

Research Article

Applying Machine Learning to Chemical Industry: A Self-Adaptive GA-BP Neural Network-Based Predictor of Gasoline Octane Number

Xingzhen Tao,¹ Yue Liu ,¹ Haiping Li,¹ Yufei Xie,¹ Lin Peng,¹ Chao Li ,² Lingling Guo,³ and Yinling Zhang³

¹School of Information Engineering, Jiangxi College of Applied Technology, China

²Zhijiang College, Zhejiang University of Technology, China

³College of Chemical Engineering, Zhejiang University of Technology, China

Correspondence should be addressed to Chao Li; cynthia0217@163.com

Received 18 February 2022; Revised 22 March 2022; Accepted 1 April 2022; Published 22 April 2022

Academic Editor: Yan Huang

Copyright © 2022 Xingzhen Tao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Octane number is a measure of gasoline's ability to resist detonation and combustion in the cylinder; the higher the value, the better the resistance to detonation. The accurate prediction of octane loss during gasoline refining could facilitate production management and ensure gasoline octane. The backpropagation neural network is a traditional method adopted for the octane loss prediction, but there exists the issues of low training accuracy and poor generalization in the traditional BP neural network model caused by randomly generated weights and thresholds at input. In this paper, we propose a novel approach to optimize the weights and thresholds for gasoline octane number prediction based on a self-adaptive genetic algorithm. The experimental result shows that the proposed model outperforms in accuracy and generalization in the competition with the traditional BP neural network. The coefficient of determination R_2 of the performance index in the experiment is improved from 0.81502 to 0.95628, and the average prediction error among 10 groups of experiments was reduced from 0.0061 to 0.0041.

1. Introduction

Octane number is one of the most important properties of gasoline [1, 2], which directly affects the antiknock performance, fuel consumption, and low-temperature start and acceleration performances of automobiles. For product oil industry, gasoline octane number is an important quality index in the process of purchasing, storage, transportation, and sales [3]. Currently, the ASTM-CFR standard is the most commonly used standard octane test method; however, this method is expensive, and the test dosage is large, time-consuming, and complicated to operate [4]. In order to make up for the defects and shortcomings of experimental methods, theoretical prediction studies of octane number of gasolines can be carried out to establish a reliable prediction model since octane number of gasolines is closely related to its chemical composition [5]. In recent years, with

the establishment of the Sinopec Sales Enterprise Laboratory Information Management System, the accumulation and sharing of quality data have been realized. Relying on the massive physical and chemical index data of gasoline in the database, it is possible to establish a gasoline octane number prediction model using machine learning algorithms [6]. Predictive models can be divided into two categories [7]: one is linear models for predicting octane number, such as multiple linear regression analysis and partial least squares; the other is nonlinear models for predicting octane number, such as artificial neural network algorithms and support vector machine regression. A backpropagation (BP) neural network, one of the most reliable and classical neural networks among artificial neural networks, can be chosen as the base model with convenient operation and powerful learning ability [8–10]. However, the traditional BP neural network training suffers from slow convergence and low prediction

accuracy. To address these problems, a genetic algorithm is used to optimize the parameter training of the model [11–14].

In this paper, a BP neural network gasoline octane number prediction model is proposed based on self-adaptive genetic algorithm optimization with gasoline physical and chemical indexes as its independent variables and octane number as its dependent variable. Comparative experiments are further conducted to validate this model. The contribution in this work can be summarized as follows: (1) an optimized BP neural network model is proposed to address the issues existing in the traditional BP neural network while dealing with the task of gasoline octane number prediction and (2) a comprehensive experiment and analysis to validate the proposed method are carried out. The rest of this paper is organized as follows: the related work is introduced in Section 2, the proposed BP-based algorithm is detailed in Section 3, the experiment is detailed in Section 4, and Section 5 concludes the paper.

2. Related Work

2.1. Gasoline Octane Number Prediction. Octane number is a measure of the gasoline's ability to resist detonation in the cylinder; the higher the value, the better the resistance to detonation. The quantitative analysis models of octane value are divided into linear and nonlinear models; linear models include multivariate statistical analysis methods [15], Raman spectral data combined with the partial least squares regression (PLS) algorithm [13], Raman spectrometer combined with the PLS algorithm [16], momentary combined with the local PLS algorithm (MC-PLS) [14], Fourier transform spectroscopy combined with PLS [17], linear predictive coding combined with MLR [18], NIR spectroscopy based on ANN, support vector machine (SVM) and multivariate statistical analysis [19], and the use of NIR spectroscopy. Most of the nonlinear models used algorithms such as multiple linear regression (MLR), Principal Component Regression (PCR), and ANN [20], among which ANN is the most effective; [21] used short-wave NIR spectra with laser-induced spectroscopy to develop an octane analysis model. Quantitative octane analysis requires a large sample set and a high level of model complexity. Linear and nonlinear models are slightly simpler to construct compared to linear models but are not as accurate as building complex nonlinear models. The nonlinear model requires higher sample set capacity and depends on the optimization of parameters and extraction of features.

