

Research Article

Concrete Compression Test Data Estimation Based on a Wavelet Neural Network Model

Haiying Wang ¹, Xinping Wang ¹, Chao Wang ¹ and Jian Xu ²

¹School of Construction Machinery, Chang'an University, Xi'an 710064, China

²Xi'an University of Science and Technology, Xi'an 710054, China

Correspondence should be addressed to Haiying Wang; whying@chd.edu.cn

Received 4 September 2018; Revised 12 January 2019; Accepted 29 January 2019; Published 11 February 2019

Academic Editor: Alberto Campagnolo

Copyright © 2019 Haiying Wang 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.

Firstly, a genetic algorithm (GA) and simulated annealing (SA) optimized fuzzy c-means clustering algorithm (FCM) was proposed in this paper, which was developed to allow for a clustering analysis of the massive concrete cube specimen compression test data. Then, using an optimized error correction time series estimation method based on the wavelet neural network (WNN), a concrete cube specimen compressive strength test data estimation model was constructed. Taking the results of cluster analysis as data samples, the short-term accurate estimation of concrete quality was carried out. It was found that the mean absolute percentage error, e_1 , and the root mean square error, e_2 , for the samples were 6.03385% and 3.3682KN, indicating that the proposed method had higher estimation accuracy and was suitable for concrete compressive test data short-term quality estimations.

1. Introduction

Expressway and railway construction projects have become more dependent on information technology in the past decade [1]; however, much of the collected information is not being extracted or effectively utilized [2]. Much of this construction data is related to laboratory based concrete cube specimen compression tests, the quality of which directly affects the quality of the whole project and is relevant to the project operations and maintenance stages. Because concrete quality has far-reaching impacts on the overall project, strengthening compressive test process monitoring is vital to construction safety and project success. While test machine data is monitored over a long period of time, it is also necessary to make accurate short-term estimations based on previous test data to identify any possible problems before any abnormalities occur and to make corrections to ensure the concrete quality being used for the project. Compared with the traditional laboratory sampling of concrete cube specimen, conducting concrete quality assessments by applying time series estimation algorithms to the test data can more effectively utilize the massive information data and more accurately estimate the concrete compressive data.

There has been significant research conducted on estimation methods, with the most classical estimation algorithms being support vector machines (SVM), WNN, and decision trees. While SVM methods [3] use statistical theory to minimize structural risk, when there is a large data quantity, the algorithm is slow, takes up a great deal of computer memory, and is unable to resolve multiclassification problems [4]. Consequently, many alternative optimization methods have been proposed. In [5], for example, a cuckoo search (CS) algorithm was used to optimize the unknown parameters in a support vector machine model and in [6], a decision tree inductive algorithm was applied to classify specific data. However, when there is incomplete data, decision tree data performance degrades, which leads to overfitting and uneven distributions [7]. For that, Yang and Fong (2013) [8] developed an incremental optimization mechanism to optimize fast decision trees. The compression test data classifications were fixed and the data volume was large, which can be suitably predicted using artificial neural network (ANN). Filik (2016) developed a new hybrid approach for wind speed estimation using a fast block least mean square algorithm and an artificial neural network [9]; however, artificial neural networks have been found to have inherent faults such as weak error-tolerance and missing information [10].

WNN, which apply wavelet theory to neural networks, have been found to make up for the lack of a Fourier Transformation in time domains when predicting time series problems [11, 12]. For example, Zhang and Wu (2015) [13] proposed a GA-WNN model to achieve optimization of piezoresistive pressure sensors and corresponding measurement systems. Sharma and Yang (2016) [14] proposed a mixed WNN for short-term solar irradiance forecasting and compared three algorithms with the mixed WNN wavelet algorithm and proved that the wavelet algorithm significantly reduced estimation errors in solar radiation. Guan and Luh (2013) [15] used an improved WNN to solve a short-term load forecasting problem without an estimated forecast interval and Falamarzi and Palizdan (2016) [16] used a wavelet transformation to decompose input data and improve the estimation accuracy of a transpiration estimation model.

Therefore, WNN has been found to have good estimation performances and better practicability and application for compression data. On this basis, this paper applies wavelet technology to ensure effective estimations. While previous research has mainly focused on pressure transmitter research, solar radiation estimations, short-term load forecasting, and other fields, there has been no research to date that has examined concrete test data estimating methods for engineering construction nor the methods needed for the sample extraction and classification of mass test monitoring data.

In this paper, a time series estimation method based on WNN is therefore proposed, with the fracture load, $F(\text{KN})$, and the compressive strength, $f_{\text{cu}}(\text{MPa})$, being selected as the concrete cube specimen compressive data attributes. A FCM algorithm is optimized using a developed simulated annealing algorithm and genetic algorithm (SA+GA) applied to classify the test specimens from a YAW-2000 compression-testing machine in the TJ-01 contract section of the Shaanxi Xi'an-Hancheng Intercity Railway. Then, by using the improved WNN algorithm, the estimation model is used to train the clustering results, after which the trained estimation model is employed to estimate the test data. Finally, compare the estimation data with the actual measurements to verify the accuracy and the validity of the proposed method.

The remainder of this paper is organized as follows. In Section 2, a hybrid optimization FCM algorithm based on the GA+SA is proposed to cluster the sample data. In Section 3, an improved time series estimation model based on WNN is proposed. Section 4 uses the clustering method proposed in Section 2 to cluster the sample data, after which the sample data are trained to construct the compression test data estimation model, and the measured data compared with the estimation data from the estimation model. Finally, concluding comments are given in Section 5.

2. Optimized FCM Clustering Based on the GA+SA

All the data in this paper comes from the information management system of the Shaanxi Xi'an-Hancheng Intercity Railway construction project. The concrete cube specimen

compressive test data from the YAW-2000 compression-testing machine of the TJ-01 tender section are analyzed and estimated, the aim of which is to estimate the concrete quality in advance and to prevent unqualified concrete being used in the construction materials. The fracture load, $F(\text{KN})$, and the compressive strength, $f_{\text{cu}}(\text{MPa})$, are used to estimate the concrete compressive test data.

