Research Octane Number Prediction Based on Feature Selection and Multi-Model Fusion

Junlin Gu
Jiangsu Vocational College of Electronics and Information, China

***Abstract*—The catalytic cracking-based process for lightening heavy oil yields gasoline products with sulfur and olefin contents surpassing 95%, consequently diminishing the Research Octane Number (RON) of gasoline during desulfurization and olefin reduction stages. Hence, investigating methodologies to mitigate RON loss in gasoline while maintaining effective desulfurization is imperative. This study addresses this challenge by initially performing data cleaning and augmentation, employing box plot modeling and Grubbs' test for outlier detection and removal. Subsequently, through the integration of mutual information and the Lasso method, data dimensionality is reduced, with the top 30 variables selected as primary factors. A predictive model for RON loss is then established based on these 30 variables, utilizing random forest and Support Vector Regression (SVR) models. Employing this model enables the computation of RON loss for each data sample. Comparing with existing methods, our approach could ensure a balance between effective desulfurization and mitigated RON loss in gasoline products.**

***Keywords*—Feature selection; random forest model; support vector machine model; RON loss**

I. Introduction

Gasoline stands as a cornerstone of automotive fuels, yet its combustion releases harmful substances into the atmosphere, notably sulfur and olefin components. Given that gasoline production predominantly hinges on heavy oil as a feedstock, characterized by high impurity levels, the quest for cleaner gasoline has emerged as a central concern within the industrial sphere.

The Research Octane Number (RON) serves as a crucial indicator of gasoline's ability to withstand compression ratios. In scenarios where gasoline attains high quality, devoid of impurities and undesired chemical substances, the RON stands as the most scientifically robust, precise, and widely embraced benchmark for evaluating gasoline's actual performance. However, the presence of desulfurization and olefin reduction technologies often leads to a decrease in gasoline's RON, directly impacting economic efficiency. Consequently, within the realm of catalytic cracking gasoline production, the focus has shifted towards reducing sulfur and olefin content while preserving RON.

Currently, numerous scholars are actively engaged in researching the accurate calculation of Research Octane Number (RON). Regression analysis methods are commonly employed for constructing RON prediction models owing to their simplicity and convenience [1]. However, in industrial settings, collected data may contain unnecessary redundancies, leading to collinearity issues among variables. To address this challenge, some researchers have proposed algorithmic enhancements aimed at eliminating data collinearity.

Kardamakis et al. [2] were among the first to utilize Linear Predictive Coding (LPC) to process noise and eliminate collinearity, subsequently employing the MLR algorithm to construct an RON prediction model based on near-infrared spectroscopy. Similarly, Benavides [3] introduced regularization to constrain the objective function of MLR, effectively resolving collinearity issues. They further combined ridge regression with near-infrared spectroscopy to develop an RON prediction model. Moreover, recognizing the limitations of traditional single models in addressing diverse and complex operating conditions, Xie et al. [4] proposed a research-based RON prediction model utilizing the random forest regression algorithm. Wang et al. [5], by optimizing the desulfurization process, established an RON loss model using residual analysis and the least squares method, analyzing the impact of reducing operational steps on decreasing RON loss during desulfurization. Furthermore, Liu et al. [6] incorporated gasoline RON as one of the modeling variables and constructed a prediction model based on the principles of random forest classification, with gasoline RON as the dependent variable, to predict RON loss during the desulfurization process.

Despite the significant progress made by numerous scholars in RON prediction research, meeting increasingly stringent fuel standards and the growing demand for accurate RON predictions remains challenging.

Given the complex and variable nature of operating conditions, this paper focuses on the RON and sulfur content of the products. To establish a predictive model for RON loss, we utilized a large volume of historical data accumulated over nearly four years from a petrochemical company's refining and desulfurization unit, and employed data mining techniques [7] to construct an optimization model. The main contributions of this paper are as follows:

1) We employed box plot modeling and Grubbs' test to pinpoint data samples and eliminate outliers. Subsequently, in conjunction with mutual information and the Lasso method [8], we performed dimensionality reduction on the data variables to select the key variables.
2) We established a prediction model for Research Octane Number (RON) loss using the random forest and support vector regression (SVR) models [9]. This model facilitated the computation of RON loss for each data sample.
3) We employed the conjugate gradient method [10] to

establish an optimization model for the key variables, ensuring that the sulfur content of the product does not exceed 5 μg/g while RON loss remains below 30%. We used the random forest model [11] to optimize the key variables in the data samples, progressively reducing RON loss by iteratively adjusting operational variables.

In the forthcoming sections of this paper, we delineate a structured approach. Section II furnishes the preliminary knowledge essential to our scheme, while Section III describes the details of our proposed solution. Section IV describes our experimental results and analysis. Finally, Section V offers a conclusion of our study.

II. Preliminary Knowledge

The scheme proposed in this paper mainly involves techniques such as data preprocessing, feature selection, and feature extraction. This section provides an introduction to these pertinent technologies.

A. Data Pre-processing

1) Box plot model: The samples of data that fall outside the operational range were detected and removed using the box plot method [12]. Box plot not only provides a visual representation of identifying outliers in the dataset, but also helps determine the dispersion and skewness of the data. It consists of five values: the minimum value (min), the lower quartile (Q1), the median, the upper quartile (Q3), and the maximum value (max). The lower quartile, median, and upper quartile together form a "box with whiskers" structure. A line extends from the upper quartile to the maximum value, and this line is referred to as the "whisker".