The prediction of quality indicators such as gasoline octane number is usually modeled by combining nonlinear models with intelligent algorithms. Since the actual octane detection process often has the disadvantages of slow detection speed and large amount of pollutants emitted, NIR spectroscopy gradually becomes the mainstream octane detection method. The BP neural network is a very mature multilayer feedforward network, which mainly contains three parts: input, hidden, and output layers, and it is very sensitive to the initial weights and has different convergence speeds when given different initial values. The key to detect-

ing octane values by NIR spectroscopy is to build a mathematical model, and BP neural networks are widely used in octane prediction problems because of their strong generalization, self-adaptive capability, and ability to approximate any nonlinear connectivity function with arbitrary accuracy. Octane loss modeling using BP begins with preprocessing the data to filter out some of the variables with the highest correlation to octane values. The number of neurons of the network directly affects the prediction results, and the BP neural network is sensitive to the initial parameters, and some optimization algorithms have been proposed by scholars in order to accelerate its convergence [4, 22–25].

2.2. BP Neural Network. The BP neural network was firstly proposed by Rumelhart and McClelland in 1986 [26]. This algorithm mainly includes two calculation processes [15, 27–29]. The first is to propagate the output error through the direction from input to output and, at the same time, to continuously adjust the weights and thresholds according to the training objectives of the network. If the actual output is not consistent with the expected output, it is necessary to switch to the second calculation stage, i.e., the error backpropagation process. In the second process, the input layer is retransmitted layer by layer to decrease the error, which adjusts parameters along the gradient direction. Through learning and training these two processes repeatedly, the network weight and threshold corresponding to the minimum error are determined, and the network model is created, leading to the end of the model. The algorithm pseudo-code is described in Algorithm 1.

The output of a neuron j on the output and hidden layers of the BP neural network is formulated as follows:

$$O_j = f_j(\text{Net}_j) = f_j\left(\sum_{i=1}^l w_{ij}x_i + b_j\right), \quad (1)$$

where x_i denotes the individual input values of neuron j , O_j denotes the output value of neuron j , w_{ij} denotes the individual connection weights between the corresponding input i and neuron j , f_j denotes the activation function of neuron j , the sigmoid function $y = 1/(1 + e^{-x})$ is commonly used as the activation function, and b_j denotes the threshold value of neuron j .

The commonly used empirical formula for the number of implicit layers is

$$h = \sqrt{m + n} + a. \quad (2)$$

Also, the loss function of the error is

$$F(w, b) = \frac{1}{2} \sum_{i=1}^M (d_i - O_i(w, b))^2, \quad (3)$$

```

BPBtrain(){
  Initialize network's weights and thresholds;
  While termination conditions are not met {
    For each training sample X in the samples {
  // Forward propagation of inputs
  For Each cell j of the hidden or output layer {
     $I_j = \sum_j w_{ij} O_j + \theta_j$ ; ;
    // The net input to the computational cell is relative to the previous layer i
     $O_j = 1/(1 + e^{-I_j})$ ;
    // Calculate the output of cell j. Choose the sigmod function as the activation function
  } // Reverse propagation error
  For each cell j of the output layer{
     $Err_j = O_j(1 - O_j)(T_j - O_j)$ ; // Calculate error
  }
  From the last to the first hidden layer, for each cell j of the hidden layer{
     $Err_j = O_j(1 - O_j) \sum_k Err_k w_{kj}$ ;
    // k is a neuron in the next layer of j
  }
  For each weight wij in the network {
     $\Delta W_{ij}(l) = Err_j O_j$ ;
    // weighted value added, where is the learning rate
     $W_{ij} = W_{ij} + \Delta W_{ij}$ ;
    // weight update
  }
  For each deviation in the network {
     $\Delta \theta_j = (l) Err_j$ ; // Value added deviation
     $\theta_j = \theta_j + \Delta \theta_j$ ; // Deviation update
  }
  }
}

```

ALGORITHM 1: BP neural network training algorithms.

where w and b denote the weight and threshold, respectively; M is the number of samples; d_i denotes the expected result of the i_{th} sample; O_j records the actual network output of the i_{th} sample; and $f(w, b)$ is the loss function.

2.3. Genetic Algorithm. In 1970, Professor Holland proposed the genetic algorithm, which is a self-adaptive global optimization search algorithm. This algorithm is efficient, practical, and robust and has been widely used in different fields [30–33], such as machine learning, pattern recognition, neural networks, control system optimization, and social sciences [34–38], by repeating three key operations on the current population, that is, selection, crossover, and mutation. To help the population gradually evolve to a state close to the optimal solution, it uses population search techniques, which can be realized through repeating three key operations (i.e., selection, crossover, and mutation) on the current population [39–43]. The genetic algorithm is an encoding of the problem parameters to be optimized, and its basic operations are as follows:

(i) The population initialization: randomly generate N individuals as initialized populations to ensure sufficient diversity in the population

- (ii) The fitness function: it is the criterion to distinguish the good and bad individuals in the population, and the direction of increasing fitness function is in the same direction as the change of the genetic algorithm
- (iii) Selection arithmetic: application of the roulette wheel selection method to select chromosomes into the next generation according to their cumulative probability
- (iv) Crossover operation: randomly selecting two chromosomes and then generating a random number to produce two new individuals, which is crossed over if it is less than the crossover rate
- (v) The mutation operation randomly selects a chromosome and generates a random number, which mutates if it is less than the mutation rate, and then, a new individual is produced
- (vi) Population update: the selected solutions by genetic manipulation are saved and finally an optimal population is obtained

The pseudo-code description of the genetic algorithm [34] is shown in Algorithm 2.

```

Parents < - { Randomly generated populations }
While not (Termination condition)
  Calculate the fitness of each parent in the population
Children < - ∅
While | Children| < |Parents|
  Using fitness to select a mating pair of sires based on probability
  The parents mated to produce offspring c1 and c2
  Children < - Children{c1, c2}
Loop
  Some offspring random mutation
Parents < - Children
Next Generation

```

ALGORITHM 2: Genetic algorithm.