With the Lagrange multiplier as the objective function, the FCM algorithm calculates the clustering center by optimizing the objective function. However, because the initial FCM clustering center is random, it can easily fall into a local optimal solution if the clustering center is not properly chosen. Therefore, as the clustering results depend strongly on the clustering center, incorrect clustering boundary divisions and inaccurate clustering results could result [17]. To overcome this problem, following [18], a GA+SA is used to optimize the initial FCM algorithm clustering center.

2.1. Genetic Algorithm (GA). GA [19] are practical algorithms that imitate nature's "selection and survival of the fittest" evolutionary process, in which the genes adapted to the environment remain. Individuals who fit the environment best should have a better chance to propagate their offspring. Similarly, GA transform data into corresponding binary numbers, corresponding to genes in genetic process, after which a natural selection process using an adaptive function takes place. The genetic process involves genetic inheritance, genetic variation, and gene selection to eventually produce a new population, in which the binary number crossovers are heredity; the jumps from 0 to 1 in the binary numbers are the genetic mutations, and the adaptive judgment is associated with natural gene selection.

The flowchart of GA is shown in Figure 1 and the overall structure of GA is described as follows.

Step 1 (algorithm parameters initialization). For any GA, before starting to calculate, the value of the parameters should be determined, which are group number, N , crossover probability, P_c , mutation probability, P_m , and the fitness function, $f(x)$.

Step 2 (encoding). A chromosome encoding is needed to describe each chromosome in the population. The encoding method determines how the problem is structured in the algorithm and the genetic operators that are used. Each chromosome is made up of a sequence of genes from certain alphabet which can consist of binary digits (0 and 1), floating-point numbers, integers, symbols (i.e., A, B, C, and D), etc. Each chromosome represents a solution for the problem. In our GA model, taking the sample of the concrete cube specimen compressive test data as the initial population and transforming the quantitative characteristics into genes, the encoding is binary.

Step 3 (fitness evaluation). The fitness is computed for each chromosome in the current generation. The fitness value, f_i , of each chromosome (string) needs to be calculated and to be sorted by value size. The total population fitness, F_s , needs to be calculated as well.

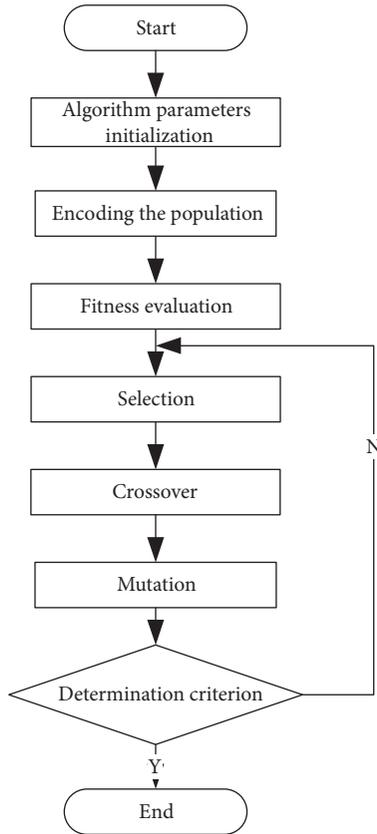


FIGURE 1: The GA flowchart.

Step 4 (selection). At each iteration, the best chromosomes are chosen for reproduction. The selection probability of each string should be calculated by using the formula of $P_i = f_i/F_s$, and the cumulative probability, P_s , should be calculated as well. The selection process can be based on the roulette method, which ensures a high probability of a large fitness value selection.

Step 5 (crossover). This is the procedure to generate new chromosomes by exchanging genes in the same position of two different chromosomes which are selected for breeding the next generation. First, each chromosome's random numbers, r , between $[0,1]$ should be computed, and if $r > P_c$, this chromosome and the other selected chromosome are selected to cross. Then, according to value of r , the crossover location of each pair chromosomes is obtained. Subsequently, the content on both sides of the intersection is exchanged.

Step 6 (mutation). In the selected chromosome, some genes in the chromosome undergo hetero-transformation. If $r > P_m$, then the gene in this position is turned over. If the gene is 1, when it is mutated, it becomes 0 and vice versa.

Step 7 (determination criterion). When the group number, N , reached and the stop criterion is satisfied, the algorithm ends and the best chromosome, together with the corresponding result, is given as an output. Otherwise, the algorithm iterates again Steps 4–7.

As GA has random selection characteristics based on probability, it can avoid the problem suffered by other optimization algorithms of falling too quickly into a local optimal solution. However, genetic algorithms also have disadvantages. After it finds the optimal solution, it prematurely ends the search, and its parameter selection and general form have no quantitative stipulation or uniform format. Yet because the general compatibility is good, it can be combined with a variety of other algorithms [20].

2.2. Simulated Annealing Algorithm (SA). When the temperature is high, the internal energy in metal is high, and the entropy value is large and therefore tends to chaos. As the temperature decreases, the entropy value of metal also decreases, and the energy commensurately decreases. If the search for the object energy value is taken as the objective function, a value can be determined using an annealing process. Therefore, the simulated annealing algorithm takes the initial temperature, T , as the starting point, sets the objective function and acceptance probability, and then continuously reduces the temperature to determine the optimal solution [21]. Therefore, it has strong local search ability and short running time advantages; however, to overcome its poor global searching abilities, so it is combined with GA [22] to get better.

The flowchart of SA is shown in Figure 2 and the overall structure of SA is described as follows.

Step 1 (initialization). Setting an initial solution, S_0 , and making the best solution, $S_{best} = S_0$, to calculate the value of the objective function, $E(S_0)$, then, setting an initial temperature, T_0 , the genetic number of iterations, G , and making $k=1$.

