

Retraction

Retracted: Prediction of Urban Scale Expansion Based on Genetic Algorithm Optimized Neural Network Model

Journal of Function Spaces

Received 15 August 2023; Accepted 15 August 2023; Published 16 August 2023

Copyright © 2023 Journal of Function Spaces. 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.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] H. Kuang, "Prediction of Urban Scale Expansion Based on Genetic Algorithm Optimized Neural Network Model," *Journal of Function Spaces*, vol. 2022, Article ID 5407319, 11 pages, 2022.

Research Article

Prediction of Urban Scale Expansion Based on Genetic Algorithm Optimized Neural Network Model

Hewu Kuang ^{1,2}

¹*School of Economics & Management, South China Normal University, Guangzhou 510631, China*

²*School of Insurance, Guangdong University of Finance, Guangzhou 510521, China*

Correspondence should be addressed to Hewu Kuang; 2017010013@m.scnu.edu.cn

Received 30 April 2022; Revised 14 June 2022; Accepted 27 June 2022; Published 14 July 2022

Academic Editor: Miaochoao Chen

Copyright © 2022 Hewu Kuang. 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.

With the continuous development of urbanization, the urban population is becoming more and more dense, and the demand for land is becoming more and more tense. Urban expansion has become an indispensable part of urban development. This paper studies the optimization of neural network structure by genetic algorithm, puts forward the prediction model of urban scale expansion based on a genetic algorithm optimization neural network, and compares the performance of the model with the basic model. A genetic algorithm BP neural network (GA-BP) optimized by the genetic algorithm is used to shorten the running time of the algorithm and improve the prediction accuracy, but it is easy to fall into local solution. The genetic algorithm is improved by immune cloning algorithm, and the CGA-BP neural network model is established to obtain the global optimal solution. Compared with the BP neural network model and GA-BP neural network model, the CGA-BP neural network model converges faster, and the training times reach the error condition after 79 times, while the BP neural network model and GA-BP neural network model need 117 times and 100 times, respectively, and the fitness value corresponding to the number of iterations of the model is larger. Therefore, the CGA-BP neural network algorithm can make prediction more accurately and quickly and predict the expansion of urban scale through urban conditions.

1. Introduction

Urban spatial expansion is the most direct manifestation of urban land use change. Urban spatial expansion is a comprehensive reflection of the changes of spatial layout and structure in the process of urbanization. With the advancement of global urbanization, urban spatial expansion has become a hot field in urban development research at home and abroad. The rapid development of urbanization in China has brought the unprecedented characteristics of urbanization. The continuous expansion and disorderly spread of cities in space have brought a series of problems. The contradiction between supply and demand of urban land is becoming more and more acute, which has become one of the main challenges that China will face now and in the next decade. Under the background that the concept of urban space management has been widely recognized, the rational and orderly expansion of urban space has more strategic significance in China. Therefore, the research on the simulation

and prediction of urban spatial expansion has important and urgent social and economic needs.

With the development of urbanization, urban expansion has become an inevitable trend. Predicting its scale is conducive to the direction of social development and the proportion of economic investment [1]. A neural network model is one of the common prediction means. The neural network has the characteristics of direct input to direct output. It adjusts its parameters to make the output results meet the conditions. Its high accuracy is one of the necessary conditions for prediction [2]. The genetic algorithm is to make up for the defect of neural network falling into local solution. The common genetic algorithms are the niche genetic algorithm and immune clonal genetic algorithm. The niche genetic algorithm is to preserve the population, reduce the fitness of the population, increase the probability of elimination, and increase the diversity of the population. It is a more accurate model [3]. The research is using the genetic algorithm to optimize the model constructed by the neural

network, which is more accurate and scientific for prediction. In order to reduce the redundant connection and unnecessary computational cost of neural network, quantum immune cloning algorithm is applied to the optimization process of neural network. The neural network structure is optimized by generating weights with sparsity. The algorithm can effectively delete the redundant connections and hidden nodes in the neural network and improve the learning efficiency, function approximation accuracy, and generalization ability of the neural network at the same time.

Immune clonal genetic algorithm is used to optimize neural network. Immune clonal genetic algorithm essentially adopts proportional replication operator and proportional mutation operator. Select the antibody with the highest affinity before antigen and antibody for replication, get a new set, and generate the global optimal antibody through high mutation probability. This makes the immune clonal algorithm obtain the optimal solution quickly on the basis of maintaining the diversity of the population, which is the advantage of the model, such as more adaptability and efficiency.

The innovation of this research is to solve the defect that the neural network model is easy to fall into the local optimal solution by combining the characteristics of introducing new species and increasing species and group diversity into the genetic algorithm. The urbanization expansion is gradually mature this year. In this field, the prediction is made by combining the genetic algorithm and neural network for the first time, and the nonlinear function characteristics of the neural network are used to make the prediction of urban expansion more accurate and provide a scientific basis for urban development.

The research is mainly divided into four parts. Section 2 mainly analyzes and summarizes the current application and effect of the genetic algorithm and neural network. Section 3 introduces the factors affecting the development of urban scale and constructs the prediction model of the genetic algorithm optimization neural network. Section 4 analyzes and compares the new capabilities of the optimization model and the traditional model. Section 5 evaluates the analysis and research results and puts forward the deficiencies in the research.

2. Related Work

Dawid's team applied the genetic algorithm to the spider web model and explained the results of different settings by simulating different coding schemes and fitness functions with state dependence. It found the relationship between the coding and convergence attributes of the genetic algorithm, the adaptability of the genetic algorithm in the economic system and the learning behavior of bounded rational groups [4]. Researchers such as Domashova et al. proposed a genetic algorithm to build the best framework of multilayer perceptron, create a random population, confirm the selection method of fitness function and parent sample selection, and propose the modification of crossover and mutation operators to ensure the operability of the algorithm for large and small individuals and solve the classification problem with minimal error [5]. Shukla and other scholars use the