3. Optimized Neural Network Model

3.1. Establishment of the Neural Network Model

3.1.1. Building the BP Neural Network Model. The gasoline octane number rationalization index contains 401 feature data; thus, the input port neurons of the neural network are 401 and the output port neuron is 1. According to the best test results of the empirical formula, the number of neurons in the hidden layer is set at 25. The neural network model is constructed as shown in Figure 1, and the sigmoid function is used for the activation function of the hidden layer and the output layer.

There are still some shortcomings in the BP neural network: (1) the learning rate of the neural network is determined by experimental experience, and it is difficult to find the optimal value and (2) the initial weights and thresholds of the neural network are randomly generated, which is easy to obtain the local optimal values and affects the model prediction performance. Therefore, the following improvements are made to address these problematic topics: (1) the learning rate is no longer set at a fixed rate, and the self-adaptive learning rate is used to improve the learning efficiency of the network and (2) the genetic algorithm is very effective in finding optimal values, so it can be used to find the optimal weights and thresholds, preventing the local optima and improving the model prediction effect.

3.1.2. Learning Rate Optimization Based on the Self-Adaptive Algorithm. Traditional neural networks are trained with a fixed learning rate, which has a great impact on the training results. Unfortunately, it is uncertain about the learning rate. If the learning rate is too small, the more training times are required and the network converges are slower, while if the learning rate is too large, the stability of the network structure is poor. Therefore, it is important to examine the method of the self-adaptive learning rate. The formula is as follows:

$$\gamma_{t+1} = \begin{cases} e^{0.002} * \gamma_t, & \Delta E < 0, \\ e^{-0.002} * \gamma_t, & \Delta E \geq 0, \end{cases} \quad (4)$$

where γ_t is the learning rate used at the t_{th} training, γ_{t+1} is the learning rate used at $(t+1)_{th}$, and ΔE is the amount of variation in error.

3.2. BP Neural Network Model Optimized by the Genetic Algorithm

3.2.1. Improvement of the Genetic Algorithm

(1) *Initial population and individual codes.* N individuals are randomly generated as the initial population, considering that the number of input nodes of the neural network is m , the number of nodes of the hidden layer is h , and the number of nodes of the output layer is n . The following issues should be noted when coding chromosomes: (1) the weight matrix is a two-dimensional matrix, while chromosomes are one-dimensional; (2) there are multiple weights and thresholds, but a chromosome is fixed. Thus, the two-dimensional matrix of weights is mapped into a one-dimensional matrix, and multiple weights and thresholds are spliced into chromosomes. Using real number encoding, each chromosome is actually a string of real numbers, which consists of connected weights and thresholds for each layer of the network model. The formula to determine the length of individual chromosomes is as follows:

$$L = m * h + n * h + n + h. \quad (5)$$

(2) *Fitness function.* The fitness function is a criterion to distinguish the good and bad individuals in the population. For the randomly generated weights and thresholds, the resulting error is calculated, and the direction of the increasing fitness function follows the same direction of the genetic algorithm evolution. Here, the inverse of the loss function is chosen as the fitness function:

$$\text{Fit} = \frac{1}{1/2 \sum_{i=1}^M (d_i - O_j)^2}, \quad (6)$$

where M denotes the number of training set samples, d_i denotes the expected output of the i_{th} training sample

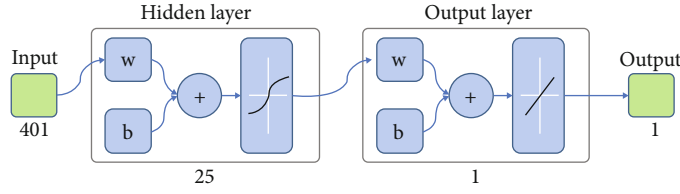


FIGURE 1: The structure of the neural network model.

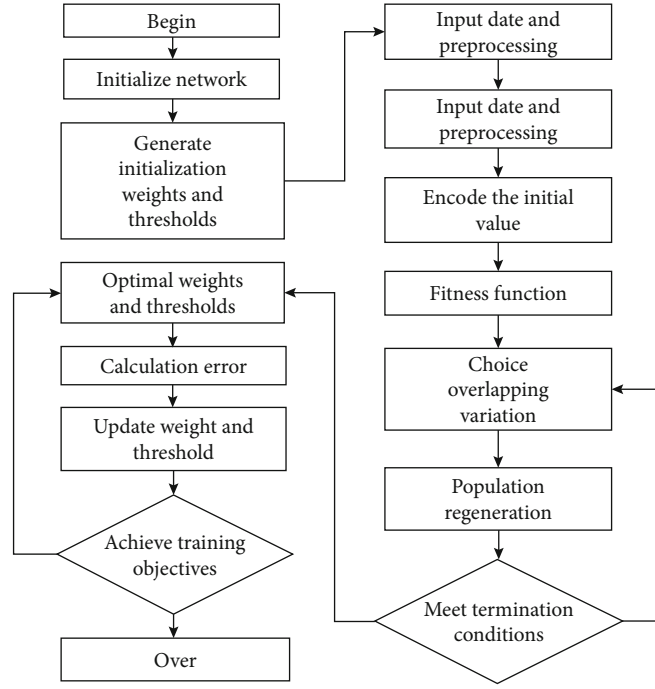


FIGURE 2: The flow chart of the specific GA-BP network model.

network, and O_i denotes the actual output of the i_{th} training sample network.