Step 2 (setting loop part). Do while ($T_i > T_{min}$)

(1) Produce a new state solution, S_{new} , calculate the value of the objective function, $E(S_{new})$, and $\Delta E = E(S_{new}) - E(S_0)$.

(2) If $\Delta E < 0$, make $S_{best} = S_{new}$; otherwise, if $\min\{1, \exp[-(E(S_{new}) - E(S_{best}))/T_k]\} \geq \text{random}[0,1]$, then $S_{best} = S_{new}$.

Step 3. Until the Metropolis criteria is satisfied, reduce temperature, $T_{k+1} = \text{update}(T_k)$, and $k = k + 1$.

Step 4 (terminating condition). When the stop criterion is satisfied, output the search results.

2.3. GA+SA+FCM Clustering Algorithm. The GA+SA algorithm firstly generates the initial population and then generates a new population through selection, crossover, and mutation of GA. The new population is selected and replaced by SA. When the iteration algebra reaches the genetic number of iterations, G , or the fitness value reaches the expected value, it jumps from the inner loop. After exiting, the outer loop is performed by reducing the temperature parameter, T , of SA until the temperature parameter reaches the preset value, T_d , or the algorithm converges to end the cycle. The GA+SA+FCM clustering algorithm flowchart is shown in Figure 3 and the steps of the GA + SA optimized FCM algorithm can be described as follows.

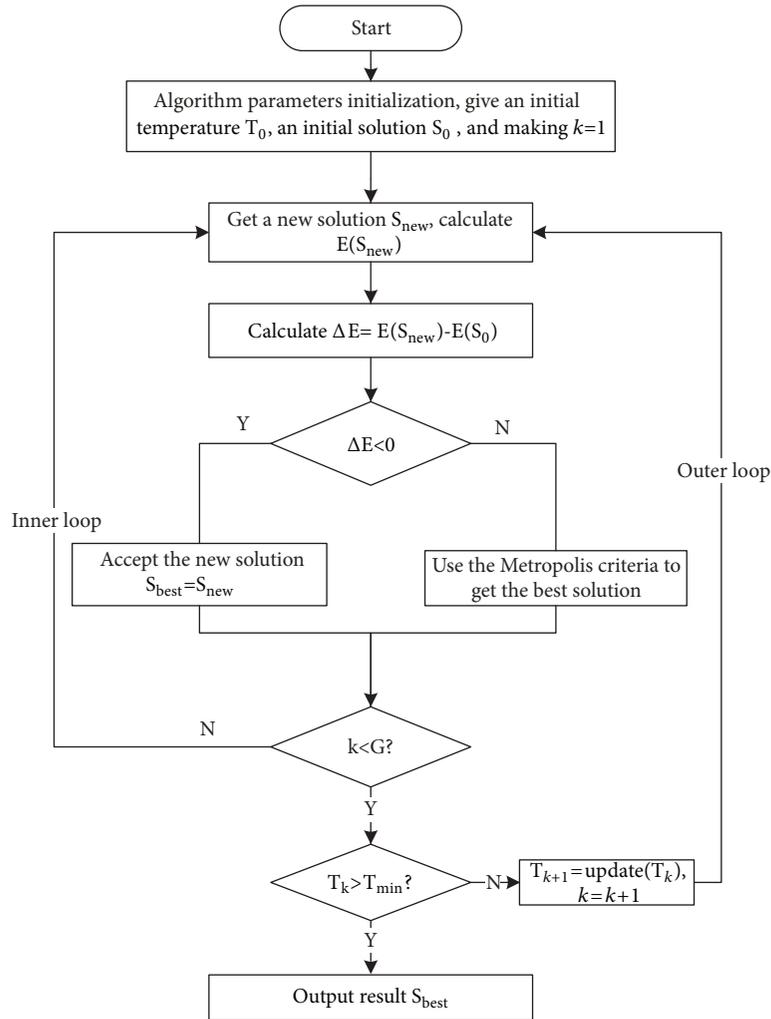


FIGURE 2: The SA flowchart.

Step 1 (parameters initialization). Set group number, N , genetic number of iterations, G , crossover probability, P_c , mutation probability, P_m , starting temperature, T_0 , cooling coefficient, q , and end temperature, T_d .

Step 2 (fitness evaluation). According the number of clusters, c , to generate cluster groups, using GA to encode the initial value and to generate the initial population, the fitness value, f_i , of each chromosome (string) needs to be calculated and sorted. The total population fitness, F_s , needs to be calculated as well.

Step 3. Set the cycle count variable $g=0$.

Step 4 (cycling part). Perform GA steps such as selection, crossover, and mutation to generate a new population. Use SA to calculate a best solution to replace the new population obtained from the GA.

Step 5. While $g < G$, set $g=g+1$ and cycle the Step 4. Otherwise, exit from the cycling part and jump to Step 6.

Step 6 (get the optimal solution). If $T_i < T_d$, then take the best solution calculated by SA as the optimal solution. Otherwise, continuously reduce the temperature to seek the optimal solution.

The GA + SA optimized FCM algorithm is used to cluster the compression test data sample. First, the number of clusters, c , is determined according to [23], and the control parameters initialized as follows: group number, $N=10$, genetic number of iterations, $G=50$, crossover probability, $P_c=0.4$, and mutation probability, $P_m=0.02$. According to [24], the control parameters are valued as follows: cooling coefficient, $q=0.95$, starting temperature, $T_0=100$, end temperature, T_d , and Metropolis chain length, L .

The GA+SA+FCM algorithm method divides the concrete compressive test specimen data into three categories, which lays the foundation for the construction of the concrete compressive test data estimation model.