model established by the convolutional neural network to diagnose COVID-19 patients with chest X-ray images and use the multiobjective genetic algorithm to adjust the recognition parameters. The experiment shows that 20x cross-validation can overcome the overfitting dilemma, and the recognition of the COVID-19 model has achieved obvious results, providing technical support for the process of diagnosis and screening of diseases [6]. Feng and other researchers proposed a network structure based on genetic algorithm, which generates a basic model dominated by an integrated algorithm by adjusting parameters and takes the original personal identification data as the numerical value. The results show that the prediction accuracy of the integrated model is better than the basic model, making personal identification occupy an important position in biological feature recognition [7]. Nikbakht's team applied the genetic algorithm to the neural network to find the optimal solution and optimized the hidden layer, integration point, and neuron number of each layer to achieve the highest accuracy, so as to predict the stress distribution of the structure. The results show that the prediction accuracy of the neural network has been significantly improved after the optimization of the genetic algorithm [8]. Han and other researchers used the genetic algorithm to optimize the BP neural network and build a GA-BP hybrid model. 13 of the 16 selected product design schemes were added to the hybrid model as parameters. After training, the predicted and actual values were obtained. Finally, the remaining schemes were verified. The results showed that the errors of several groups of data were 3.4%, 9%, and 3.1%, reflecting that the hybrid GA-BP model can predict the design scheme quickly, conveniently and accurately [9]. Ni's team proposed a router optimization algorithm based on the genetic ant colony algorithm in an IPv6 environment. The algorithm integrates genetic algorithm and ant colony algorithm, rewards or punishes by comparing the smoothness of the search path and the best path, and trains the obtained optimal solution. Experimental results show that the algorithm effectively solves the network quality problem and improves the service quality to users [10].

Researchers such as Gamidi and Ke proposed an artificial neural network model to predict the properties of eutectic based only on pure component physical properties. In the model grouping, the proportion of training model and test model is 7:3. When the training stops, the total value is higher than 0.986. The corresponding dissolution properties of eutectic and ideal solubility of eutectic are successfully predicted [11]. Zhang and other scholars combined PM2.5. The prediction model based on the Bi-LSTM neural network is constructed based on the concentration and meteorological information data. The meteorological characteristics are introduced and compared with the neural network model based on Gru. The results show that the model is effective for PM2.5. The prediction error of the concentration is smaller [12]. Pattnaik's team developed a prediction model based on the artificial neural network and Taguchi to predict the response. By using a lower level Taguchi orthogonal array to conduct a small number of experiments in the process of wire laying out, the results show that the modified model can predict the response more accurately [13].

Kenanoğlu and other researchers established an artificial neural network model by using LM learning algorithm to adjust the weight in the forward network. The results show that the model has high-precision prediction of motor torque, motor power, and emissions and plays an important role in promoting low-carbon emissions [14]. Abuelezz's team proposed a model for predicting the vertical total electron content of the upper ionosphere based on neural network and compared it with the IRI2016 model. The results show that the IRI2016 model is inferior to this model in all cases [15]. Chen and other researchers established a film scoring prediction model based on the convolutional neural network. Through experiments, it is found that the performance of a ten-layer convolutional neural network is the best, and the accuracy is about 56%, and the predicted value is close to the actual value, which provides a new model for film scoring prediction [16]. Gade's team proposed a Newmark sliding displacement prediction model based on the neural network. The mixed effect algorithm was used to evaluate the event residual. It was found that the model had no deviation and was consistent with the observed displacement characteristics, proving the applicability of the model and effectively predicting the seismic slope displacement hazard [17]. Ma and other researchers put forward a network state prediction algorithm of intelligent production line, which calculates and predicts the operation of the network through the optimized BP neural network. The results show that the optimal data prediction cycle is obtained in a large number of network data prediction experiments, and the accuracy is more than 90%, which is of great significance to improve the quality of network communication [18].

Based on the above analysis, genetic algorithm and neural network structure play an important role in classification, screening, and prediction, improve the accuracy of results, and promote the development of various fields. However, the relevant algorithms are rarely used in urban scale expansion, and the model provides a new scientific basis for urban scale expansion prediction.

3. Application of CGA-BP Neural Network in Urban Scale Expansion Model

3.1. Prediction of Urban Scale Expansion and Construction of Neural Network. Urban construction land is directly related to the rational layout and optimal utilization of urbanization land and has an important impact on the competitiveness and development of the city. With the rapid development of social economy and the improvement of urbanization level, the demand for urban construction land will continue to increase. The acceleration of urban construction land expansion has become an inevitable trend. This paper plans and optimizes the layout to improve the utilization efficiency and rational utilization of urban construction land.

The expansion of urban scale is to meet the balance of supply and demand. Therefore, it is necessary to meet the rational layout and maximize the benefits. The feasibility of urban scale expansion should consider the following factors: demographic factors, economic factors, location factors, social factors, policy factors, etc. [19]. The traditional predic-

tion methods include the linear regression method and gradient descent method. The linear regression method takes the factors affecting the price as variable parameters, which are expressed by $x = [x_1, x_2, \dots, x_m]^T$ (T represents matrix transposition), d represents the sum of output variables, w_1, w_2, \dots, w_m represents unknown parameters, and the expected error of the model is expressed by ε , so the linear regression equation shown in formula (1) can be established.

$$d = \sum_{j=1}^m w_j x_j + \varepsilon. \quad (1)$$

The gradient descent method randomly initializes the factors affecting house prices to achieve the purpose of optimizing the function and obtains the optimized linear regression model, as shown in

$$h(\theta) = \sum_{j=0}^n \theta_j x_j. \quad (2)$$

In formula (2), m represents the number of training set samples, n represents the number of features, $h(x)$ represents the fitting function, $J(\theta)$ represents the loss function, and the loss function formula is shown in

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^m (y^i - h_{\theta}(x^i))^2. \quad (3)$$

Random gradient descent method and batch gradient descent method are commonly used gradient descent methods. In the random gradient descent method, the partial derivative of the error function $J(\theta)$ is shown in

$$\frac{\partial J(\theta)}{\partial \theta_j} = -\frac{1}{m} \sum_{i=1}^m (y^i - h_{\theta}(x^i)) x_j^i \quad (4)$$

After the update of each θ , the function of risk minimization is obtained, as shown in

$$\theta_j^i = \theta_j + \frac{1}{m} \sum_{i=1}^m (y^i - h_{\theta}(x^i)) x_j^i. \quad (5)$$