(3) *Selection operation.* The selection operation uses the roulette wheel selection method, whereby chromosomes are selected to produce populations with the same number of individuals as N populations. In the selection process, there may be duplicate individuals, and duplicate individuals are irrelevant when crossover is performed. Therefore, duplicate individuals are also eliminated during the selection process.

(4) *Crossover operations.* The crossover operation uses a single-point crossover, in which two paired individuals are selected from the initial population. During the process, a random crossover point is set, and parts of the chromosomes are swapped with the formation of two new individuals.

The purpose of the crossover operation is to generate new population individuals to improve the population diversity, and thus, the value of the crossover rate is of great importance to the performance of the genetic algorithm. Generally, the standard genetic algorithm uses a fixed cross-

over rate, which leads to problems such as premature algorithm or slow convergence. If the crossover rate is too small, it is difficult for the population to produce good individuals. If the crossover rate is too large, it is difficult to retain the good individuals in the population in the later stage of the algorithm. Therefore, the crossover rate used in this paper is varied with the change of fitness value, and the formula is as follows:

$$P_C = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f' - f_{avg})}{f_{max} - f_{avg}}, & f' \geq f_{avg}, \\ P_{c1}, & f', \end{cases} \quad (7)$$

where $P_{c1} = 0.99$, $P_{c2} = 0.4$, f_{max} is the largest fitness value in the population, f_{avg} is the average fitness value of the population per generation, and f' is the larger fitness value of the two individuals that will cross over.

(5) *Mutation operations.* Due to the long length of the chromosome, it is not suitable to choose the traditional single-point mutation operation. The number of mutation sites

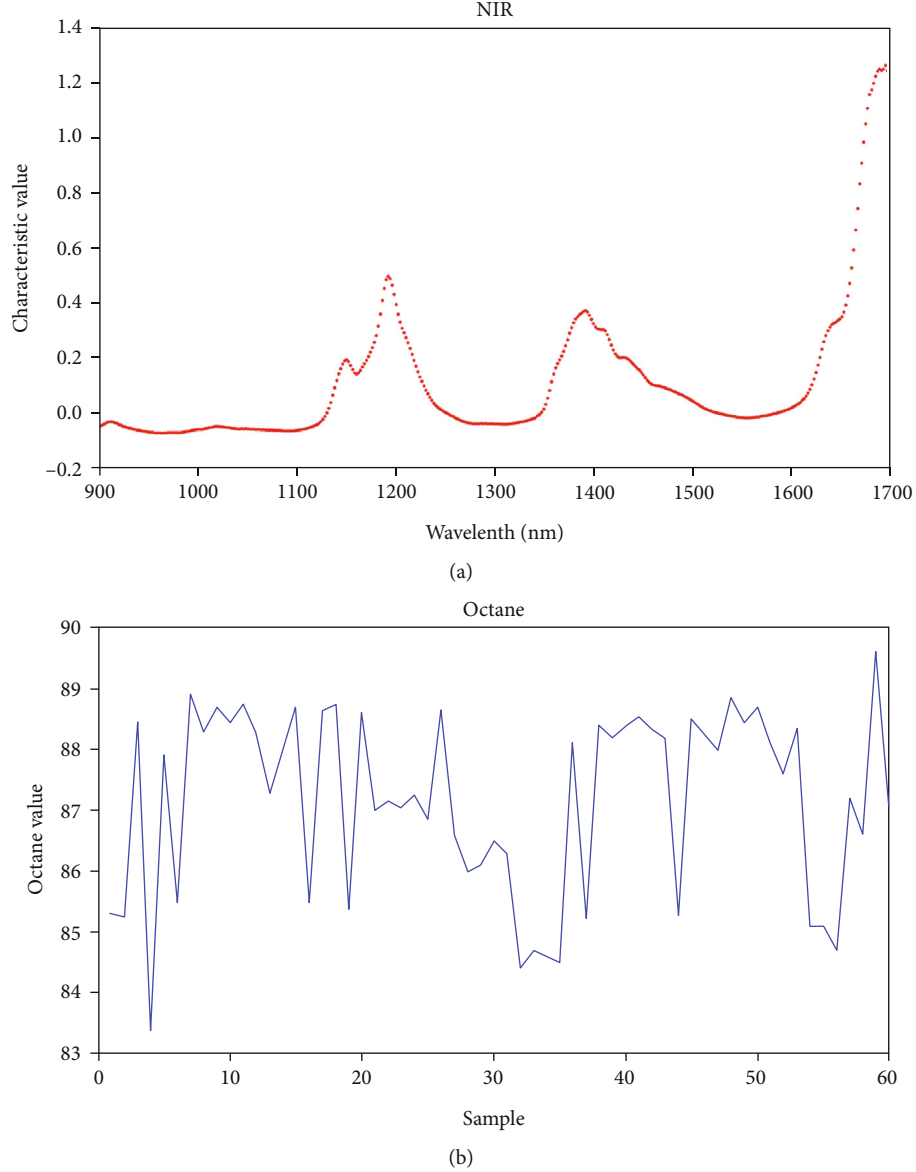


FIGURE 3: The dataset (a) shows the NIR spectral analysis results of one group of gasoline samples and (b) shows the octane number of 60 groups of gasoline samples.