The performance of the different clustering algorithmic results is evaluated by comparing the cluster validity indices [25], which are made up of a partition coefficient (PC), a

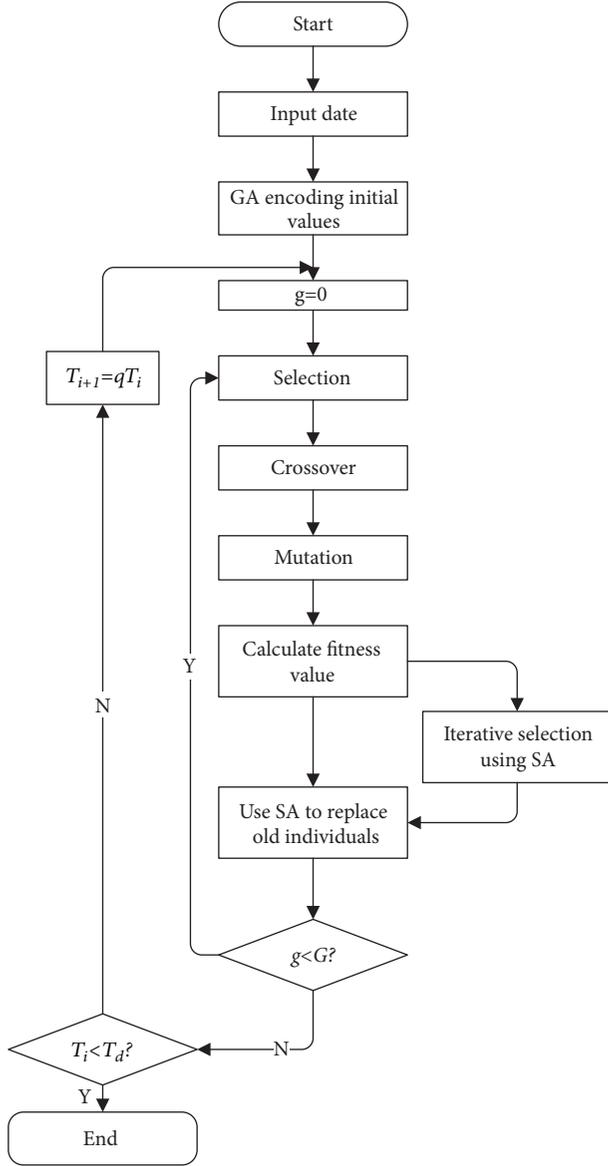


FIGURE 3: Flow of the GA+SA+FCM algorithm.

partition entropy (PE), a sector index (SC), a separation index (S), a Xie-Beni validity index (XB), a Dunn validity index (DI), and an ADI validity index.

3. Wavelet Neural Network Time Series Estimation Algorithm

3.1. Wavelet Neural Network (WNN). The WNN combines the wavelet function and the neural network and then transforms the activation function into a wavelet function. Because the wavelet function is able to analyze the signal in both time and frequency domains, it is able to express the complete signal information and determine the local optimal solution; therefore, it has a simpler structure and higher estimation accuracy than a general neural network [26].

The WNN computation process is divided into three steps.

Step 1 (calculate the hidden layer's output).

$$h(j) = h_j \left(\frac{\sum_{i=1}^k \omega_{ij} x_i - b_j}{a_j} \right) \quad j = 1, 2, \dots, l \quad (1)$$

where $h(j)$ is the output value of the j -th node of the hidden layer, ω_{ij} is the connection weight of the input layer and the hidden layer, h_j is the wavelet basis function, b_j is the translation factor for the wavelet basis function, and a_j is the telescopic factor for the wavelet basis function. This paper uses the Mexican Hat (Mexh) function as the basis function, with the Mexh wavelet being the second derivative of the Gaussian function.

The Mexh function expression is

$$y = (1 - x^2) e^{-x^2/2} \quad (2)$$

Step 2 (calculate the output layer's output).

$$y(k) = \sum_{i=1}^l \omega_{ik} h(i) \quad (3)$$

where ω_{ik} is the weight from the hidden layer to the output layer, $h(i)$ is the output of the i -th hidden layer node, l is the number of hidden layer nodes, and m is the number of output layer nodes.

Step 3 (correct the error). In this paper, a gradient modification method is used to modify the weights of the WNN and its basis function parameters. As the gradient learning method efficiency is low, an increasing momentum method is used to improve the efficiency [27]. The correction error is set to e , and the correction processes are described as follows:

(1) Calculate the network estimation errors:

$$e = \sum_{k=1}^m y_n(k) - y(k) \quad (4)$$

where $y_n(k)$ is the desired output and $y(k)$ is the output value predicted by the WNN.

(2) Correction formula of weigh and parameters adds the momentum term:

$$\omega_{n,k}^{(i+1)} = \omega_{n,k}^i + \Delta \omega_{n,k}^{(i+1)} + k \times (\omega_{n,k}^i - \omega_{n,k}^{(i-1)}) \quad (5)$$

$$a_k^{(i+1)} = a_k^i + \Delta a_k^{(i+1)} + k \times (a_k^i - a_k^{(i-1)}) \quad (6)$$

$$b_k^{(i+1)} = b_k^i + \Delta b_k^{(i+1)} + k \times (b_k^i - b_k^{(i-1)}) \quad (7)$$

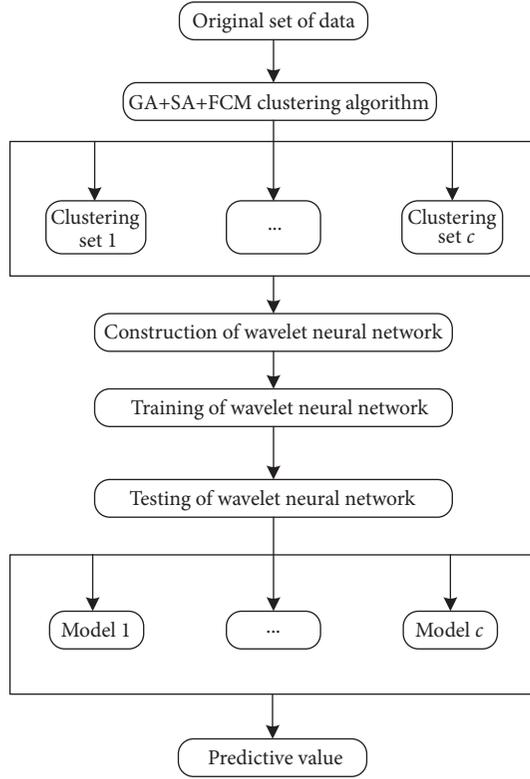


FIGURE 4: Flow of the concrete compressive test data estimation.