It can be seen from equations (4) and (5) that when the value of m is particularly large, the process of fitting the function will consume more time. In the batch gradient descent method, all sample sizes are trained and the partial derivatives of parameters are obtained. The calculation formula is shown in

$$\theta_j^i = \theta_j + (y^i - h_{\theta}(x^i)) x_j^i \quad (6)$$

After eliminating the characteristic function, the batch gradient descent method can obtain the optimal solution of more data with fewer samples and determine which gradient descent method to choose according to the actual demand (brindle k m et al. 2021) [20]. During data analysis, some

original data have problems such as duplication or deletion. Therefore, the original data needs to be preprocessed so that the data can be applied to the model. The data processing method is to remove the uniqueness. The data that does not belong to the characteristics of the data sample and has no impact on the modeling can be deleted directly. The missing data can be supplemented or deleted by the average. The data often has orders of magnitude differences, so the data of different orders of magnitude and different quantity levels are standardized and regularized to ensure the experimental reliability. The common ones are min max standardization and Z-score standardization [21].

Min max standardization is to map the value x of attribute A into the $[0, 1]$ interval through standardization, and the formula is shown in

$$x' = \frac{x - \min x}{\max x - \min x}. \quad (7)$$

Z-score standardization is to regularize the average and standard deviation of the original data x . The obtained formula is shown in

$$x' = \frac{x - x_{\text{mean}}}{x_{\text{std}}}. \quad (8)$$

In equation (8), mean represents the average value and std represents the standard deviation. Set the data set as $D = \{(\vec{x}_1, y_1), (\vec{x}_2, y_2), \dots, (\vec{x}_N, y_N)\}$, $\vec{x}_i = (x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(d)})^T$ and calculate the norm L_p as shown in

$$L_p(\vec{x}_i) = \left(\|x_i^{(1)}\|^p + \|x_i^{(2)}\|^p + \dots + \|x_i^{(d)}\|^p \right)^{1/p}. \quad (9)$$

The result of regularization is shown in

$$\vec{x}_i = \left(\frac{x_i^{(1)}}{L_p(\vec{x}_i)}, \frac{x_i^{(2)}}{L_p(\vec{x}_i)}, \dots, \frac{x_i^{(d)}}{L_p(\vec{x}_i)} \right)^T. \quad (10)$$

The specific process of BP neural network model is shown in Figure 1.

3.2. Neural Network Optimized by Immune Clonal Genetic Algorithm. The good genes suitable for the environment will be selected by the good genes suitable to survive, and the unsuitable genes will be gradually eliminated. This natural law idea of survival of the fittest is applied to the process of finding the optimal solution to obtain the genetic algorithm (GA) [22]. Genetic algorithm can adjust the search algorithm through its own adaptation, without formulating certain rules. It can also find the optimal solution with probability when the objective function is not clear and can process multiple sample data at the same time, so as to improve the efficiency of the algorithm without avoiding the local optimal solution [23].

At the initial stage of the algorithm, data samples will be randomly generated. The samples have the attributes to solve the problem. After imitating the way of biological

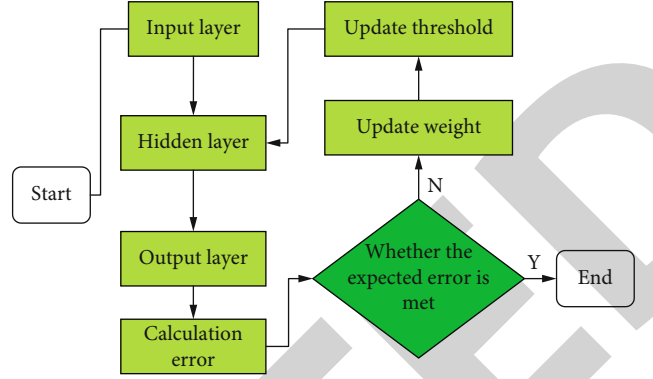


FIGURE 1: BP neural network algorithm flow.

selection, cross-reproduction, and obtaining mutated genes, the algorithm will generate a sample set more suitable for solving the problem and finally obtain the optimal solution through repetition. Then, the algorithm is accelerated by adjusting the parameters to obtain the optimal solution without falling into the local optimal solution [24]. The main steps are to code with specified numbers, letters, and other methods, generate random initial samples, evaluate the fitness in the initial sample individuals, and judge whether the operation meets the termination conditions through selection, crossover, and mutation. The flow chart is shown in Figure 2.

The chromosome coding of genetic algorithm is very important to the evolutionary efficiency. The commonly used coding includes binary coding, gray coding, and floating point coding. Although binary coding has many processing features and high accuracy in genetic algorithm, it is not universal. The commonly used coding is floating-point coding. Floating-point coding uses real numerical values to realize algorithm operation, which is direct, simple, and widely used. The fitness function of genetic algorithm is used to evaluate the fitness value of sample individuals. The design of fitness function needs to avoid the problems that large individual differences in the early stage of evolution affect the search ability and small differences in the later stage of evolution weaken the performance of the algorithm. The selection of fitness function directly affects the convergence speed of genetic algorithm and whether it can find the optimal solution. Because genetic algorithm basically does not use external information in evolutionary search, it only uses the fitness of each individual of the population to search based on the fitness function. Because the complexity of fitness function is the main component of the complexity of genetic algorithm, the design of fitness function should be as simple as possible to minimize the time complexity of calculation. The transformation method of fitness function value is shown in

$$\text{Fit}(f(x)) = \frac{1}{1 + c + f(x)}, \quad c \geq 0, \quad c + f(x) \geq 0, \quad (11)$$

where c represents the bound estimation of $f(x)$, and equation (11) is used to find the minimum value problem.

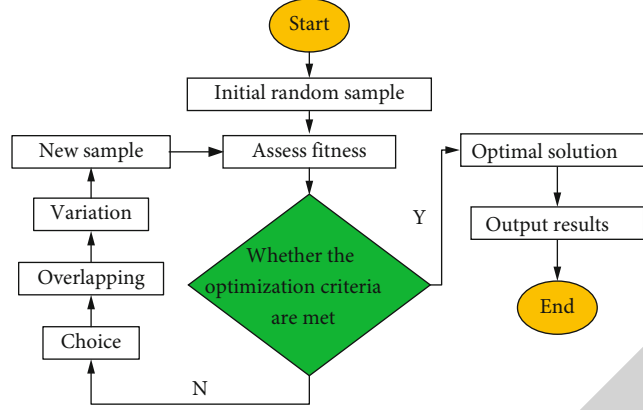


FIGURE 2: Chart of genetic algorithm.

