Research Article

Han Lai*

A comparative study of different neural networks in predicting gross domestic product

https://doi.org/10.1515/jisys-2022-0042
received November 24, 2021; accepted March 01, 2022

Abstract: Gross domestic product (GDP) can well reflect the development of the economy, and predicting GDP can help better grasp the future economic trends. In this article, three different neural network models, the genetic algorithm – back-propagation neural network model, the particle swarm optimization (PSO) – Elman neural network (Elman NN) model, and the bat algorithm – long short-term memory model, were analyzed based on neural networks. The GDP data of Sichuan province from 1992 to 2020 were collected to compare the performance of the three models in predicting GDP. It was found that the mean absolute percentage error values of the three models were 0.0578, 0.0236, and 0.0654, respectively; the root-mean-square error values were 0.0287, 0.0166, and 0.0465, respectively; and the PSO-Elman NN model had the best performance in GDP prediction. The experimental results demonstrate that neural networks were reliable in predicting GDP and can be used for further applications in practice.

Keywords: neural network, Sichuan province, gross domestic product, optimization algorithm, forecasting

1 Introduction

Gross domestic product (GDP) [1] can reflect the economic operation comprehensively and is an important basis for the relevant national departments to formulate economic development strategies and plans. With the rapid development of productivity, GDP data have received more and more attention. The prediction of GDP is related to the formulation of various economic and monetary policies and has a close relationship with the development of society. Many methods have been applied in the prediction of data. Suzuki et al. [2] predicted micro-meteorological data based on the support vector regression model and found that the method reduced the prediction error by 0.1% and the computation time by 98.7% through experiments on the prediction of temperature in Sapporo. Rau et al. [3] predicted the development of liver cancer within 6 years of diagnosis with type II diabetes. They established artificial neural network (ANN) and logistic regression models on 2,060 cases and found that ANN had better performance and could correctly predict 75.7% of diabetic patients receiving a future diagnosis of liver cancer and could correctly predict 75.5% not being diagnosed with liver cancer. Bo et al. [4] designed a homologous gray prediction model with one variable and one first-order equation (HGEM(1,1)) to predict the total energy consumption of the manufacturing industry of China. The experiment found that the method had a favorable comprehensive performance. Xu et al. [5] studied the prediction of short-term time series data of customer electricity consumption, reduced the dimensionality of the data with principal component analysis, and used a long-short memory (LSTM) network for prediction. The experimental results showed that the method

* **Corresponding author: Han Lai,** Sichuan TOP IT Vocational Institute, No. 2000, West District Avenue, High-Tech West District, Chengdu, Sichuan 611743, China, e-mail: jihan522819@21cn.com

Open Access. © 2022 Han Lai, published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.

had high prediction performance and generality. Jiang et al. [6] extracted data related to COVID-19 from the data of two hospitals in Wenzhou, Zhejiang, China, predicted severe cases using an artificial intelligence framework, and found through experiments that the accuracy of the method was 70–80%. Ingle and Deshmukh [7] used term frequency-inverse document frequency features extracted from online news data of companies in the Bombay Stock Exchange along with other stock market features for prediction, predicted the next day's stock price by ensemble deep learning framework, and found that the method had an accuracy of about 85%. Most of the current research on data prediction is about the optimization and study of a particular data prediction method, and there are fewer studies on the performance comparison of different methods on the same data set. Thus, this article compared the performance of several different neural network models for GDP prediction and carried experiments using GDP data from Sichuan Province to find the model with the best performance for effective GDP prediction. This article verifies the reliability of the particle swarm optimization (PSO) – Elman neural network (Elman NN) model in predicting GDP by comparing different neural network models and provides a more accurate method for data prediction.

2 Different neural network models

The performance of the model can be significantly improved by optimizing the data model through meta-heuristic algorithms and swarm intelligence algorithms. The principle of these algorithms is to solve optimization problems that are difficult to solve directly by defining group behavior and individual behavior so that the group has group evolutionary diversity and behavioral directionality. These algorithms are simple to implement and will not affect the solution of the problem due to individual failures, so they have been extensively used. For example, Noori et al. [8] optimized the ANN by the PSO algorithm and achieved higher precision in predicting the performance of tunnel boring machines. Mikaeil et al. [9] determined the clustering of tunnel engineering risks with the metaheuristic algorithm (genetic algorithm [GA]). Salemi et al. [10] classified the concrete lining of tunnels with a GA. Guido et al. [11] evaluated road safety data by combining PSO and GA with the K-means algorithm. Therefore, to obtain better effects in GDP prediction, this article optimized different prediction models with different algorithms.