where $\Delta\omega_{n,k}^{(i+1)}$, $\Delta a_k^{(i+1)}$ and $\Delta b_k^{(i+1)}$ are calculated based on the network estimation error.

$$\Delta\omega_{n,k}^{(i+1)} = -\eta \frac{\partial e}{\partial \omega_{n,k}^i} \quad (8)$$

$$\Delta a_k^{(i+1)} = -\eta \frac{\partial e}{\partial a_k^i} \quad (9)$$

$$\Delta b_k^{(i+1)} = -\eta \frac{\partial e}{\partial b_k^i} \quad (10)$$

where η is the learning efficiency.

3.2. Build Estimation Model. Based on the clustering results from Section 2.3, WNN is used to construct the test data estimation model for each concrete compressive test data set [28]. The network is divided into three layers: an input layer, a hidden layer, and an output layer [29]. The input for the input layer is the fracture load and compressive strength of the first n times from the test machine, the hidden layer node is composed of the wavelet function, and the output layer is the predictive compressive test data for the current test times. Figure 4 shows the flowchart for the concrete compressive test data estimations.

In this paper, the mean absolute error percentage, e_1 , and the root-mean-square error, e_2 , are selected as the evaluation

index [30] to assess the accuracy of the estimation results, the formula for which is

$$e_1 = \frac{1}{N} \sum_{i=1}^N \frac{|z_i - y_i|}{z_i} \quad (11)$$

$$e_2 = \sqrt{\frac{1}{N} \sum_{i=1}^N (z_i - y_i)^2} \quad (12)$$

where y_i is the estimation data and z_i is the actual measured data.

3.3. Analysis of Restrictive Conditions and Applicability. WNN estimation is not a completely unfamiliar tool, but it is still in the exploratory stage in concrete test data estimation. Wavelet analysis itself has the ability to solve unstable time series and is widely used in data flow prediction with complex characteristic. It is feasible to apply WNN to concrete quality test data estimation.

There are many factors that affect the prediction results of concrete test data, such as the loading rate of the compression-testing machine, the different operator's operations, changes in the compression-testing machine, concrete cube test blocks of different size, the singular test data, and so on. However, for a certain compression-testing machine, using standard concrete cube specimen, the time series of the compression test data has a self-correlation. In this estimation model, all the test data come from the same YAW-2000 compression-testing machine. The equipment is computer-controlled and has a hydraulic synchronous loading device, which eliminates the influence of the change of the different compression-testing machines and the instability of the loading rate on the estimation results. For the estimation error caused by singular test data, the training test data and the actual measured verification data both come from the Shaanxi Xi'an-Hancheng Intercity Railway construction information platform. The information platform system has the function of alerting for the unqualified data and reviewing for suspicious data. The data remaining in the platform system are all reliable.

4. Simulation Analysis

Using concrete compressive test data from the Shaanxi Xi'an-Hancheng Intercity Railway construction information platform, a detailed simulation was conducted. Using one-step-ahead estimation, test data from the YAW-2000 compression-testing machine in the TJ-01 contract section field laboratory were selected for clustering and training. There are three types concrete cube specimen depending on the concrete ratio (which are C20, C30, and C40). The standard concrete specimen is a 150mm * 150mm * 150mm cube. Through the compressive strength test, we got the fracture load, F (KN), and the compressive strength, f_{cu} (MPa), the formula for which is

$$f_{cu} = \frac{F}{A} \quad (13)$$

TABLE 1: Concrete compressive test data sets using as training sample.

Number	Type	F/KN	f_{cu} /MPa	Number	Type	F/KN	f_{cu} /MPa
1	C30	983.25	43.7	2	C30	972.00	43.2
3	C40	1131.75	50.3	4	C40	1125.00	50.0
5	C20	663.75	29.5	6	C40	1127.25	50.1
7	C40	1136.25	50.5	8	C40	1120.50	49.8
9	C30	888.75	39.5	10	C30	904.50	40.2
...
191	C30	893.25	39.7	192	C30	909.00	40.4
193	C40	1131.75	50.3	194	C40	1122.75	49.9
195	C40	1136.25	50.5	196	C40	1138.5	50.6
197	C40	1149.75	51.1	198	C30	981.00	43.6
199	C30	983.25	43.7	200	C40	1120.5	49.8

TABLE 2: Validity measure values for the four algorithms.

Algorithm	PC	PE	SC	S	XB	DI	ADI
FCM	0.6935	0.6096	0.9224	0.0008	21.4251	0.0180	0.0116
GA+FCM	0.7018	0.5976	0.9362	0.0070	3.3545	0.0192	0.0056
SA+FCM	0.7194	0.5700	0.9304	0.0081	10.4561	0.0185	0.0108
GA+SA+FCM	0.7550	0.4588	0.9420	0.0133	3.2631	0.0308	0.0084

where A is the pressure-resistant area of the specimen and $A=150\text{mm}\times 150\text{mm}$.

Take 200 times concrete compressive test data from the YAW-2000 compression-testing machine test, training them as samples to build estimation models. After that, using the data after 200 times tests as the actual measured data samples to compare with the estimate data, the comparison results prove the validity of the estimation model.

The 200 times concrete compressive test data using as training sample are shown in Table 1.