Similarly, the problem of finding the maximum value can be obtained, and the function is shown in

$$\text{Fit}(f(x)) = \frac{1}{1 + c - f(x)}, c \geq 0, c - f(x) \geq 0. \quad (12)$$

Genetic operation includes the process of selection, crossover, and mutation. Individuals with problem-solving attributes are selected from large samples to form new samples. The quality of individuals directly affects the results of the algorithm. In order to meet the randomness of sampling, the sampling mode of roulette can be selected. The probability formula of Roulette is shown in equation (13). Genetic operation includes the process of selection, crossover, and mutation. Individuals with problem-solving attributes are selected from large samples to form new samples. The quality of individuals directly affects the results of the algorithm. In order to meet the randomness of sampling, the sampling mode of roulette can be selected. The probability formula of roulette is shown in

$$p_k = \frac{f(x_k)}{F} = \frac{f(x_k)}{\sum_{k=1}^n f(x_k)}, k = 1, 2, \dots, n., \quad (13)$$

where p_k represents the probability of being selected, x_k represents any individual, and $f(x_k)$ represents the fitness value of any individual. After the selection operation is completed through the above function, the crossover operation is carried out. First, the parent is selected. The parent is a combination obtained by random pairing of two individuals, and the individual selection is random sampling from the sample with the highest fitness value. The individual attributes are exchanged through one-point crossing and multipoint crossing to obtain the offspring. The one-point crossing diagram is shown in Figure 3.

Randomly select a point on the chromosome, divide the chromosome into two parts, and exchange the parts before or after the intersection to obtain new offspring. This process is cross-operation [25]. Now, swap the 5678 alleles of the parent in Figure 2, and the chromosome number is encoded in binary. Then, the fitness values f of the parent before

crossover are $(2^4 + 2^2 + 2^0)^2 = 441$ and $(2^7 + 2^4 + 2^1)^2 = 21316$, respectively. After exchange, the fitness values of the offspring are $(2^4 + 2^1)^2 = 324$ and $(2^7 + 2^4 + 2^2 + 2^0)^2 = 22201$. The results show that the fitness value of the offspring may evolve more than that of parents through the recombination of parents. Mutation operation is based on the theory of small probability. Some genes of chromosome mutate to form new individuals. Binary coding is to change "1" into "0" or "0" into "1." When such mutation operator is used in GA, it can not only prevent the immature phenomenon of the algorithm in the training process but also achieve the effect of optimizing GA. The parameters of the genetic algorithm depend on the population size M , and according to the actual situation, the value range is 10 – 200, the crossover probability P_c and the general value range of crossover probability 0.4 – 0.99, and the mutation probability P_m and the general value range 0.0001 – 0.1.

Due to the great limitations of the BP neural network, the convergence speed of function is slow and the accuracy of prediction value is relatively low. Using Hamming distance method to measure the common points or similarities between different individuals, the penalty function is applied to increase the elimination probability of individuals with low fitness and maintain individuals with high fitness. This method is niche technology. Adjusting the fitness of all individuals is to use the mechanism of sharing function to express the sharing degree with the sum of the sharing values among all individuals, as shown in

$$S_i = \sum_{j=1}^n S(d_{ij}). \quad (14)$$

The sharing value of individual i and individual j is represented by d_{ij} , S represents the sharing function, and the sum of sharing degrees of all individuals is represented by S_i . The adjustment formula of individual fitness is shown in

$$\text{Fit}(f_s(i)) = \frac{f(i)}{S_i}. \quad (15)$$

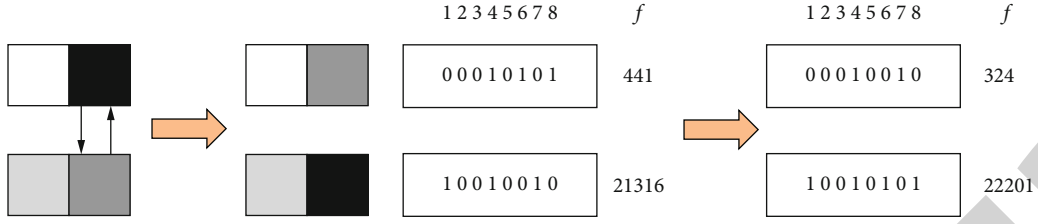


FIGURE 3: One-point intersection example.

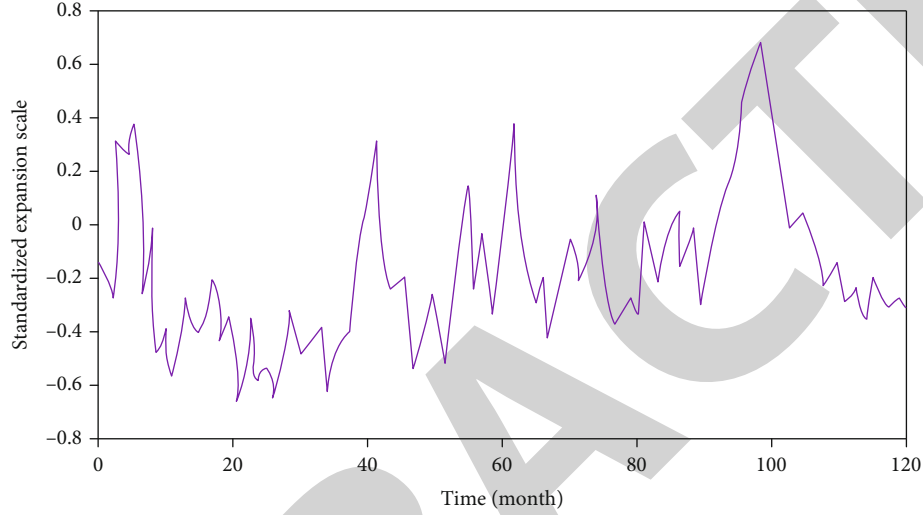


FIGURE 4: Normalized urban expansion curve.