The whiskers in the box plot are used to identify and remove outliers from the skewed population. In this context, the maximum and minimum values are set as 1.5 times the interquartile range (IQR), which is the range between the upper and lower quartiles. Specifically, the whiskers extend up to a distance of 1.5 times the IQR from the upper and lower quartiles. The formula of IQR is as follows:

$$IQR = Q3 - Q1. \tag{1}$$

The IQR also represents the length of the box plot. Therefore, the minimum value (min) and maximum value (max) can be determined as follows:

$$min = Q1 - 1.5 \times IQR. \tag{2}$$

$$max = Q3 - 1.5 \times IQR. \tag{3}$$

When applying the box plot analysis to data, if there are outliers that fall below the minimum observed value, the lower whisker is set at the minimum observed value, and the outliers are individually marked as points. If there are no values lower than the minimum observed value, the lower whisker extends to the minimum value. Similarly, if there are outliers that exceed the maximum observed value, the upper whisker is set at the maximum observed value, and the outliers are individually marked as points. If there are no values greater than the maximum observed value, the upper whisker extends to the maximum value.

2) Grubbs' criterion model: Based on Grubbs' criterion (3σ criterion) [13] for removing outliers from a sample, we first assume that the measured variable is measured with equal precision, resulting in x_1, x_2, ..., x_n. Then, we calculate the arithmetic mean x and the residual errors $v_i = x_i - x$ (for i=1,2,...,n). Based on these variables, we use the Beale's formula to calculate the standard error σ. If there exists a measurement value x_b with a residual error v_b (1¡=b¡=n) satisfying —v_b—=—$x_b - x$—¿3σ, it is considered an outlier with a gross error and should be removed. The Beale's formula is given as follows:

$$\sigma = [\frac{1}{n-1}\sum_{i=1}^{n} {v_i}^2]^{1/2} = \{[\sum_{i=1}^{n} {x_i}^2 - (\sum_{i=1}^{n} x_i)^2/n]/(n-1)\}^{1/2} \tag{4}$$

B. Feature Selection

1) Mutual information model: Mutual information [14] is a useful information measure in information theory that quantifies the amount of information contributed by the occurrence of one event to the occurrence of another event. It can be viewed as the amount of information about one random variable contained in another random variable or as the reduction in uncertainty of one random variable due to the knowledge of another random variable. Mutual information graph is shown in Fig. 1.

Let the joint distribution of two random variables (X,Y) be denoted as p(x,y), and their marginal distributions be denoted as p(x) and p(y), respectively. The mutual information I(X,Y) is the relative entropy between the joint distribution p(x,y) and the marginal distributions p(x), p(y). According to the definition of entropy, the derivation formulas are as follows.

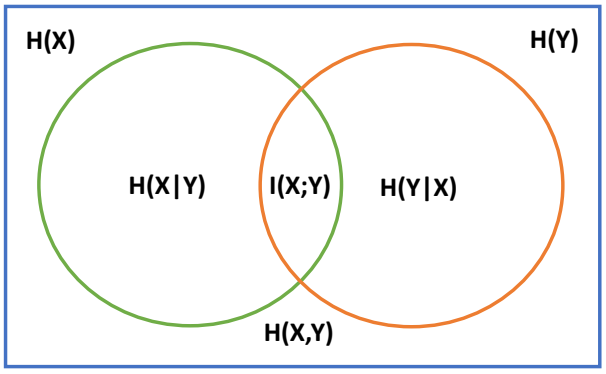


Fig. 1. Mutual information graph.

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y), \tag{5}$$

$$H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{6}$$

Therefore, the final calculation formula is as follow:

$$I(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) log \frac{p(x,y)}{p(x)p(y)}. \tag{7}$$

2) Lasso regression model: The Lasso method is a compression estimation technique based on the idea of shrinking the variable set. By constructing a penalty function, it compresses the coefficients of the variables and forces some regression coefficients to become zero, thereby achieving variable selection.

Regularization [15] is a method to prevent overfitting, which usually occurs when there are too many variables or features. In such cases, the resulting equation can fit the training data very well, with a loss function that may be very close to or equal to zero.

However, such a curve may fail to generalize to new data samples. In regularization, all feature variables are retained, but the magnitude of the feature variables is reduced. When there are many feature variables, each variable can have some impact on the prediction. Lasso regression adds L1 regularization to the loss function. The coefficients trained by Lasso regression are sparse and can be used for feature selection. Because the absolute value function is not differentiable at zero, directly applying gradient descent is not feasible. Therefore, alternative algorithms such as coordinate descent are used. Coordinate descent method [16] updates one attribute at a time, and the loss function is given as follows:

$$L(w) = f(w) + \lambda \|w\|_1^1 = \left\|y - X^T w\right\|_2^2 + \lambda \|w\|_1^1. \tag{8}$$

C. Regression Model

1) Random forest model: In machine learning [17], a random forest is a classifier that consists of multiple decision trees, and its output class is determined by the majority vote of individual tree outputs. Random forests have several advantages which can produce highly accurate classifiers for various types of data, handle a large number of input variables, and evaluate the importance of variables when determining class labels. During the construction of the forest, they can generate unbiased estimates of generalized errors, and they can balance errors for imbalanced classification datasets. The specific algorithm is as follows:

1) Let N represent the number of training examples and M represent the number of features.
2) Input the number m of features to determine the decision result at a node in the decision tree, where m should be much smaller than M.
3) Randomly sample N times with replacement from the N training examples (samples) to form a training set, and use the unsampled examples (samples) for prediction to evaluate their errors.
4) For each node, randomly select m features, and the decision at each node in the decision tree is based on these features. Based on these m features, compute the optimal splitting method.
5) Each tree grows fully without pruning, which may be adopted after building a complete tree-based classifier.