changes, and the method of self-adaptive mutation sites is used with the following equation:

$$L = L \left[L_{\max} \left(1 - \frac{G}{G_{\max}} \right) \right]. \quad (8)$$

When the mutation operator mutates the k_{th} gene of an individual a with a certain probability. The mutation operation used is as follows:

$$a_k' = \begin{cases} a_k + (a_{\max} - a_k) * r_1 \left(1 - \frac{G}{G_{\max}} \right)^2 r_2 > 0.5, \\ a_k + (a_{\min} - a_k) * r_1 \left(1 - \frac{G}{G_{\max}} \right)^2 r_2 \leq 0.5, \end{cases} \quad (9)$$

where a_k denotes a mutation in the k_{th} gene of an individual a , a_k' is the mutated individual, a_{\max} and a_{\min} are the upper and lower bounds of individual gene values, G is the current number of iterations, G_{\max} is the maximum number of iterations, r_1 and r_2 are random numbers between $[0, 1]$, and L_{\max} is the predetermined maximum number of mutated bits. The advantages of using this variation operation are as follows: (1) the setting of the random number, r_1 , can influence the degree of variation; (2) the setting of the random number, r_2 , can ensure that the gene value increases or decreases with equal probability, while the existence of the upper and lower bounds of the gene value ensures an appropriate variation of the gene value; and (3) the self-adaptive module used to adjust the number of variance bits and take into account the balanced search ability of the algorithm in both global and local. The degree

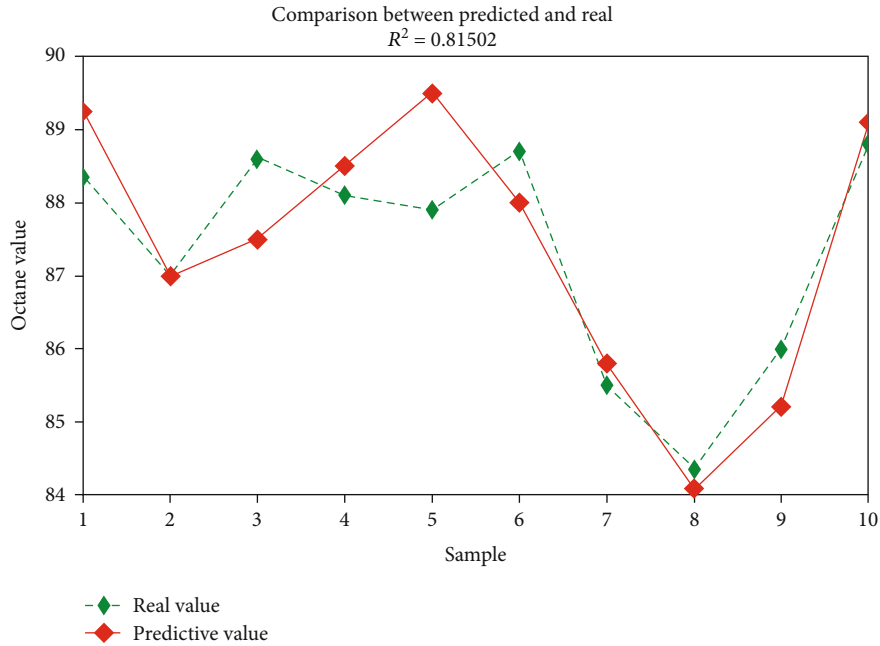


FIGURE 4: The results of the predicted values were compared with the real gasoline octane number in the BP model.

of variance decreases gradually with the increase in iterations, which ensures the strong global search ability at the beginning and the local search ability of the algorithm at the later stage. It can greatly prompt the individual to converge to the global optimal solution.

3.2.2. *Optimization of Weights and Thresholds for the BP Neural Network.* The processes of building a BP neural network (as described in Algorithm 1) based on the self-adaptive genetic algorithm (as described in Algorithm 2) optimization are as follows:

- (1) Creating a BP neural network, randomly generating initial weights and thresholds, initializing populations, coding with real numbers, and determining the number of populations
- (2) Calculating the fitness function and selecting the best individuals according to the roulette selection method and inserting them into the next-generation population
- (3) Generating new individuals by crossover and mutation in the new generation of populations
- (4) Reinserting new individuals into the population and calculating their fitness values
- (5) Terminating the algorithm if a satisfactory individual can be found; otherwise, go back to step (2)
- (6) After finding the optimal individual, the individual is decoded to obtain the optimized weights and thresholds, which are then used in the BP neural network

The flow chart of the specific GA-BP network model is shown in Figure 2.

4. Experiment

4.1. *Experimental Dataset.* In the experiment, 60 groups of gasoline samples were selected and analyzed by Fourier near infrared spectroscopy (900-1700 nm). The wavelength point was taken as an eigenvalue at an interval of 2 nm, and 401 eigenvalues were obtained to form the dataset (i.e., spectra_data.mat), containing two sets of values, matrix NIR and matrix octane. Among them, NIR stores the physico-chemical data of gasoline collected by infrared spectroscopy. Octane stores the real octane number corresponding to these 60 eigenvalues. Figure 3(a) shows the NIR spectral analysis result of one specific group of gasoline samples, and Figure 3(b) shows the octane number of a total of 60 groups of gasoline samples.

In this experiment, the dataset is randomly split into two parts, in which 50 * 401 data are used as the training set and the other 10 * 401 data are used as the test set. Since the order magnitude of each feature data of gasoline octane number is inconsistent, this will affect the final mapping results. The dataset needs to be normalized by the following formula:

$$t = \frac{t - t_{\min}}{t_{\max} - t_{\min}}, \quad (10)$$

where t_{\min} is the minimum value for each column of data, t_{\max} is the maximum value for each column of data, and t is the value to be normalized.