The step of immune cloning algorithm is to randomly generate a set H , which contains antibodies, and the size of the antibody is expressed in N . Then, the affinity between each antibody in the set is calculated, and the n antibodies with the highest affinity value are reconstituted into a new set. Copy the antibodies in the new set. Each antibody copies a certain number to temporarily form a clone set. The clonable set is given a certain mutation probability, and the mutated set is represented by C_n . Calculate the antibody affinity in the mutated set and then select the first few antibodies with the highest affinity value. These antibodies form a memory cell set and replace the antibodies with low affinity value in the initial set H , so as to maintain the diversity of antibody population.

TABLE 1: Contribution rate of different dimensions.

Dimension	Contribution rate (%)	Dimension	Contribution rate (%)
1	47.1767	8	93.0468
2	58.2130	9	95.1792
3	67.7811	10	96.8751
4	74.3844	11	97.5912
5	80.8125	12	98.3073
6	85.8746	13	99.6110
7	89.9968	14	100.0000

4. Experimental Design and Analysis

Too much sample data selection will lead to slow operation speed. On the contrary, too little sample data will affect the accuracy of prediction. Here, 500 groups of data samples are selected, of which 400 groups are randomly selected as input data, and the remaining 100 groups are used as test samples to test the effect of the model through error comparison. The columns contained in a dimension that represent the dimension are called dimension attributes. Get dimension properties from the report. Or talk to business personnel to get. When the attribute hierarchy is instantiated as a series of dimensions instead of a single dimension, it is called

a snowflake pattern. The operation of merging the attribute hierarchy of a dimension into a single dimension is called denormalization. Using snowflake mode, users need a lot of correlation operations in the process of statistical analysis, which has high complexity and poor query performance. It is convenient, easy to use, and has good performance. Due to different dimensions between data sources, effective comparison cannot be carried out. It needs to be normalized to convert dimensional data into dimensionless data and map the data of statistical probability between $[0, 1]$, and the mapping of data is in the range of $[-1, 1]$. The normalization formula of the original data through the premnmx function is shown in

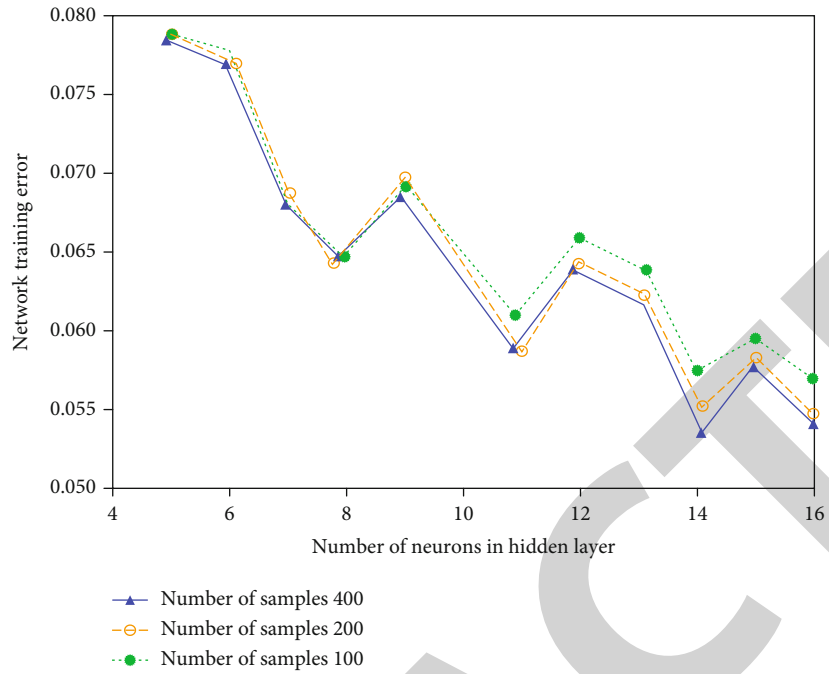


FIGURE 5: Network training error corresponding to the number of hidden layers.

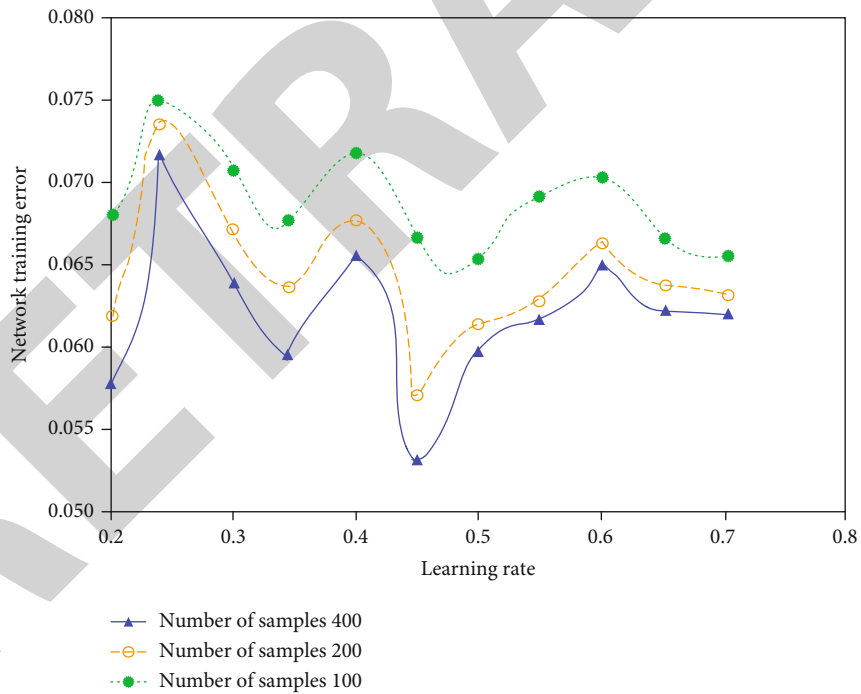


FIGURE 6: Network training error corresponding to different learning rates.

$$\begin{aligned}
 \text{PN} &= \frac{2(p - \min p)}{\max p - \min p} - 1, \\
 \text{TN} &= \frac{2(t - \min t)}{\max t - \min t} - 1.
 \end{aligned}
 \tag{16}$$