When tuning the parameters of a random forest using sklearn [18], it is significant to perform parameter tuning based on the relationship between generalization error and model complexity. By assessing the impact of parameters on the model, they can be sorted in descending order of influence, determining which parameters reduce model complexity and which ones increase it. Suitable parameters are then selected sequentially, and parameter tuning is carried out through methods such as plotting learning curves or performing grid searches, until the maximum accuracy score is achieved.

The prediction error rate of a random forest depends on two factors: the correlation between any two trees in the forest and the classification ability of each individual tree. Higher correlation leads to a higher error rate. The stronger the classification ability of an individual tree, the stronger the overall classification ability of the entire forest. If, within a tree, samples split based on a certain feature m are more likely or less likely to split on feature k, there exists a certain degree of interaction between m and k.

The key issue in building a random forest is how to select the optimal value of m. To address this problem, the calculation of the out-of-bag error (oob error) [19] is crucial. One important advantage of random forests is that there is no need for cross-validation or an independent test set to obtain an unbiased estimate of the error. It can be internally evaluated, meaning that an unbiased estimate of the error can be established during the generation process. When constructing each tree, we utilize different bootstrap samples (randomly and with replacement) from the training set. Consequently, for each tree (let's assume the k-th tree), approximately one-third of the training instances are not involved in the generation of the k-th tree. These instances are referred to as the oob samples for the k-th tree.

Such sampling characteristics allow us to perform the oob estimation, and its calculation method is as follows:

1) For each sample, compute its classification by the trees for which it serves as an oob sample (approximately one-third of the trees).
2) Use a simple majority vote as the classification result for that sample.
3) Finally, calculate the oob error rate of the random forest as the ratio of misclassified samples to the total number of samples.

2) Support vector regression model: Support Vector Machine (SVM) [20] is a classification algorithm that can also be used for regression, offering different models based on the input data. By seeking to minimize structured risk, SVM enhances the generalization ability of the learning machine, achieving the minimization of empirical risk and confidence interval. This allows obtaining good statistical patterns even with limited statistical samples. In simple terms, SVM is a

binary classification model, with the basic model defined as the linear classifier in feature space with the maximum margin, known as the maximal margin classifier. The learning strategy of SVM is to maximize the margin, ultimately transforming into solving a convex quadratic programming problem.

In Support Vector Regression (SVR) [21], the objective is to find a regression plane that minimizes the distance between the plane and a set of data points. SVR is an important application branch of Support Vector Machines (SVM). In traditional regression methods, a prediction is considered correct only if the regression function $f(x)$ is exactly equal to y. However, in support vector regression, a prediction is considered correct as long as the deviation between $f(x)$ and y is not too large. In other words, if the absolute difference between the predicted value $y(x)$ and the target value t is smaller than ϵ, the error given by the error function is zero, where ϵ ¿ 0.

The regularization error function is as follows.

$$C\sum_{n}[E_{\varepsilon}(y_n - t_n)] + \frac{1}{2}\|w\|^2, y_n - \varepsilon \le t_n \le y_n + \varepsilon \quad (9)$$

The error function after introducing slack variables is as follows.

$$C\sum_{n}\{\widetilde{\xi}_n + \xi_n\} + \frac{1}{2}\|w\|^2 \quad (10)$$

The discriminant function is as follows.

$$y(x) = \sum_{n}(a_n - \widetilde{a}_n)\, k(x, x_n) + b \quad (11)$$

3) The correlation coefficient model: Correlation [22] is a non-deterministic relationship, and the correlation coefficient is a measure of the linear relationship between variables. Due to variations in the subjects under study, there are several different ways to define the correlation coefficient.

The simple correlation coefficient, also known as the correlation coefficient or linear correlation coefficient, is typically represented by a letter and is used to measure the linear relationship between two variables. The definition formula is as follows.

$$r(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)\,Var(Y)}} \quad (12)$$

In which, $Cov(X,Y)$ represents the covariance between X and Y, $Var(X)$ represents the variance of X, $Var(Y)$ represents the variance of Y.

D. Optimization Algorithm

The system of linear equations is known to be representable as $Ax = b$. When A is a real symmetric matrix, that is, the expression for the derivative of the quadratic form $f(x) = \frac{1}{2}x^T Ax - b^T x + c$ with respect to x when it equals zero, is as follows.

$$f'(x) = \frac{1}{x}A^T x + \frac{1}{2}Ax - b \quad (13)$$

When A is a real symmetric matrix in the formula, $f'(x) = 0$ is equivalent to $Ax = b$, and thus, solving the system of linear equations can be transformed into solving $minf(x)$.