4.1. Clustering Analysis. Cross-validation was used to determine the clustering number for the FCM algorithm, $c=3$, and the GA+SA+FCM clustering algorithm was used to train the samples for the analysis. The concrete compressive test sample clustering results are shown in Figure 5.

To verify the clustering effect of the optimization algorithm, the FCM clustering algorithm, the GA + FCM clustering algorithm, the SA + FCM clustering algorithm, and the GA+SA + FCM clustering algorithm were each used to cluster the samples. The validity measure values for these four methods are shown in Table 2.

In Figure 5, X and Y were the coordinates. The group located in the lower left corner was the C20, the group in the middle was the C30, the group in the rightmost was the C40, and the small red circle in each type shows the cluster center. In Table 2, the GA+SA+FCM algorithm XB index was 3.2631, the DI index was 0.0308, and the ADI index was 0.0084. To compare the simulation results with the other three algorithms, it was found that the clustering effect was closer and the separation ratio was better. The results indicated that the GA+SA+FCM algorithm can be used to cluster the concrete compressive test data.

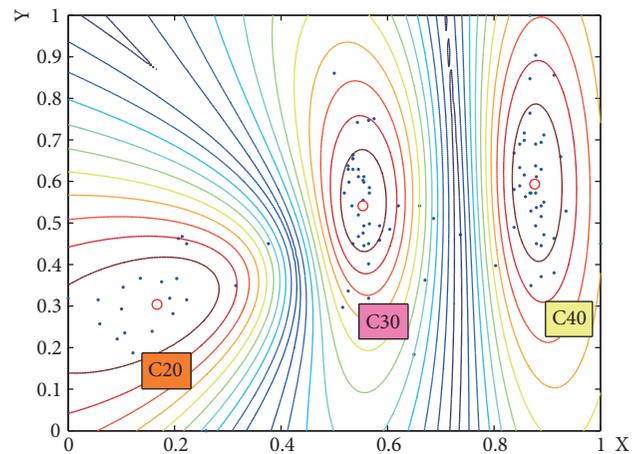


FIGURE 5: GA+SA+FCM concrete compression data clustering results.

4.2. Simulation Analysis. To verify the estimation effects of the WNN, a BP neural network and a WNN were used to estimate the concrete compressive test data under the same experimental conditions. Compared the two estimation data with the actual measured data, the scatter diagram results are shown in Figure 6. The actual measured data samples for fifty C30 concrete cube specimen test data and fifty C40 concrete cube specimen test data were selected from the same YAW-2000 compression-testing machine after the 200 times training sample tests.

A comparison of the measured data with the WNN estimated data indicated that the WNN was able to estimate the fracture load and compressive strength of concrete compressive test data, with the estimated data trends being

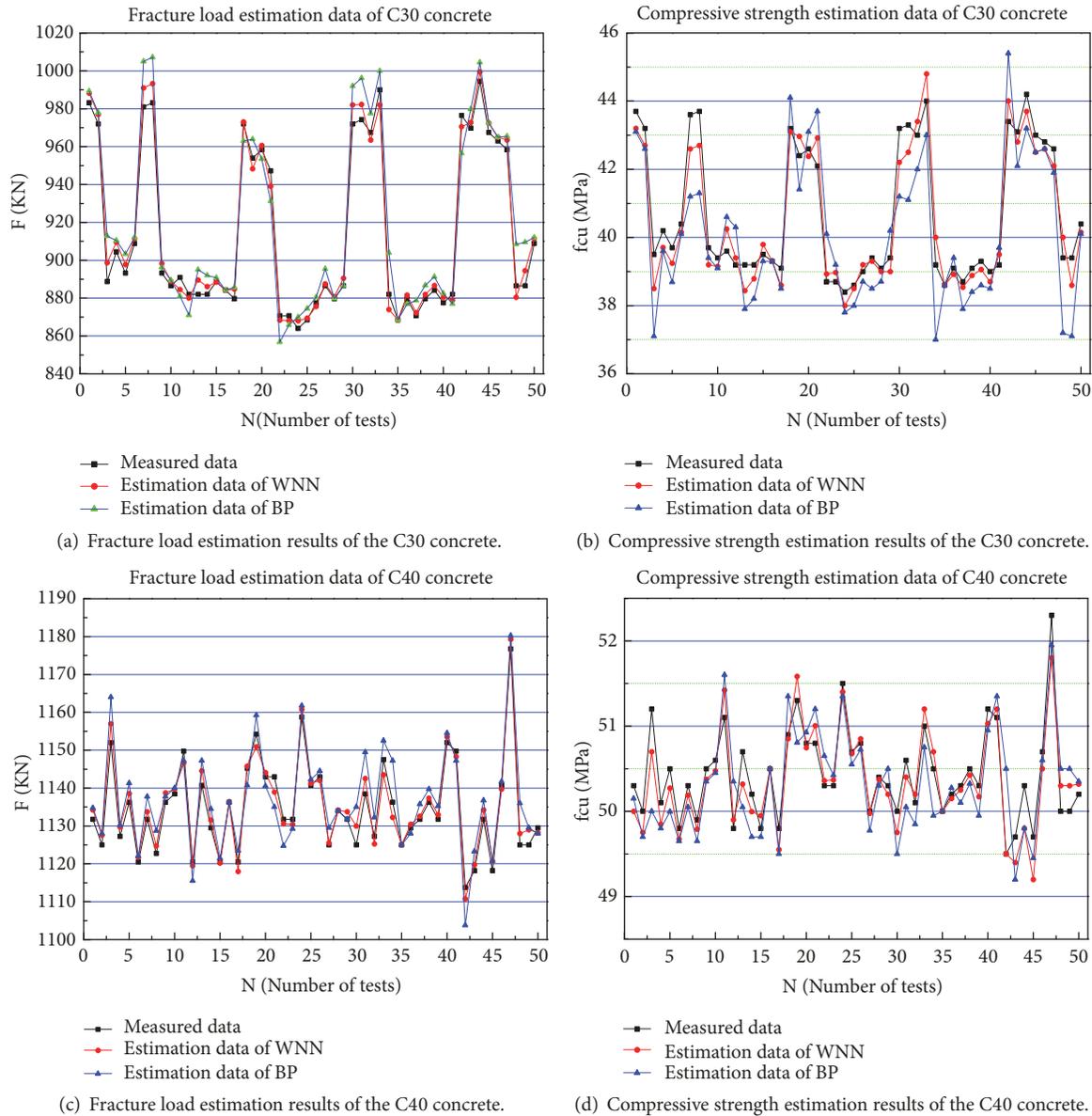


FIGURE 6: Comparison diagrams for the different estimation methods.