In equation (16), p is the input sample of the original data, t is the output sample of the original data, $\min p$ and

$\min t$ represent the minimum value of the original input sample and output sample, respectively, and $\max p$ and $\max t$ represent the maximum value of the original input sample and output sample, respectively. PN and TN obtained by normalization represent the factors affecting the expansion of urban scale and the value of expansion scale, respectively. The data obtained after fitting by the BP neural network is still the normalized data, and the inverse

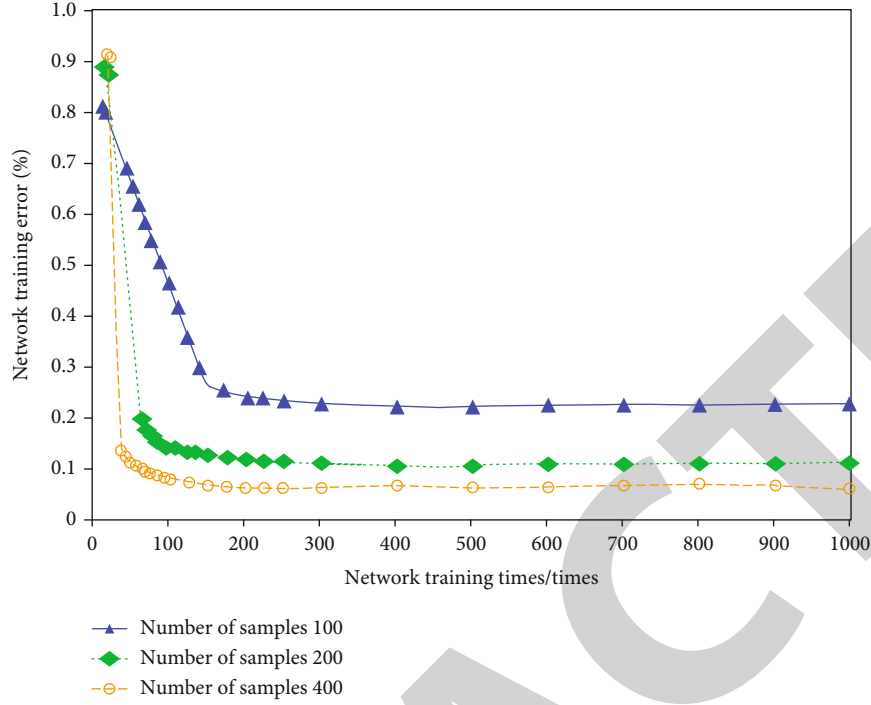


FIGURE 7: Network training error corresponding to different training times.

normalization function is also required to obtain the normal value. After normalizing the original data output samples, the numerical diagram is obtained, as shown in Figure 4.

The dimensionality of data samples is different and needs to be reduced. PCA dimensionality reduction method can be used. Based on linear transformation, the high-dimensional data is projected into the low-level space, that is, the dimensions with a large amount of information in the data are saved and the unimportant information is deleted, which can effectively reduce the running time of the algorithm. Firstly, the sample features are averaged, that is, the difference between the sample features and the feature mean of the rigid sample, the eigenvalues and eigenvectors in the covariance matrix of the sample are calculated, and the eigenvectors corresponding to the previous maximum eigenvalues are selected and recombined into a new matrix to obtain a new sample set. According to the experimental data, the contribution rates corresponding to different dimensions are obtained according to PCA dimensionality reduction, as shown in Table 1.

According to the dimension reduction experiment, the convergence curve of contribution rate after 9-dimensional class is gradually slow, so 9-dimensional data are selected for experimental simulation. In the construction of the BP neural network model, a three-layer network structure is selected to meet the requirements of the operation rate. Nine-dimensional data is used as the number of neurons in the input layer of neural network, and the number of neurons in the output layer is the result of urban expansion prediction. The number of neurons in the hidden layer needs to be determined by trial and error. Too few or too many neurons in the hidden layer will affect the output error. There-

TABLE 2: Algorithm and simulation time corresponding to different cross-probabilities.

Crossover probability	Algorithm error	Simulation time (seconds)
0.4040	0.0438	0.2679
0.4888	0.0396	0.2679
0.5737	0.0815	0.1733
0.6585	0.0374	0.2836
0.7434	0.0754	0.1576
0.8282	0.0840	0.1733
0.9130	0.0281	0.6618
0.9979	0.0399	0.8193

fore, the number of neurons in the hidden layer is determined to be 5-16 according to the common empirical formula. When the number of sample sets is different and other conditions are the same, the comparison of network errors corresponding to different numbers of nodes in the hidden layer is shown in Figure 5.

According to the simulation results in the figure, when the training error reaches the minimum value of 0.053645, the number of neurons in the hidden layer is 14, so the number of neuron nodes is 14. To sum up, the number of neurons in the input layer of the experimental model is 9, the number of neurons in the hidden layer is 14, and the number of neurons in the output layer is 1.

If the learning rate of BP neural network is set too high, it may lead to the error of optimal solution. If the learning rate is too small, it will lead to the long running time of the algorithm. The appropriate learning rate can make the

TABLE 3: Parameters of neural network.

Parameter name	Parameter value	Parameter name	Parameter value
Number of network layers	3	Training times	150
Enter the number of layers	9	Population size	50
Number of hidden layers	14	Number of iterations	80
Number of output layers	1	Crossover probability	0.652
Learning rate	0.45	Variation probability	0.01

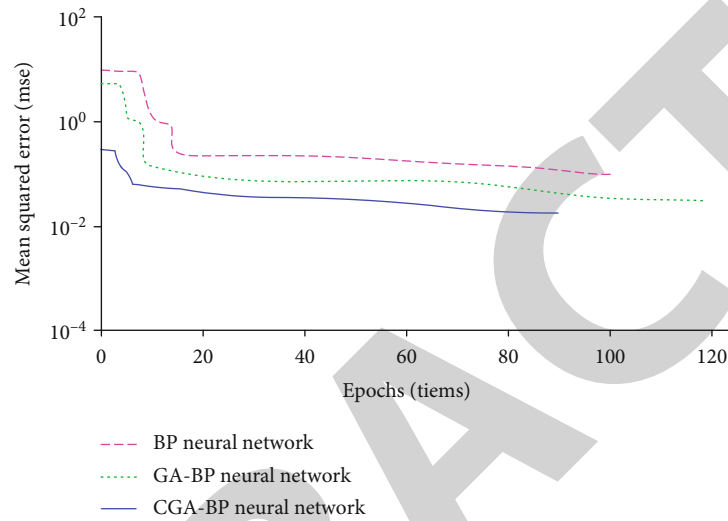


FIGURE 8: Comparison of BP neural network training under different models.