From the knowledge of algebra, it is known that when the matrix A is positive definite, positive semi-definite, negative definite, or indefinite, the equation set $Ax = b$ has different solutions, corresponding to different minimum values of $f'(x) = 0$. The impact of different situations of matrix A on equation $f(x)$ is shown in Fig. 2.

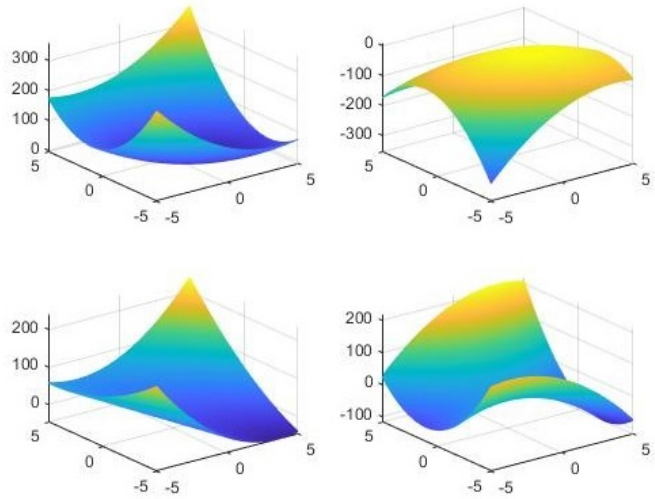


Fig. 2. Equation solution plot.

The distribution of solution patterns in Fig. 2 corresponds to A being a positive definite, negative definite, positive semi-definite, and indefinite matrix, respectively. From the figure, it can be observed that when A is an indefinite matrix, it is not possible to find the minimum value of $f(x)$ by setting its derivative to zero.

The conjugate gradient method [23] can solve the above problem. First, it is assumed that A has good properties, namely, symmetry and positive definiteness. When seeking the minimum value of the function $f(x)$, its derivative leads to a sequence of solution vectors: $x_{(1)}$, $x_{(2)}$, ..., from which we obtain $f'(x_{(i)}) = Ax_{(i)} - b$.

From calculus knowledge, we know that to consider $f(x_{(i+1)})$ as a function of $a_{(i)}$, and to find the most appropriate step length, we need to set it as follows:

$$\frac{d}{d\alpha_{(i)}}f(x_{(i+1)}) = 0 \quad (14)$$

By applying the chain rule [24], we find that $r_{(i)}$ is orthogonal to $r_{(i+1)}$, meaning that the step length for each step can be determined based on the current residual $r_{(i)}$.

Based on the iterative process of the steepest descent method [25], we can obtain $r_{(i+1)}^T r_{(i)} = 0$. However, the steepest descent method has a significant issue: in order to converge to the vicinity of the solution, the same iteration direction may be followed more than once. To address this problem, if we can select a series of linearly independent direction vectors $d_{(0)}$, $d_{(1)}$, $d_{(2)}$, ..., $d_{(n-1)}$, and move along each direction only once, we can eventually reach the solution x without

encountering the issue of repeating the same direction. The most straightforward idea comes from the Cartesian coordinate system. If each direction is orthogonal, there will naturally be no problem of repeating the same direction. This leads to the condition $\alpha_{(i)} = -\frac{d_{(i)}^T e_{(i)}}{d_{(i)}^T d_{(i)}}$. Assuming that the selected series of direction vectors are all pairwise orthogonal with respect to matrix A, the formula of $\alpha_{(i)}$ is as follows.

$$\alpha_{(i)} = \frac{d_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}} \tag{15}$$

According to this formula, for an n-order system of equations, it will take at most n steps to converge to the correct solution.

From the above formulas, it is evident that the residuals between each iteration are mutually orthogonal. Therefore, we can define the residual $r_{(0)}, r_{(1)}, r_{(2)}, ..., r_{(n-1)}$ as the basis before conjugation. Since using conjugate directions for iteration requires at most n steps, and each step eliminates the error in that direction, this set of bases is not only linearly independent but also possesses the desirable property of orthogonality.

III. The Proposed Scheme

This article aims to construct a predictive model for octane loss. To achieve this, we first filter the data features, then build a predictive model to calculate potential octane loss. Furthermore, we employ optimization algorithms to adjust variables in order to reduce octane loss.

A. Data Filtration

Industrial data often contain a significant amount of invalid and outlier data. For data with a high degree of missing values that cannot be filled, we delete sample data where all values are missing and use the average of data from the two hours before and after to fill in missing values. For samples that fall outside the original data variable operation range or contain outliers, we establish mathematical models for resolution. The entire data processing workflow is illustrated in Fig. 3.

B. Feature Selection

High-dimensional feature variables often increase the complexity of engineering problem analysis. In practical engineering applications, it's common to employ dimensionality reduction techniques before modeling. This approach can improve prediction accuracy, enable the construction of more efficient predictive models, and enhance the understanding and interpretability of the models. It helps in ignoring minor factors and identifying and analyzing the key variables and factors influencing the model.

To achieve this, we use mutual information entropy, correlation coefficients, and Lasso regression to select important features, making it easier to establish subsequent predictive models. As shown in Fig. 4, we adopt two different approaches for feature selection in this article. We use two combinations of methods to filter the main variables affecting octane loss: one approach uses mutual information and correlation coefficients, while the other employs Lasso regression.

