average after step 1
sum2 = zeros(1,354);%sum after step 2
avg2 = zeros(1,354);%average after step 2
numof0 = zeros(1,354);%number of 0 deleted (1)
numof2 = zeros(1,354);%number of points deleted (2)
numof3 = zeros(1,354);%number of points deleted (3)
avg3 = zeros(1,354);
sum3 = zeros(1,354);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  step 1
for i =1 : 354
   a = num313(:,i);
   len = length(a);
   num0 = 0;
   for j =1 :len
      if a(j) == 0
         num0 = num0 + 1;
         afterdeal(j,i) = 0;
      else
         afterdeal(j,i) = num313(j,i);
      end
   end
   numof0(1,i) = num0;%%%%%%number of points deleted
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%   step 2
for i = 1 : 354
   a = afterdeal(:,i);
   len = length(a);
   for j = 1: len
```

```
        sum1(1,i) = sum1(1,i) + afterdeal(j,i);
    end
end
for i = 1:354
    a = afterdeal(:,i);
    len = length(a);
    n = nnz(afterdeal(:,i));%!number of 0
    avg1(1,i) = sum1(1,i)/n(1,1);
    num2 = 0;
    for j = 1 : len
        if a(j)<avg1(1,i)-0.2*abs(avg1(1,i))||a(j)>avg1(1,i)+0.2*abs(avg1(1,i))
            afterdeal(j,i) = 0;
            num2 = num2 +1;
        end
    end
     numof2(1,i) = num2;%%%%%%number of points deleted
end
for i =1:354
    a = afterdeal(:,i);
    len = length(a);
    n = nnz(a);%!number of zero
    for j = 1 :len
        sum2(1,i)= sum2(1,i) + a(j);
    end
    avg2(1,i) = sum2(1,i) / n;
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% step 3
seta = zeros(1,354);
for i = 1 : 354
    a = afterdeal(:,i);
    len = length(a);
    sumvv = 0;
    n = nnz(afterdeal(:,i));%!number of zero
    for j = 1: len
        if a(j) ~= 0
            v = a(j) - avg2(i);%avgall average of all manipulating variables
            sumvv = sumvv + v*v;
        end
    end
```

```
    seta(1,i) = sqrt(sumvv/(n - 1));
end
for i = 1:354
    a = afterdeal(:,i);
    len = length(a);
    num3 = 0;
    for j = 1 : len
        if a(j)~= 0
            vb = a(j,1) - avg2(i);
            if vb>3 * seta(1,i)
                num3 = num3 + 1;
                afterdeal(j,i) = 0;
            end
        end
    end
     numof3(1,i) = num3;%%%%%%number of points deleted
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%find average
for i =1:354
    a = afterdeal(:,i);
    len = length(a);
    n = nnz(a);%!number of 0
    for j = 1 :len
        sum3(1,i)= sum3(1,i) + a(j);
    end
    avg3(1,i) = sum3(1,i) / n;
end
avg3 = double(avg3);
```

8.2 Selecting main variables

```
clc;
close;
clear all;
[numall,txtall,~]=xlsread("all_s2.xlsx");%325 * 368
x0 = numall(:,10);%The reference seqeunce is RON loss
X = numall();
```