Further, the performance evaluation of the gasoline octane number prediction model is divided into two parts: the relative error E and the coefficient of determination R_2 , which are defined as follows:

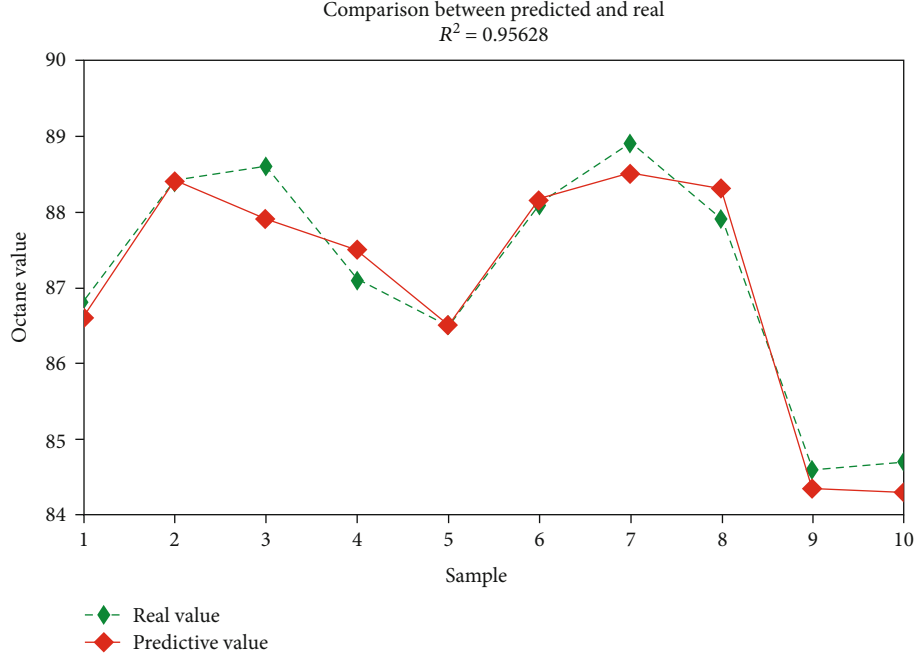


FIGURE 5: The results of the predicted values were compared with the real gasoline octane number in the GA-BP model.

$$E_i = \frac{|\hat{y}_i - y_i|}{y_i} \quad (i, 2 \dots, n),$$

$$R_2 = \frac{\left(l \sum_{i=1}^l \hat{y}_i y_i - \sum_{i=1}^l \hat{y}_i \sum_{i=1}^l y_i \right)^2}{\left(l \sum_{i=1}^l \hat{y}_i^2 - \left(\sum_{i=1}^l \hat{y}_i \right)^2 \right) \left(l \sum_{i=1}^l y_i^2 - \left(\sum_{i=1}^l y_i \right)^2 \right)}, \quad (11)$$

where \hat{y}_i is the predicted value of the i_{th} sample, y_i is the true value of the i_{th} sample, and n is the number of samples. It is clear that the smaller the relative error, the better the performance of the model. The closer the coefficient of determination to 1 in the range of [0,1], the better the performance of the model. Conversely, the closer the coefficient of determination tends to 0, the worse the performance of the model.

4.2. Experimental Results and Simulation Analysis. The BP neural network model uses the BP network described in Section 3.1, with an initial learning rate of $\lambda = 0.1$, a maximum number of iterations set to 2000, and a minimum acceptable error set to 0.001. The training set was first applied to train the constructed model, and then, the validation test of the model was completed with the test set. Finally, the predicted values of the model were compared with the real gasoline octane number in the dataset, and the results were obtained as shown in Figure 4. The model performance index R^2 was 0.81502, which shows that the BP neural network model can predict the gasoline octane number. Unfortunately, the limitations of the traditional model lead to a low accuracy of the prediction.

TABLE 1: The comparison of the relative errors produced by 10 predictions for the two models.

| No. | BP | GA-BP |
|-----|--------|--------|
| 1 | 0.0073 | 0.0035 |
| 2 | 0.0092 | 0.0061 |
| 3 | 0.0108 | 0.0040 |
| 4 | 0.0031 | 0.0024 |
| 5 | 0.0026 | 0.0071 |
| 6 | 0.0071 | 0.0035 |
| 7 | 0.0030 | 0.0065 |
| 8 | 0.0064 | 0.0045 |
| 9 | 0.0057 | 0.0034 |
| 10 | 0.0055 | 0.0063 |

Based on the optimized BP neural network based on the genetic algorithm designed in Section 3.2, the gasoline octane number GA-BP model is established. In the genetic algorithm, the population size is set to 100, and the maximum number of iterations is 500. In this model, the crossover rate is varied by self-adaptive, the variation rate is 0.09, the upper bound of the gene value is 1, the lower bound of the gene value is -1, and the maximum number of variation bits is set to 10. The same comparative experiments were used, and the experimental results are shown in Figure 5. The model performance index R_2 was 0.95628. The results show that the accuracy of the optimized BP neural network for predicting the gasoline octane number is improved by 14%.

In order to obtain accurate judgment of the experimental results, the evaluation index was the average error of the loss function through multiple experiments, resulting from the

randomness of some parameters in the experiments. Each set of data was summed and averaged the loss values through the loss function. The data in Table 1 shows the comparison of the relative errors produced by 10 predictions for the two models, respectively.

It is notable in Table 1 that the average value of the relative error is 0.0061 in the traditional BP neural network, while the average value of the relative error of the BP neural network optimized by the self-adaptive genetic algorithm is 0.0047, which reduces the error value by 23%.