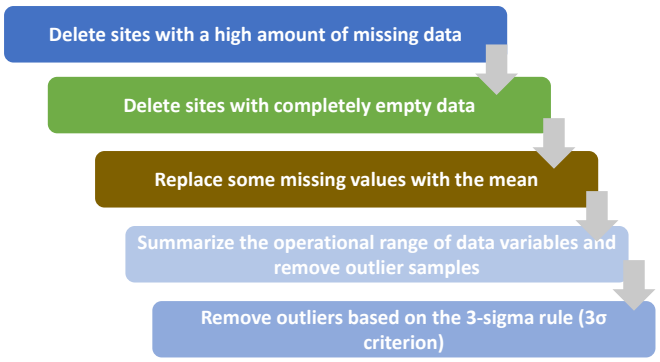


Fig. 3. Data processing workflow. There are five steps, including deleting missing data, deleting empty data, replacing missing data, summarizing data distribution, and removing outlier data.

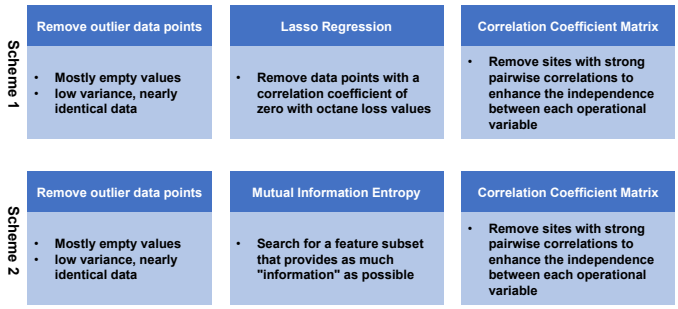


Fig. 4. Scheme model diagram. The lasso regression and mutual information entropy are used to filter features.

C. Development of RON Loss Prediction Model

In this section, we employ machine learning-based [17] models for regression prediction of RON as illustrated in the framework diagram in Fig. 5. Initially, the original data is subjected to outlier removal and standardization using box plots. After standardization, 80 percent of the samples are used for training, while the remaining 20 percent are reserved for testing. We establish RON loss prediction model using Random Forest prediction models and Support Vector Machine (SVM) techniques, followed by model validation.

IV. Experiments

A. Experiment settings

Dataset: We used a dataset comprising 325 data points obtained from actual production in a petrochemical enterprise. The dataset includes seven raw material properties, two properties of the adsorbents used in the initial adsorption stage, two properties of the adsorbents used in the regeneration stage, two product properties, and an additional 354 operating variables, totaling 367 variables.

Experimental Parameters: In our experiments, we set the standard deviation threshold to 0.3 in the Lasso regression process. For the random forest, we used 100 decision trees,

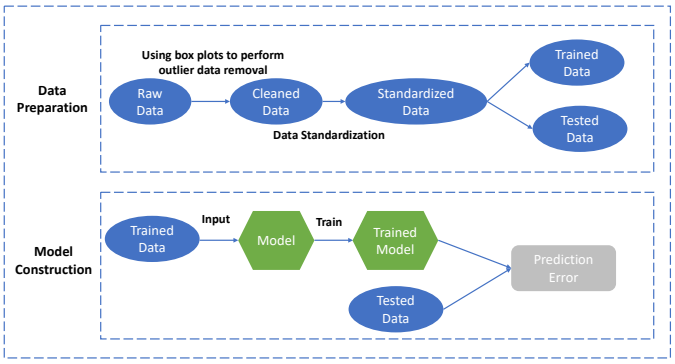


Fig. 5. Algorithm framework diagram.The data is first splited into training dataset and testing dataset. Then, we train models with the training dataset and the evaluate it on the testing dataset.

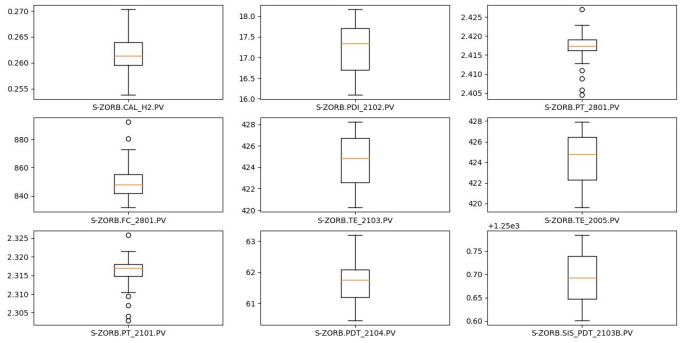


Fig. 6. Box Plot Method for Removing Sample Data Graph.

the Mean Absolute Error (MAE) as the error function, and a minimum sample size of 4 for leaf nodes. The support vector regression model had a penalty coefficient of 0.1 and a gamma value of 0.01.

B. Data Filtering Results

Based on the box plot model, a check was conducted on data samples. Due to the large number of data points, it is not feasible to display all of them. Fig. 6 below shows only a portion of the data points in sample that need to be removed, as indicated in the graph. It is necessary to delete the data points in sample that fall outside the numerical range defined by the upper and lower ends of the box plot. Further examination using the Grubbs' test revealed that there were no outliers requiring removal in the samples.