2.1 GA-optimized back-propagation neural network (BPNN) model for GDP prediction

BPNN [12] has good performance in the prediction of time series, including short-term and long-term prediction, and has successful applications in the prediction of industrial [13] and medical [14] data. To enhance the performance of BPNN, this article uses a GA to improve it.

Taking a simple three-layer BPNN as an example, let the node in the input layer as

$$X = (x_1, x_2, \cdots, x_n)^T, \tag{1}$$

the node in the output layer as

$$Y = (y_1, y_2, \cdots, y_m)^T, \tag{2}$$

the node in the hidden layer as

$$H = (h_1, h_2, \cdots, h_p)^T, \tag{3}$$

the input of the output layer is h_k and the output is y_k. Then,

$$h_k = \sum_i w_{ij} h_i, \tag{4}$$

$$y_k = f(h_k), \tag{5}$$

$$h_i = \sum_i w_{ji}h_i, \tag{6}$$

$$y_i = f(h_i), \tag{7}$$

where w_{ij} and w_{ji} refer to the weight between the hidden layer and the input layer and the weight between the hidden layer and the output layer and $f(\cdot)$ is the activation function, i.e., the sigmoid function

$$\left(f(x) = \frac{1}{1 + e^{-x}}\right), \tag{8}$$

in this article.

Then, the error is back-propagated to optimize the weights and thresholds, and the objective function of the model can be written as:

$$E_n = \frac{1}{2}\sum_k (y_{nk} - \widehat{y_{nk}})^2, \tag{9}$$

where y_{nk} and $\widehat{y_{nk}}$ are the expected and actual outputs of the kth neuron in the output layer.

A GA is a biological evolution-based method applicable to the optimization of complex systems [15], enabling the search for optimal solutions through the simulation of natural evolutionary processes. It can be well integrated with other algorithms [16]. The specific steps of the GA-BPNN model are as follows.

(1) The population is initialized, and the parameters to be optimized are encoded. For a BPNN network with an N–M–L structure, the length S of chromosomes in GA is as follows:

$$S = L \times (N + 1) + M \times (L + 1). \tag{10}$$

(2) The adaptation function used is the error of the BPNN:

$$E_n = \frac{1}{2}\sum_k (y_{nk} - \widehat{y_{nk}})^2. \tag{11}$$

(3) The selection, crossover, and mutation operations are performed. The selection is done by random traversal sampling. The crossover is done by a single-point crossover operator. The mutation is performed using a random method: the gene subject to mutation is selected, e.g., if the selected gene has a code of 1, it is mutated as 0; otherwise, it is mutated as 1.
(4) Steps (2) and (3) are repeated until the error meets the requirements, and the optimized parameters are output.
(5) The obtained parameters are input into BPNN to build the GDP prediction model.

2.2 Particle swarm algorithm-optimized Elman NN model for GDP prediction

Elman NN is a local regression network [17], which is characterized by an additional undertaking layer compared to BPNN. Let the input layer be U, the output layer be Y, the hidden layer be X, and the undertaking layer be X_c. The expressions of each layer can be written as:

$$\begin{cases} Y(k) = g(w_2X(k) + b_2), \\ X(k) = f(w_1X_c(k) + w_1(U(k-1)) + b_1), \\ X_c(k) = X(k-1), \end{cases} \tag{12}$$

where w_1 is the weight from the input layer to the hidden layer, w_2 refers to the weight from the undertaking layer to the hidden layer, b_1 refers to the threshold value of the hidden layer, b_2 refers to the threshold value of the output layer, $g(\cdot)$ is the activation function of the output layer, and $f(\cdot)$ is the activation function of the hidden layer.

Similar to BPNN, the weights and thresholds of Elman NN also have an impact on the performance of the network. Elman NN was optimized using the PSO algorithm in this article. The PSO algorithm is a global stochastic optimization algorithm [18]. Based on the predation process of birds [19], every bird is considered a particle. Suppose there are m particles in a D-dimensional space, where the position of the ith particle is

$$x_i = (x_{i1}, x_{i2}, \cdots, x_{iD}), \tag{13}$$

the velocity is

$$v_i = (v_{i1}, v_{i2}, \cdots, v_{iD}), \tag{14}$$

the current optimal position is

$$p_i = (p_{i1}, p_{i2}, \cdots, p_{iD}), \tag{15}$$

and the optimal global position is

$$p_g = (p_{g1}, p_{g2}, \cdots, p_{gD}). \tag{16}$$

Then, the update formula for the velocity and position of the particle can be written as:

$$v_{id}(t+1) = v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (p_{gd} - x_{id}(t)), \tag{17}$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1), \tag{18}$$

where c_1 and c_2 are non-negative constants and r_1 and r_2 are random numbers in [0, 1].