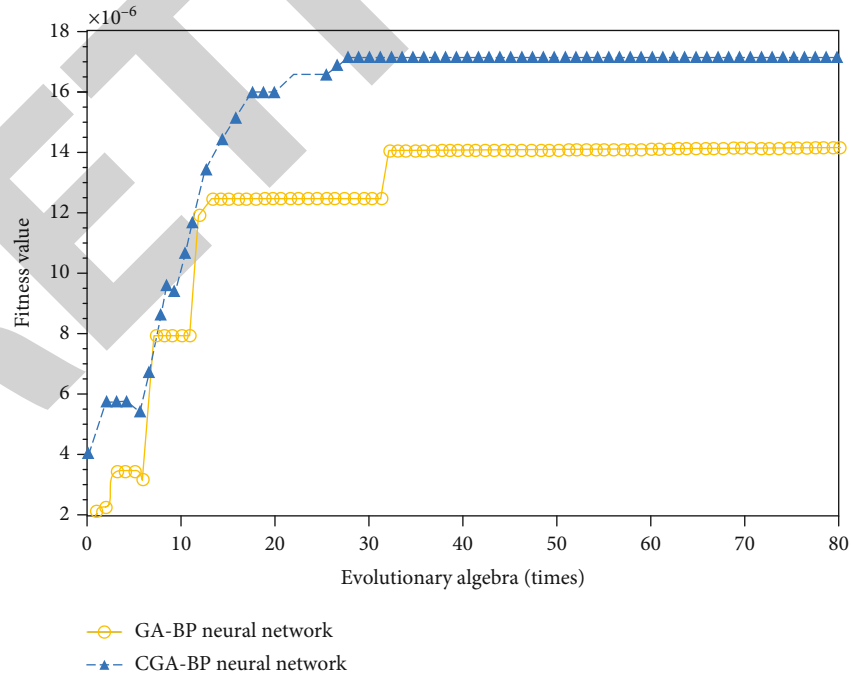


FIGURE 9: Comparison of iteration times of genetic algorithm under different algorithms.

algorithm have better operation performance. Set the learning rate to 0.2 to 0.7. The results are shown in Figure 6.

Different training times also have a certain impact on the prediction accuracy. Therefore, gradually increase the training times based on the determination of the neural network framework structure and analyze its correlation with the accuracy prediction. The results are shown in Figure 7. The population size can be determined by the optimized performance of the algorithm. After comparing the prediction accuracy and simulation time, it can be determined that the population size is the best of 50.

The convergence accuracy of the genetic algorithm will be affected by the number of iterations. It is necessary to find a balance between efficiency and accuracy when the algorithm converges. When the crossover probability is large, the offspring will produce unstable changes and destroy the excellent model of the population. On the contrary, when the crossover probability is small, the evolution process of the algorithm will slow down and spend a lot of time in the genetic search stage. Therefore, the crossover probability is generally limited to 0.4-0.99. The error size and simulation time results corresponding to different crossover probabilities are shown in Table 2. When the algorithm error is small and the simulation time is appropriate, the crossover probability is 0.652.

Because the mutation probability will affect the generation of new individuals, the mutation probability is too high, and the generation probability of new individuals will also increase, which may also destroy the existing excellent pattern structure. On the contrary, the mutation probability is too small, there is no change of new individuals, and it is not conducive to the convergence of the algorithm. Therefore, the range of mutation probability is limited to 0.0001-0.1, and the initial value is 0.01. Through the above experiments, the parameters of the prediction model for predicting urban scale expansion are obtained, which are summarized as shown in Table 3.

After the model parameters are determined, the performance of the model is reflected through the model results. The relationship between the training times under different models and the target error is shown in Figure 8.

The mean square error levels of the three neural networks can be seen from Figure 8. Among them, the mean square error level of the BP neural network is the highest, and GA-BP neural network. The lowest error is the CGA-BP neural network. The difference between the estimated quantity and the estimated quantity is the smallest. After training the GA-BP neural network and CGA-BP neural network model, the relationship between iteration times and fitness value is shown in Figure 9.

5. Conclusion

For urban development, a prediction model of urban scale expansion based on the BP neural network optimized by Xiaosheng genetic algorithm is proposed and compared with other models. The results show that the neural network parameters have suitable values, the number of model dimensions is about 9, the contribution rate of the model

is more than 95% and tends to be stable, the network training error is the smallest, the number of hidden layer nodes is 14, and the model learning rate is 0.45. The number of data samples will also affect the network training error, and with the increase of training times, the error change is no longer significant. When the number of samples is 400, the number of training times is 50, and the error tends to be flat. Considering the optimization accuracy and simulation time, the population size is the best value of 50. When the number of iterations of the model is 150, the genetic algorithm begins to converge and the running time of the algorithm is less. The crossover probability can be seen from Table 2. When the crossover probability is 0.652, the error is small on the basis of moderate simulation time, and the mutation probability can neither be large nor small, so it is set to 0.01. When the parameter conditions are the same, the BP neural network model, GA-BP neural network model, and CGA-BP neural network model are compared. The CGA-BP neural network model achieves the target error when the number of training is 79, which is better than the other two models. The fitness value corresponding to the number of iterations also shows that the best fitness value of the CGA-BP neural network when the number of iterations is the least is also higher than other models. Therefore, the neural network model optimized by genetic algorithm is more suitable for the prediction of urban scale expansion. However, there are still some limitations in this paper. Limited by various conditions, the research still has deficiencies, and the parameter setting is not comprehensive enough. Genetic algorithm is insufficient to optimize the neural network structure model. In the follow-up work, it is necessary to further improve the quality of the research model and obtain the prediction results more accurately.