```
X(:,10) = [];%Data set except RON loss



[coeff,score,latent]= pca(X);
cums=cumsum(latent)./sum(latent);
rdata=X*coeff;
%randomly obtain 150 points
opts = statset('Display','final');
%use k means function
%X N*P matrix
%Idx N*1 vector, store cluster numbering
%Ctrs K*P matrix, store k's centers' locations
%SumD 1*K sum vector,store all cluster points' distance from closest centre of mass
%D N*K matrix£¬store all points from all centre of mass
X = rdata;
[Idx,Ctrs,SumD,D] = kmeans(X,20,'Replicates',100,'Options',opts);
%draw out cluster 1¡£X(Idx==1,1), is the first point coordinate for the
%first cluster£»X(Idx==1,2)second point for the second cluster
plot(X(Idx==1,1),X(Idx==1,2),'rv','MarkerSize',14)
hold on
plot(X(Idx==2,1),X(Idx==2,2),'bv','MarkerSize',14)
hold on
plot(X(Idx==3,1),X(Idx==3,2),'gv','MarkerSize',14)
hold on
plot(X(Idx==4,1),X(Idx==4,2),'mv','MarkerSize',14)
hold on
plot(X(Idx==5,1),X(Idx==5,2),'cv','MarkerSize',14)
hold on
plot(X(Idx==6,1),X(Idx==6,2),'wv','MarkerSize',14)
hold on
plot(X(Idx==7,1),X(Idx==7,2),'yv','MarkerSize',14)
hold on
plot(X(Idx==8,1),X(Idx==8,2),'kv','MarkerSize',14)
hold on
plot(X(Idx==9,1),X(Idx==9,2),'b.','MarkerSize',14)
hold on
plot(X(Idx==10,1),X(Idx==10,2),'g.','MarkerSize',14)
hold on
plot(X(Idx==11,1),X(Idx==11,2),'m.','MarkerSize',14)
```

```
hold on
plot(X(Idx==12,1),X(Idx==12,2),'c.','MarkerSize',14)
hold on
plot(X(Idx==13,1),X(Idx==13,2),'w.','MarkerSize',14)
hold on
plot(X(Idx==14,1),X(Idx==14,2),'y.','MarkerSize',14)
hold on
plot(X(Idx==15,1),X(Idx==15,2),'k.','MarkerSize',14)
hold on
plot(X(Idx==16,1),X(Idx==16,2),'bo','MarkerSize',14)
hold on
plot(X(Idx==17,1),X(Idx==17,2),'go','MarkerSize',14)
hold on
plot(X(Idx==18,1),X(Idx==18,2),'mo','MarkerSize',14)
hold on
plot(X(Idx==19,1),X(Idx==19,2),'co','MarkerSize',14)
hold on
plot(X(Idx==20,1),X(Idx==20,2),'wo','MarkerSize',14)

%draw out mid point for clusters,kx means circle
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)
```

```
plot(Ctrs(:,1),Ctrs(:,2),'kx','MarkerSize',14,'LineWidth',4)

legend('Cluster 1','Cluster 2','Cluster 3','Cluster 4','Cluster 5','Cluster 6','Cluster 7','Cluster 8',...
   'Cluster 9','Cluster 10','Cluster 11','Cluster 12','Cluster 13','Cluster 14','Cluster 15',...
   'Cluster 16','Cluster 17','Cluster 18','Cluster 19','Cluster 20','Centroids','Location','NW')

Ctrs
SumD



fprintf('result£º\n',20)
for i=1:20
   tm=find(Idx==i); %
   tm=reshape(tm,1,length(tm)); %Transfer into row vector
   fprintf('%d cluster includes %d variables, They are: %s\n',i,length(tm),int2str(tm)); %displace
clustering results
end
```

8.3 neural network

```
import torch
import torch.nn as nn

class SELayer(nn.Module):
   def __init__(self, input, hidden, output):
      super(SELayer, self).__init__()
      self.avg_pool = nn.AdaptiveAvgPool1d(1)
      self.fc = nn.Sequential(
         nn.Linear(20, 2, bias=False),
         nn.ReLU(inplace=True),
         nn.Linear(2, 20, bias=False),
         nn.Sigmoid()
      )

      self.linear1 = nn.Linear(input, hidden)
      self.relu = nn.ReLU(inplace=True)
      self.linear2 = nn.Linear(hidden, output)
```

```
    def forward(self, x):
        y = self.fc(x)
        y =  x * y.expand_as(x)

        y = self.linear1(y)
        y = self.relu(y)
        out = self.linear2(y)
        return out
    def load(self, path):
        '''
        load designated model
        '''
        self.load_state_dict(torch.load(path))
```

8.4 Genetic Algorithm for optimisation