C. Primary Variable Selection

We employed two approaches for selection and then compared their effectiveness. First, we utilized the mutual information model to filter out 50 primary features.

Furthermore, we conducted additional filtering using the correlation coefficient model to identify 30 primary features.

Main feature variables can also be selected using Lasso regression, which involves the following steps:

TABLE I. MAIN VARIABLES SELECTED BY LASSO REGRESSION

S-ZORB.FT_5104.PV	S-ZORB.FT_9102.PV
S-ZORB.FT_5201.TOTAL	S-ZORB.FT_5101.TOTAL
S-ZORB.FT_9201.TOTAL	S-ZORB.FT_9202.TOTAL
S-ZORB.FT_9402.TOTAL	S-ZORB.FT_9403.TOTAL
S-ZORB.FT_5102.TOTAL	S-ZORB.FC_1202.TOTAL
S-ZORB.FT_1001.TOTAL	S-ZORB.PDT_2503.DACA
S-ZORB.TC_2201.PV	S-ZORB.FC_5103.DACA
S-ZORB.FT_1006.DACA.PV	S-ZORB.CAL.LEVEL.PV
S-ZORB.FT_1503.TOTALIZERA.PV	S-ZORB.FT_1504.TOTALIZERA.PV
S-ZORB.PT_7510.DACA	S-ZORB.TE_3111.DACA
S-ZORB.FT_1004.TOTAL	S-ZORB.FC_5203.DACA
S-ZORB.FT_1003.TOTAL	S-ZORB.TE_2001.DACA
S-ZORB.FT_9401.TOTAL	S-ZORB.FT_1503.DACA.PV
S-ZORB.FC_1101.TOTAL	S-ZORB.FT_5204.TOTALIZERA.PV
S-ZORB.FT_9102.TOTAL	S-ZORB.FT_1001.TOTAL

1) Calculate the standard deviation [26] for each of the 325 samples' variables. Variables with a standard deviation less than the threshold will be removed. The calculation formula is as follows.

$$\delta = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n}} \tag{16}$$

When δ¡ 0.3, the variable will be removed.

2) Count the number of zero elements in each variable. If the number of zeros exceeds 30% of the total elements in that column, the variable will be removed. If the number of zeros does not exceed 30% of the total elements in that column, the zero values in the variable will be replaced with the mean of its non-zero values.

3) Perform Lasso regression on the remaining variables to select 30 main variables, as shown in Table I.

For the two aforementioned approaches, we constructed the same model and then separately used the features selected by these two approaches for training and testing to assess the quality of the feature sets.

Specifically, we employed a support vector regression model with identical parameter settings as the base model to evaluate the quality of the feature sets based on its detection performance. The experimental results are presented in Table *.

In terms of specific metrics, we used the Mean Squared Error (MSE) between predicted values and actual values as the performance indicator. The features selected by the Lasso regression model ultimately resulted in an MSE of 0.0249, whereas the features selected using mutual information entropy yielded an MSE of 0.0258, slightly higher than that of the Lasso regression. Therefore, we opted for the Lasso regression model as the feature selection approach.

D. RON Loss Prediction Performance

Based on the primary operating variable features, we utilized random forest and SVR (Support Vector Regression) for prediction separately.

Fig. 7. Prediction performance graph of random forest regression model. The value of y-axis means the feature values.

TABLE II. THE REGRESSION PERFORMANCE UNDER DIFFERNT SVR KERNELS

Kernel	Regression Performance		
	R2	MAE	RMSE
Linear	**0.9666**	**0.0757**	**0.1533**
Polynomial	0.8258	0.2169	0.2657
Gauss	0.8803	0.1962	0.2396
Laplace	0.7364	0.3305	0.3129
Sigmoid	-12.6431	2.0192	3.2221

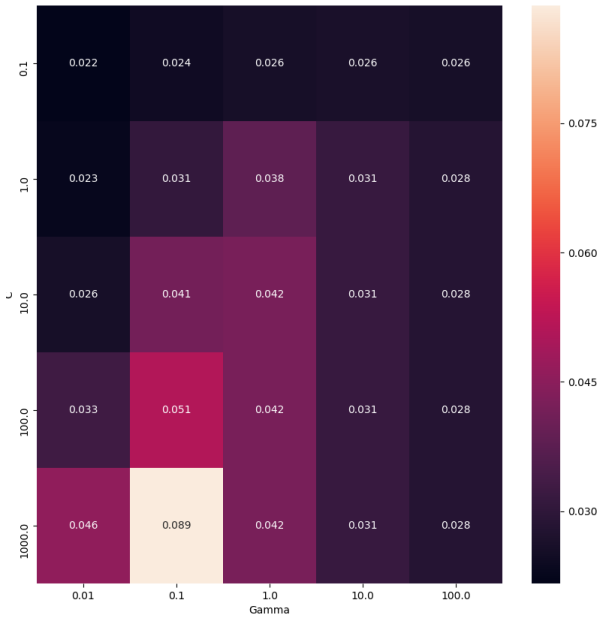
1) Random Forest Prediction Performance: Based on the selected primary operational variable features, we used a random forest for regression prediction. The random forest model involves multiple model parameters. To choose the model that best suits the current data, we conducted a grid search for parameter tuning.