5. Conclusion

Octane number is an important metric describing gasoline's ability to resist detonation and combustion in the cylinder; the higher the value, the better the resistance to detonation. The accurate prediction of octane loss during gasoline refining could facilitate production management and ensure gasoline octane. A neural network, which is with excellent performance in dealing with nonlinear system problems, is widely used in a number of fields. However, there are still some deficiencies. In this paper, we optimize the weights and thresholds of the neural network by the self-adaptive genetic algorithm and self-adaptively adjust the learning rate to improve the accuracy and generalization ability of the model. A novel GA-BP model was established, and this model was used for gasoline octane number prediction. Through the comparison of simulation results, the GA-BP model has more accurate prediction ability and better generalization ability than the traditional BP model. In one specific experiment, the model performance index decision coefficient R_2 was improved from 0.81502 to 0.95628, and the 10-experiment average prediction error was reduced from 0.0061 to 0.0041. In the future, we will work towards further improving algorithm performance. The prediction accuracy should be further improved; meanwhile, the error value should not increase. Besides, other intelligent algorithms (e.g., extreme gradient boosting) will be tested and tailored for this industrial context.

Data Availability

The data that support the findings of this study are available on request from the corresponding author.

Conflicts of Interest

The authors declare no conflict of interest.

References

- [1] D. Bogdan and O. Ion, "Octane number estimation using neural networks," *Revista de Chimie*, vol. 65, pp. 599–602, 2014.
- [2] Y. Ryzhov, E. S. Strizhakova, and A. Lapidus, "Modeling the octane numbers of alkenes by the inverse function method," *Petroleum Chemistry*, vol. 51, no. 5, pp. 354–362, 2011.
- [3] S. Kish, A. Rashidi, H. Aghabozorg, and L. Moradi, "Increasing the octane number of gasoline using functionalized carbon nanotubes," *Applied Surface Science*, vol. 256, no. 11, pp. 3472–3477, 2010.
- [4] S. Wang, S. Liu, J. Zhang, X. Che, Z. Wang, and D. Kong, "Feasibility study on prediction of gasoline octane number using NIR spectroscopy combined with manifold learning and neural network," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 228, article 117836, 2020.
- [5] B. Li and C. Qin, "Predictive analytics for octane number: a novel hybrid approach of KPCA and GS-PSO-SVR model," *IEEE Access*, vol. 9, pp. 66531–66541, 2021.
- [6] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, "Deep-learning-enhanced multitarget detection for end-edge-cloud surveillance in smart IoT," *IEEE Internet of Things Journal*, vol. 8, no. 16, pp. 12588–12596, 2021.
- [7] F. Lu, R. Niu, Z. Zhang, L. Guo, and J. Chen, "A generative adversarial network-based fault detection approach for photovoltaic panel," *Applied Sciences*, vol. 12, no. 4, p. 1789, 2022.
- [8] D. Paulo, J. Barros, and P. Barbeira, "A PLS regression model using flame spectroscopy emission for determination of octane numbers in gasoline," *Fuel*, vol. 176, pp. 216–221, 2016.
- [9] X. Sun, G. Yuan, and J. M. Dai, "Multi-spectral thermometry based on GA-BP algorithm," *Spectroscopy and Spectral Analysis*, vol. 27, no. 2, pp. 213–216, 2007.
- [10] Z. Cai, Z. He, X. Guan, and Y. Li, "Collective data-sanitization for preventing sensitive information inference attacks in social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 577–590, 2018.
- [11] M. Zou, L. Xue, H. Gai, Z. Dang, S. Wang, and P. Xu, "Identification of the shear parameters for lunar regolith based on a GA-BP neural network," *Journal of Terramechanics*, vol. 89, pp. 21–29, 2020.
- [12] W. Liang, Y. Hu, X. Zhou, Y. Pan, and K. Wang, "Variational few-shot learning for microservice-oriented intrusion detection in distributed industrial IoT," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.
- [13] J. Cooper, K. Wise, J. Groves, and W. T. Welch, "Determination of octane numbers and Reid vapor pressure of commercial petroleum fuels using FT-Raman spectroscopy and partial least-squares regression analysis," *Analytical Chemistry*, vol. 67, no. 22, pp. 4096–4100, 1995.
- [14] H. Chung, S. Cho, Y. Toyoda, K. Nakano, and M. Maeda, "Moment combined partial least squares (MC-PLS) as an improved quantitative calibration method: application to the analyses of petroleum and petrochemical products," *Analyst*, vol. 131, no. 5, pp. 684–691, 2006.
- [15] J. Kelly, C. Barlow, T. Jinguji, and J. B. Callis, "Prediction of gasoline octane numbers from near-infrared spectral features in the range 660–1215 nm," *Analytical Chemistry*, vol. 61, no. 4, pp. 313–320, 1989.
- [16] P. Flecher, W. Welch, S. Albin, and J. B. Cooper, "Determination of octane numbers and Reid vapor pressure in commercial gasoline using dispersive fiber-optic Raman spectroscopy," *Spectrochimica Acta Part A: Molecular and Biomolecular Spectroscopy*, vol. 53, no. 2, pp. 199–206, 1997.
- [17] G. Fodor, K. Kohl, and R. Mason, "Analysis of gasolines by FT-IR spectroscopy," *Analytical Chemistry*, vol. 68, no. 1, pp. 23–30, 1996.
- [18] N. P. 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.
- [19] K. Brudzewski, A. Kesik, K. Kołodziejczyk, U. Zborowska, and J. Ulaczyk, "Gasoline quality prediction using gas