basically consistent with the measured data trends. However, if there is a sudden data change, the estimation error obviously increases. From a comparison of the measured data, the WNN estimated data, and the BP neural network estimated data, it was found that the two estimation algorithms were able to accurately estimate the fracture load, F , and compressive strength, f_{cu} , of the concrete compressive test data when the data changed gently and regularly. However, when the data changed abruptly, there were more BP neural network estimation errors than WNN estimation errors at the same position. As shown in Figures 6(a), 6(b), 6(c), and 6(d), at the 6th, 7th, 30th, 32nd, and 47th times in Figures 6(a) and 6(b), and the 3rd, 12th, 43rd, and 44th times in Figures 6(c) and 6(d), the WNN estimation accuracy was obviously higher than the BP neural network, which proved that the

WNN was able to overcome some of the BP neural network shortcomings and was superior to the BP neural network for the estimations. Table 3 shows the evaluation index data for these two estimation algorithms, where e_1 is the mean absolute percentage error and e_2 is the root mean square error. The e_1 of the C30 and C40 concrete fracture load estimated by WNN are 4.7121 and 4.7409 which are far less than the results 9.5870 and 9.3584 estimated by BP. The e_2 of the C30 and C40 concrete fracture load estimated by WNN are 7.2174 and 7.4650 which are far less than the results 11.2873 and 11.4894 estimated by BP. The e_1 and e_2 of the C30 and C40 concrete compressive strength have the same characteristics as the former. The evaluation index data comparison showed that the WNN estimation effect was superior to the BP neural network.

TABLE 3: Evaluation data comparison for the two estimation algorithms.

Estimation algorithms	Fracture load		Compressive strength	
	$e_1/\%$	$e_2/(\text{KN})$	$e_1/\%$	$e_2/(\text{MPa})$
C30 concrete estimation of BP	9.5870	6.8561	11.2873	7.2589
C30 concrete estimation of WNN	4.7121	3.0389	7.2174	3.2447
C40 concrete estimation of BP	9.3584	6.4984	11.4894	7.4780
C40 concrete estimation of WNN	4.7409	3.7480	7.4650	3.4412

5. Conclusions

(1) As the GA+SA optimizing clustering algorithm does not fall into a local optima and has a rapid calculation speed, it was proven to be effective when searching for the global optimal solution and had a significantly better clustering effect than traditional algorithms. It was also proven that the traditional FCM clustering algorithm was strongly depended on the clustering center; the incorrect selection of the clustering center would lead to an incorrect clustering boundary, which would significantly reduce the clustering effect.

(2) The WNN estimation algorithm was found to be able to analyze and estimate the central lab clustered concrete compressive test data, with the estimation results clearly meeting the requirements. In the WNN error correction process, an increasing momentum method was adopted to improve the learning efficiency and to remedy the low efficiency of the traditional gradient modification methods.

(3) The WNN estimation algorithm based on the GA+SA+FCM optimized clustering algorithm proposed in this paper was able to analyze and estimate the concrete compressive test data. The clustering effect was obvious and the estimation accuracy met the requirements. For example, the fracture load estimation for the C30 concrete estimation index e_1 reached 4.7121% and e_2 reached 3.0389KN, and the C40 concrete's e_1 attained 4.7409% and the e_2 was 3.7480KN. The C30 concrete compressive strength estimation e_1 was 7.2174% and e_2 was 3.2447KN, and the C40 concrete compressive strength estimation e_1 was 7.4650% and the e_2 was 3.4412KN. Therefore, as the estimation algorithm was able to predict the concrete quality in advance, any problems associated with the use of unqualified concrete could be avoided.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by Fundamental Research Funds for the Central Universities no. 300102258402, named remote

intelligent monitoring of mechanized construction quality, and the Traffic Research Project of Shaanxi Communications Department in 2017 of Research on Information Technology for Quality and Safety Management of Railway Construction Projects under Grant no. 17-55X.