From the search results, it can be observed that when the number of decision trees in the random forest is set to 100, the used error metric is MAE (Mean Absolute Error) [27], and the minimum samples per leaf node is 4, the model achieves its minimum prediction error of 0.233917. After concluding the parameter search, we constructed a new random forest model using the optimal parameters. The final model's predictive performance is illustrated in Fig. 7.

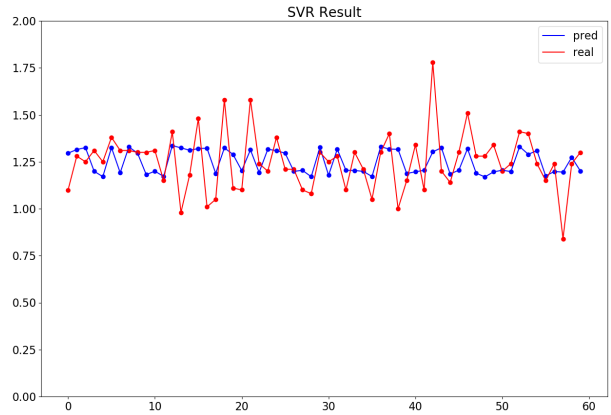
2) Model prediction performance: We also employed a support vector regression model for prediction. The support vector regression model involves multiple model parameters such as penalty coefficient [32] and Gamma value [33]. To select the model that best suits the current data, we conducted a grid search for the penalty coefficient and Gamma value.

TABLE III. COMPARISON WITH OTHER MODELS

Method	Regression Performance		
	R2	MAE	RMSE
Linear regression [28]	0.4174	6.4373	43.3017
Decision Tree [29]	0.9483	0.0863	0.1720
Simple DNN [30]	0.6989	0.5980	1.3927
RandomForest [31]	0.9724	0.0526	0.1077
SVR [21]	0.9666	0.0757	0.1533
RandomForest+SVR	**0.9868**	**0.0453**	**0.0973**



(a) Parameter Search Results



(b) Model Prediction Performance

Fig. 8. The parameter search results and model prediction performance of support vector regression model.

The prediction errors obtained for different parameter configurations are shown in Fig. 8(a). From the search results, it can be observed that when the penalty coefficient for the support vector regression model is set to 0.1 and the Gamma value is set to 0.01, the model achieves its minimum prediction error of 0.022. After completing the parameter search, we constructed a new support vector regression model using the optimal parameters. The final model's predictive performance is illustrated in Fig. 8(b).

We also evaluated the fitting performance of Support Vector Regression (SVR) under different kernel functions. It is can be seen from Table II that our approach achieves optimal results when employing the linear kernel function. Under such settings, we compared the ensemble model and other models. As shown in Table III, our approach shows better regression

performance. The combination of RandomForest and SVR could increase the accuracy of feature regression.

V. Conclusion

Gasoline octane loss optimization has become a focal point of concern in the industry. In this paper, addressing the issue of RON loss optimization, we employed the lasso regression and correlation coefficient methods to feature selection, reducing the information redundancy that affects the octane loss model. We utilized random forest and support vector machine models to establish RON loss prediction models, training and testing them with well-preprocessed data to predict RON loss values. Combining Random Forest and SVR, our proposed solution achieves an R2 value of 0.9868, surpassing the performance of multiple existing models. In future work, we will further refine feature selection algorithms and explore the utilization of genetic algorithms to determine optimal parameters for the model.

Acknowledgment

This work is supported in part by the Jiangsu Province Department of Industry and Information Technology Key Technology Innovation Project Orientation Program under grant numbers 141-62-65, in part by the Jiangsu Provincial Science and Technology Department Digital Public Service Platform Project under grant numbers 93208000931, in part by the Jiangsu Provincial Department of Science and Technology Industry-Academia-Research Project under grant numbers BY20221343, in part by Jiangsu Provincial Vocational Education Big Data Technology 'Double-Teacher' Master Studio Project, in part by the Jiangsu Province large-scale scientific instrument open sharing project under grant numbers TC2023A073.

References

[1] D. Akal, S. Öztuna, and M. K. Büyükakın, "A review of hydrogen usage in internal combustion engines (gasoline-lpg-diesel) from combustion performance aspect," *International journal of hydrogen energy*, vol. 45, no. 60, pp. 35 257–35 268, 2020.

[2] A. A. Kardamakis and N. Pasadakis, "Autoregressive modeling of near-ir spectra and mlr to predict ron values of gasolines," *Fuel*, vol. 89, no. 1, pp. 158–161, 2010.

[3] A. Benavides, C. Zapata, P. Benjumea, C. A. Franco, F. B. Cortés, and M. A. Ruiz, "Predicting octane number of petroleum-derived gasoline fuels from mir spectra, gc-ms, and routine test data," *Processes*, vol. 11, no. 5, p. 1437, 2023.

[4] Y. Xie, K. Ji, M. Chen, and J. Zhang, "Predictive modeling of gasoline octane loss based on xgboost algorithm and multiple linear regression analysis," in *Second International Conference on Digital Signal and Computer Communications (DSCC 2022)*, vol. 12306. SPIE, 2022, pp. 416–420.

[5] H. Wang, X. Chu, P. Chen, J. Li, D. Liu, and Y. Xu, "Partial least squares regression residual extreme learning machine (plsrr-elm) calibration algorithm applied in fast determination of gasoline octane number with near-infrared spectroscopy," *Fuel*, vol. 309, p. 122224, 2022.