```
import Genetic
import Fitness
import torch
import matplotlib.pyplot as plt
import numpy as np
import xlrd
from torch.autograd import Variable # get variables
from model import SELayer

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
##read excel sheet
def xlsxread(xldir, start, end):
    data = xlrd.open_workbook(xldir)#excel location
    st = data.sheets()[0]  # read the first excel sheet
    # rows = st.nrows  #st.ncols get column number,st.nrows get row number
    table = []
    for i in range(start, end):
        # print(st.row_values(i))  # st.row_values(i)：get row value/st.col_values(i)：get column
value
        values = []
```

```
        row = st.row_values(i)
        for j in range(0, len(row)):
            values.append(row[j])
        table.append(values)
    table_ = np.array(table)
    table_ = Variable(torch.from_numpy(table_).type(torch.FloatTensor))
    return table_

#chrom classes
class Chrom:
    chrom = []
    fitness = 0  #0:ROH 1:S
    #fitness2 = 0
    def showChrom(self):
        print(self.chrom)
    def showFitness(self):
        print(self.fitness)

#setup
N = 325  #number of variables in the group
mut = 0.5  #rate of mutation

pop = {}  #dictionary to store chrom
for i in range(N):
    pop['chrom'+str(i)] = Chrom()
chromNodes = 15  #number of variables
iterNum = 275  #iteration number

#chromRange = [[1, 12], [600, 1000], [0, 5], [150, 250], [40, 80], [45, 55], [0.5, 200], [0.25, 0.4],
[15, 250], [3, 25], [40, 80], [600, 800], [-0.5, 25], [0.5,2.0],[0, 420]]  #range of chrom
chromRange = [[0, 5], [0.25, 0.4], [0.5, 2.0], [600, 1000], [15, 250], [150, 250], [600, 800], [3,
35], [0.5, 200], [40, 80], [-0.5, 25], [0, 420], [45, 55], [40, 80], [40, 80], [4.5, 6.0], [0, 0.15], [4.5,
5.85], [300, 400], [0, 250]]
aveFitnessList = []  #average adaptability
bestFitnessList = []  #optimal adaptability

#initial chrom
xltraindir = 'AOH.xlsx'
traindir = '20sample.xlsx'
```

```
AOH = xlsxread(xltraindir, 0, 325)

data = xlrd.open_workbook(traindir)#excel location
st = data.sheets()[0]  # read the first sheet in excel
for i in range(0, 325):
    values = []
    row = st.row_values(i)
    for j in range(0, len(row)):
        values.append(row[j])
    pop['chrom'+str(i)].chrom = values.copy()

net = SELayer(20,40,2).to(device)
net.load('./models/net_039.pth')

pop = Fitness.calFitness(pop, net, device)  #calculate adaptibility
bestChrom = Genetic.findBest(pop, AOH)  #find optimal chrom
bestFitnessList.append(bestChrom[1])
aveFitnessList.append(Genetic.calAveFitness(pop, N))  #calculate and store average adaptability

#start iteration
x= [0]
for t in range(0, iterNum):
    #chrom mutate
    pop = Genetic.mutChrom(pop, mut, chromNodes, bestChrom, chromRange)
    #find optimal
    nowBestChrom = Genetic.findBest(pop, AOH)
    #find and compare the average before and after mutation
    bestChrom = Genetic.compareChrom(nowBestChrom, bestChrom)#find optimal chrom
    #store best and average
    bestFitnessList.append(bestChrom[1])
    bianliang = bestChrom[1]#predicted value
    aveFitnessList.append(Genetic.calAveFitness(pop, N))
    x.append(t+1)
    print(bianliang)

plt.figure(1)
plt.plot(x, bestFitnessList)
plt.show()
```