References

- [1] Y. H. Jia, Z. Q. Wang, and W. W. Li, "Discuss on programs of expressways information management in china," *Journal of Transportation Systems Engineering and Information Technology*, vol. 1, no. 4, pp. 319–322, 2001.
- [2] J. J. Huang, H. P. Pan, and Y. C. Wan, "Application research on the technology of data mining," *Computer Engineering and Applications*, vol. 39, no. 2, pp. 45–48, 2003.
- [3] G. T. Yan, G. F. Ma, and Y. Z. Xiao, "Support vector machines based on hybrid kernel function," *Journal of Harbin Institute of Technology. Harbin Gongye Daxue Xuebao*, vol. 39, no. 11, pp. 1704–1706, 2007.
- [4] C. M. Vong, W. F. Ip, and P. K. Wong, "Variation-oriented data filtering for improvement in model complexity of air pollutant prediction model," *Mathematical Problems in Engineering*, vol. 2014, Article ID 310478, 14 pages, 2014.
- [5] C. Wang, J. Wu, J. Wang, and Z. Hu, "Short-term wind speed forecasting using the data processing approach and the support vector machine model optimized by the improved cuckoo search parameter estimation algorithm," *Mathematical Problems in Engineering*, vol. 2016, Article ID 4896854, 17 pages, 2016.
- [6] W. M. Liu, L. P. Guan, and X. Y. Yin, "Prediction of freeway incident duration based on decision tree," *China Journal of Highway and Transport*, vol. 18, no. 1, pp. 103–107, 2005.
- [7] H. Duan, Z. Deng, F. Deng, and D. Wang, "Assessment of groundwater potential based on multicriteria decision making model and decision tree algorithms," *Mathematical Problems in Engineering*, vol. 2016, Article ID 2064575, 11 pages, 2016.
- [8] H. Yang and S. Fong, "Incremental optimization mechanism for constructing a decision tree in data stream mining," *Mathematical Problems in Engineering*, vol. 2013, Article ID 580397, 14 pages, 2013.
- [9] U. B. Filik, "A new hybrid approach for wind speed prediction using fast block least mean square algorithm and artificial neural network," *Mathematical Problems in Engineering*, vol. 2016, Article ID 8395751, 9 pages, 2016.
- [10] G. F. Fang, W. S. Wang, and C. Liu, "Artificial neural based wind power short term prediction system," *Power System Technology*, vol. 32, no. 22, pp. 72–76, 2008.
- [11] J. H. Xu and T. Q. Liu, "Approach to short-time load forecasting based on wavelet transform and artificial neural network," *Power System Technology*, vol. 28, no. 8, pp. 30–33, 2004.

- [12] G. R. Pan and C. Gu, "Wavelet neural network prediction method of deformation monitoring data," *Journal of Geodesy and Geodynamics*, vol. 27, no. 4, pp. 47–50, 2007.
- [13] J. Zhang, Y. Wu, Q. Liu, F. Gu, X. Mao, and M. Li, "Research on high-precision, low cost piezoresistive MEMS-array pressure transmitters based on genetic wavelet neural networks for meteorological measurements," *Micromachines*, vol. 6, no. 5, pp. 554–573, 2015.
- [14] V. Sharma, D. Yang, W. Walsh, and T. Reindl, "Short term solar irradiance forecasting using a mixed wavelet neural network," *Journal of Renewable Energy*, vol. 90, pp. 481–492, 2016.
- [15] C. Guan, P. B. Luh, L. D. Michel, Y. Wang, and P. B. Friedland, "Very short-term load forecasting: wavelet neural networks with data pre-filtering," *IEEE Transactions on Power Systems*, vol. 28, no. 1, pp. 30–41, 2013.
- [16] Y. Falamarzi, N. Palizdan, Y. F. Huang, and T. S. Lee, "Estimating evapotranspiration from temperature and wind speed data using artificial and wavelet neural networks (WNNs)," *Agricultural Water Management*, vol. 140, no. 3, pp. 26–36, 2014.
- [17] N. S. Mishra, S. Ghosh, and A. Ghosh, "Fuzzy clustering algorithms incorporating local information for change detection in remotely sensed images," *Applied Soft Computing*, vol. 12, no. 8, pp. 2683–2692, 2012.
- [18] W. Ling and D. Zheng, "A kind of GASA hybrid optimization strategy," *Control Theory & Applications*, vol. 18, no. 4, pp. 552–554, 2001.
- [19] G. Zhang, "Improved genetic algorithm for the flexible job-shop scheduling problem," *Journal of Mechanical Engineering*, vol. 45, no. 07, pp. 145–151, 2009.
- [20] S. Li, L. J. Liu, and Y. L. Xie, "Chaotic prediction for short-time traffic flow of optimized BP neural network based on genetic algorithm," *Control and Decision*, vol. 26, no. 10, pp. 1581–1585, 2011.
- [21] W. L. Goffe, G. D. Ferrier, and J. Rogers, "Global optimization of statistical functions with simulated annealing," *Fortran Codes*, vol. 60, no. 1-2, pp. 65–99, 1994.
- [22] X. M. Wang and Y. H. Wang, "The combination of simulated annealing and genetic algorithm," *Chinese Journal of Computers*, vol. 20, no. 4, pp. 381–384, 1997.
- [23] S. W. Fan and X. Z. Zhang, "Study of the global optimal method self-adaptation simulated annealing genetic algorithm," *Optics and Precision Engineering*, vol. 7, no. 4, pp. 16–21, 1999.
- [24] W. Yu, X. Yao, X. Liu, L. Li, and G. Zhai, "Ghost imaging based on Pearson correlation coefficients," *Chinese Physics B*, vol. 24, no. 5, pp. 344–349, 2015.
- [25] L. K. Meng and C. C. Hu, "Cluster validity index based on measure of fuzzy partition," *Computer Engineering*, vol. 33, no. 11, pp. 15–17, 2007.
- [26] Y. Chen, B. Yang, and J. Dong, "Time-series prediction using a local linear wavelet neural network," *Neurocomputing*, vol. 69, no. 4–6, pp. 449–465, 2006.
- [27] G. Ding, X. Hua, M. Tian, and J. Jiang, "Subsidence prediction based on the application of modified wavelet neural network," *Journal of Geomatics*, vol. 38, no. 6, pp. 27–29, 2013.
- [28] R. Gao, C. Shang, and H. Jiang, "A fault detection strategy for wheel flat scars with wavelet neural network and genetic algorithm," *Hsi-An Chiao Tung Ta Hsueh/Journal of Xian Jiaotong University*, vol. 47, no. 9, pp. 88–91, 2013.
- [29] Z. W. Chen and Z. W. Gu, "Simulation realization of prediction model based on wavelet neural network," *Computer Simulation*, vol. 25, no. 6, pp. 147–150, 2008.
- [30] C. Pan, B. Qin, Y. He, C. Yuan, and Q. Shen, "Wind speed forecasting of regularized ELM based on optimized fcm clustering," *Dianwang Jishu/Power System Technology*, vol. 42, no. 3, pp. 842–848, 2018.