[6] C. Liu, N. Deng, J. T. Wang, and H. Wang, "Predicting solar flares using sdo/hmi vector magnetic data products and the random forest algorithm," *The Astrophysical Journal*, vol. 843, no. 2, p. 104, 2017.

[7] S.-H. Liao, P.-H. Chu, and P.-Y. Hsiao, "Data mining techniques and applications–a decade review from 2000 to 2011," *Expert systems with applications*, vol. 39, no. 12, pp. 11 303–11 311, 2012.

[8] J. Ranstam and J. Cook, "Lasso regression," *Journal of British Surgery*, vol. 105, no. 10, pp. 1348–1348, 2018.

[9] F. Zhang and L. J. O'Donnell, "Support vector regression," in *Machine learning*. Elsevier, 2020, pp. 123–140.

[10] Z. Ahmed and S. Mahmood, "New formula for conjugate gradient method to unconstrained optimization," *Mustansiriyah Journal of Pure and Applied Sciences*, vol. 1, no. 2, pp. 21–27, 2023.

[11] G. Biau, "Analysis of a random forests model," *The Journal of Machine Learning Research*, vol. 13, pp. 1063–1095, 2012.

[12] M. Walker, Y. Dovoedo, S. Chakraborti, and C. Hilton, "An improved boxplot for univariate data," *The American Statistician*, vol. 72, no. 4, pp. 348–353, 2018.

[13] K. Ding, J. Zhang, H. Ding, Y. Liu, F. Chen, and Y. Li, "Fault detection of photovoltaic array based on grubbs criterion and local outlier factor," *IET Renewable Power Generation*, vol. 14, no. 4, pp. 551–559, 2020.

[14] H. Shakibian and N. Moghadam Charkari, "Mutual information model for link prediction in heterogeneous complex networks," *Scientific reports*, vol. 7, no. 1, p. 44981, 2017.

[15] G. Mustafa, A. Ghaffar, and M. Aslam, "A subdivision-regularization framework for preventing over fitting of data by a model," *Applications and Applied Mathematics: An International Journal (AAM)*, vol. 8, no. 1, p. 11, 2013.

[16] S. J. Wright, "Coordinate descent algorithms," *Mathematical programming*, vol. 151, no. 1, pp. 3–34, 2015.

[17] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[18] B. Komer, J. Bergstra, and C. Eliasmith, "Hyperopt-sklearn," *Automated Machine Learning: Methods, Systems, Challenges*, pp. 97–111, 2019.

[19] S. Janitza and R. Hornung, "On the overestimation of random forest's out-of-bag error," *PloS one*, vol. 13, no. 8, p. e0201904, 2018.

[20] D. A. Pisner and D. M. Schnyer, "Support vector machine," in *Machine learning*. Elsevier, 2020, pp. 101–121.

[21] M. Awad, R. Khanna, M. Awad, and R. Khanna, "Support vector regression," *Efficient learning machines: Theories, concepts, and applications for engineers and system designers*, pp. 67–80, 2015.

[22] B. Ratner, "The correlation coefficient: Its values range between+ 1/-1, or do they?" *Journal of targeting, measurement and analysis for marketing*, vol. 17, no. 2, pp. 139–142, 2009.

[23] J. L. Nazareth, "Conjugate gradient method," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 1, no. 3, pp. 348–353, 2009.

[24] V. E. Tarasov, "On chain rule for fractional derivatives," *Communications in Nonlinear Science and Numerical Simulation*, vol. 30, no. 1-3, pp. 1–4, 2016.

[25] J. C. Meza, "Steepest descent," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 6, pp. 719–722, 2010.

[26] D. K. Lee, J. In, and S. Lee, "Standard deviation and standard error of the mean," *Korean journal of anesthesiology*, vol. 68, no. 3, pp. 220–223, 2015.

[27] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)," *Geoscientific model development discussions*, vol. 7, no. 1, pp. 1525–1534, 2014.

[28] G. James, D. Witten, T. Hastie, R. Tibshirani, and J. Taylor, "Linear regression," in *An introduction to statistical learning: With applications in python*. Springer, 2023, pp. 69–134.

[29] Y.-Y. Song and L. Ying, "Decision tree methods: applications for classification and prediction," *Shanghai archives of psychiatry*, vol. 27, no. 2, p. 130, 2015.

[30] Q. YANa, X. LINb, Z. QINb, G. LUOc, D. Wang, and X. Xiao, "A deep learning framework in fcc process control," pp. 709–716, 2021.

[31] S. J. Rigatti, "Random forest," *Journal of Insurance Medicine*, vol. 47, no. 1, pp. 31–39, 2017.

[32] G. Liberopoulos, I. Tsikis, and S. Delikouras, "Backorder penalty cost coefficient "b": What could it be?" *International Journal of Production Economics*, vol. 123, no. 1, pp. 166–178, 2010.

[33] K. B. Pulliam, J. Y. Huang, R. M. Howell, D. Followill, R. Bosca, J. O'Daniel, and S. F. Kry, "Comparison of 2d and 3d gamma analyses," *Medical physics*, vol. 41, no. 2, p. 021710, 2014.