Data Availability

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

Conflicts of Interest

The author declares that he/she has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] B. Alharthi and T. A. El-Damaty, "Study the urban expansion of Taif City using remote sensing and GIS techniques for decision support system," *Advances in Remote Sensing*, vol. 11, no. 1, pp. 1-15, 2022.
- [2] Y. Wang, L. Wang, F. Yang, W. di, and Q. Chang, "Advantages of direct input-to-output connections in neural networks: the Elman network for stock index forecasting," *Information Sciences*, vol. 547, pp. 1066-1079, 2021.
- [3] F. I. Chou, W. H. Ho, and C. H. Chen, "Niche genetic algorithm for solving multiplicity problems in genetic association studies," *Intelligent Automation and Soft Computing*, vol. 26, no. 3, pp. 501-512, 2020.

- [4] H. Dawid and M. Kopel, "On economic applications of the genetic algorithm: a model of the cobweb type," *Journal of Evolutionary Economics*, vol. 8, no. 3, pp. 297–315, 1998.
- [5] J. V. Domashova, S. S. Emtseva, V. S. Fail, and A. S. Gridin, "Selecting an optimal architecture of neural network using genetic algorithm," *Procedia Computer Science*, vol. 190, no. 14, pp. 263–273, 2021.
- [6] P. K. Shukla, J. K. Sandhu, A. Ahirwar, D. Ghai, P. Maheshwary, and P. K. Shukla, "Multiobjective genetic algorithm and convolutional neural network based COVID-19 identification in chest X-ray images," *Mathematical Problems in Engineering*, vol. 2021, Article ID 804540, 9 pages, 2021.
- [7] X. Feng, J. Zhao, and E. Kita, "Genetic algorithm-based optimization of deep neural network ensemble," *The Review of Socionetwork Strategies*, vol. 15, no. 1, pp. 27–47, 2021.
- [8] S. Nikbakht, C. Animescu, and T. Rabczuk, "Optimizing the neural network hyperparameters utilizing genetic algorithm," *Journal of Zhejiang University-SCIENCE A*, vol. 22, no. 6, pp. 407–426, 2021.
- [9] J. X. Han, M. Y. Ma, and K. Wang, "Product modeling design based on genetic algorithm and BP neural network," *Neural Computing and Applications*, vol. 33, no. 9, pp. 4111–4117, 2021.
- [10] W. Ni, Z. Xu, J. Zou, Z. Wan, and X. Zhao, "Neural network optimal routing algorithm based on genetic ant colony in IPv6 environment," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 3115704, 2021.
- [11] R. K. Gamidi and C. R. Ke, "Analysis and artificial neural network prediction of melting properties and ideal mole fraction solubility of cocrystals," *Crystal Growth and Design*, vol. 20, no. 9, pp. 5745–5759, 2020.
- [12] M. Zhang, D. Wu, and R. Xue, "Hourly prediction of PM2.5 concentration in Beijing based on Bi-LSTM neural network," *Multimedia Tools and Applications*, vol. 80, no. 16, pp. 24455–24468, 2021.
- [13] S. Pattnaik and M. K. Sutar, "Advanced Taguchi-neural network prediction model for wire electrical discharge machining process," *Process Integration and Optimization for Sustainability*, vol. 5, no. 1, pp. 159–172, 2021.
- [14] R. Kenanoğlu, M. K. Baltacıoğlu, M. H. Demir, and M. E. Özdemir, "Performance & emission analysis of HHO enriched dual-fuelled diesel engine with artificial neural network prediction approaches," *International Journal of Hydrogen Energy*, vol. 45, no. 49, pp. 26357–26369, 2020.
- [15] O. A. Abuelezz, A. M. Mahrous, P. J. Cilliers et al., "Neural network prediction of the topside electron content over the Euro-African sector derived from Swarm-A measurements," *Advances in Space Research*, vol. 67, no. 4, pp. 1191–1209, 2021.
- [16] Y. Chen, L. Guo, and C. Zhang, "Score prediction model based on neural network," *Optical Memory and Neural Networks*, vol. 29, no. 1, pp. 37–43, 2020.
- [17] M. Gade, P. S. Nayek, and J. Dhanya, "A new neural network-based prediction model for Newmark's sliding displacements," *Bulletin of Engineering Geology and the Environment*, vol. 80, no. 1, pp. 385–397, 2021.
- [18] Y. Ma, L. Li, Z. Yin, A. Chai, M. Li, and Z. Bi, "Research and application of network status prediction based on BP neural network for intelligent production line," *Procedia Computer Science*, vol. 183, no. 20, pp. 189–196, 2021.
- [19] G. J. Oladehinde, K. O. Popoola, and A. A. Makinde, "Urban expansion and rural landscape transformations in selected communities of Obafemi Owode local government area of Ogun State Nigeria," *Social Sciences*, vol. 1, no. 7, pp. 191–191, 2021.
- [20] V. Somai, F. Kreis, A. Gaunt et al., "Genetic algorithm-based optimization of pulse sequences," *Magnetic Resonance in Medicine*, vol. 87, no. 5, pp. 2130–2144, 2022.
- [21] H. Wu, X. Wang, M. Chu, D. Li, L. Cheng, and K. Zhou, "HCMB: a stable and efficient algorithm for processing the normalization of highly sparse Hi-C contact data," *Computational and Structural Biotechnology Journal*, vol. 19, pp. 2637–2645, 2021.
- [22] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 925–938, 2020.
- [23] S. Karmakar and S. Goswami, "A new procedure for optimization of hidden layer neurons during learning through gradient descent process of neural network and improvement of performance in the chaos forecasting," *Iran Journal of Computer Science*, vol. 4, no. 4, pp. 293–303, 2021.
- [24] M. Sivaram, K. Batri, A. S. Mohammed, V. Porkodi, and N. V. Kousik, "Data fusion using Tabu crossover genetic algorithm in information retrieval," *Journal of Intelligent and Fuzzy Systems*, vol. 39, no. 4, pp. 5407–5416, 2020.
- [25] N. Jatana and B. Suri, "Particle swarm and genetic algorithm applied to mutation testing for test data generation: a comparative evaluation," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 4, pp. 514–521, 2020.