- chromatography and FTIR spectroscopy: an artificial intelligence approach," *Fuel*, vol. 85, no. 4, pp. 553–558, 2006.
- [20] R. Balabin, R. Safieva, and E. Lomakina, "Comparison of linear and nonlinear calibration models based on near infrared (NIR) spectroscopy data for gasoline properties prediction," *Chemometrics and Intelligent Laboratory Systems*, vol. 88, no. 2, pp. 183–188, 2007.
- [21] I. Litani-Barzilai, V. B. Sela, V. Bulatov, I. Zilberman, and I. Schechter, "On-line remote prediction of gasoline properties by combined optical methods," *Analytica Chimica Acta*, vol. 339, no. 1-2, pp. 193–199, 1997.
- [22] Y. Deng, H. Xiao, J. Xu, and H. Wang, "Prediction model of PSO-BP neural network on coliform amount in special food," *Saudi journal of biological sciences*, vol. 26, no. 6, pp. 1154–1160, 2019.
- [23] C. Huang, Y. Zhao, W. Yan, Q. Liu, and J. Zhou, "A new method for predicting crosstalk of random cable bundle based on BAS-BP neural network algorithm," *IEEE Access*, vol. 8, pp. 20224–20232, 2020.
- [24] Z. Cai, Z. Xiong, H. Xu, P. Wang, W. Li, and Y. Pan, "Generative adversarial networks," *ACM Computing Surveys*, vol. 54, no. 6, pp. 1–38, 2021.
- [25] R. Zong, Y. Zhi, B. Yao, J. Gao, and A. A. Stec, "Classification and identification of soot source with principal component analysis and back-propagation neural network," *Australian Journal of Forensic Sciences*, vol. 46, no. 2, pp. 224–233, 2014.
- [26] X. Zhou, X. Xu, W. Liang et al., "Intelligent small object detection for digital twin in smart manufacturing with industrial cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1377–1386, 2022.
- [27] X. Li, S. Dong, H. S. Mohamed, G. Al Aqel, and N. Pirhadi, "Prediction of tubular T/Y-joint SIF by GA-BP neural network," *KSCE Journal of Civil Engineering*, vol. 24, no. 9, pp. 2706–2715, 2020.
- [28] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 5, pp. 968–979, 2020.
- [29] X. Xue, Y. Li, X. Yang, X. Chen, and J. Xiang, "Prediction of slope stability based on GA-BP hybrid algorithm," *Neural Network World*, vol. 25, no. 2, pp. 189–202, 2015.
- [30] Z. Cao, N. Guo, M. Li, K. Yu, and K. Gao, "Back propagation neural network based signal acquisition for Brillouin distributed optical fiber sensors," *Optics Express*, vol. 27, no. 4, pp. 4549–4561, 2019.
- [31] V. Kostenko and A. Frolov, "Self-learning genetic algorithm," *Journal of Computer and Systems Sciences International*, vol. 54, no. 4, pp. 525–539, 2015.
- [32] C. Wei, J. Qin, L. Hao, J. Guo, and Y. X. Chen, "A noise based medical elites silence model and public health opinion distortion in social networks," *Frontiers in Public Health*, vol. 9, 2022.
- [33] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. Wang, "Hierarchical adversarial attacks against graph neural network based IoT network intrusion detection system," *IEEE Internet of Things Journal*, 2021.
- [34] J. Guo, S. Chen, Y. Li, J. L. Li, and J. Lu, "A blockchain and IoT based lightweight framework for enabling information transparency in supply chain finance," *Digital Communications and Networks*, 2022.
- [35] Z. Cai and Z. He, "Trading private range counting over big IoT data," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, Dallas, TX, USA, 2019.
- [36] Y. Quan and Q. Feng, "A hybrid genetic algorithm for twice continuously differentiable NLP problems," *Computers & Chemical Engineering*, vol. 34, no. 1, pp. 36–41, 2010.
- [37] Y. Huang and M. Fei, "Motion planning of robot manipulator based on improved NSGA-II," *International Journal of Control, Automation and Systems*, vol. 16, no. 4, pp. 1878–1886, 2018.
- [38] X. Zhou, X. Yang, J. Ma, and K. Wang, "Energy efficient smart routing based on link correlation mining for wireless edge computing in IoT," *IEEE Internet of Things Journal*, 2021.
- [39] J. Zhou and Q. Ma, "Establishing a genetic algorithm-back propagation model to predict the pressure of girdles and to determine the model function," *Textile Research Journal*, vol. 90, no. 21-22, pp. 2564–2578, 2020.
- [40] J. Liu, H. Wang, Y. Sun, C. Fu, and J. Guo, "Real-coded quantum-inspired genetic algorithm-based BP neural network algorithm," *Mathematical Problems in Engineering*, vol. 2015, Article ID 571295, 10 pages, 2015.
- [41] C. Li, J. Li, Y. Li, L. He, X. Fu, and J. Chen, "Fabric defect detection in textile manufacturing: a survey of the state of the art," *Security and Communication Networks*, vol. 2021, Article ID 9948808, 13 pages, 2021.
- [42] Z. Cai and X. Zheng, "A private and efficient mechanism for data uploading in smart cyber-physical systems," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2020.
- [43] Y. Zeng, J. Chen, N. Jin, Y. Du, and X. Jin, "Air quality forecasting with hybrid LSTM and extended stationary wavelet transform," *Building and Environment*, vol. 213, article 108822.