The specific steps of the PSO-Elman NN model are as follows.

(1) An Elman NN model is developed, and its network structure is determined.
(2) The parameters of PSO are initialized.
(3) The initial fitness value of the particle is calculated, and then the particle is updated according to the formula above.
(4) The updated particle fitness value is calculated. The current optimal position and the global optimal position are updated until the maximum number of iterations is reached.
(5) The obtained parameters are input into Elman NN to build the GDP prediction model.

2.3 Bat algorithm (BA) – LSTM neural network model for GDP prediction

LSTM is an improvement of recurrent neural network (RNN) [20], which has very wide applications in language models [20], artificial intelligence, etc. [21], and also performs well in the prediction of various data [22]. It has higher prediction accuracy when it is combined with other algorithms. In this article, the BA is combined with LSTM for GDP prediction.

The difference between LSTM and RNN is that the LSTM includes a memory unit to keep the historical information and includes three gates in the hidden layer: the input gate i_t, the forgetting gate f_t, and the output gate o_t. The update formula of the LSTM at layer t is written as:

$$\begin{cases} i_t = \sigma(w_i \times [h_{t-1}, x_t] + b_i) \\ f_t = \sigma(w_f \times [h_{t-1}, x_t] + b_f) \\ o_t = \sigma(w_o \times [h_{t-1}, x_t] + b_o) \\ \tilde{c}_t = \tanh(w_c \times [h_{t-1}, x_t] + b_c) \\ c_t = f_t c_{t-1} + i_t \odot \tilde{c}_t \\ h_t = o_t \odot \tanh(c_t). \end{cases} \tag{19}$$

where h_{t-1} refers to the last output, x_t refers to the current input, σ refers to the activation function, w_i and b_i are the weight and bias of the input gate, w_f and b_i are the weight and bias of the forgetting gate, w_o and b_o are the weight and bias of the output gate, c_{t-1} is the status of the previous memory unit, and c_t is the status of the current memory unit.

The loss function used in this article is the mean square error. The Adam function is used as the optimizer, the network is trained using the small-batch gradient descent algorithm, and the network parameters are optimized using the BA.

The BA is a heuristic algorithm [23], which finds the optimal global solution by simulating the bat localization process [24]. Let the number of bats in a D-dimensional space be N, the upper and lower limits of the pulse frequency are $Q_{\max}$ and $Q_{\min}$, the position of the ith bat is

$$X_i = [x_1, x_2, \cdots, x_N], \tag{20}$$

and the velocity is

$$V_i = [v_1, v_2, \cdots, v_N], \tag{21}$$

then the update equations of its frequency, velocity, and position are as follows:

$$Q_i = Q_{\min} + \beta \times (Q_{\max} - Q_{\min}), \tag{22}$$

$$v_i(t) = v_i(t-1) + Q_i \times (x_i - g_{\text{best}}), \tag{23}$$

$$x_i(t) = x_i(t-1) + v_i(t), \tag{24}$$

where $\beta \in [0,1]$ and g_{best} is the optimal global position. A random number is generated to compare with the frequency; if the random number is larger, a random perturbation is performed:

$$X_i(t) = g_{\text{best}} + A_t \times \delta, \tag{25}$$

and if the random number is smaller, the cross-border processing is performed:

$$X_i(t) = \text{judgebound}(X_i(t)). \tag{26}$$

A_t is the average loudness of the bat, and $\delta \in [-1,1]$. The judgebound() function is used to implement cross-border processing. When the bat finds the target, the update formulas for frequency and loudness are as follows:

$$A_i(t+1) = \alpha A_i(t), \tag{27}$$

$$r_i(t+1) = r_i(0)[1 - \exp(-\varepsilon t)], \tag{28}$$

where α and ε are constants. If the new fitness value is smaller than the global optimal fitness value, then

$$x_i(t) = g_{\text{best}}. \tag{29}$$

The BA-optimized parameters are input into the LSTM model. The specific steps of the BA-LSTM model are as follows.

(1) The LSTM model is established, and the network structure is set up.
(2) The parameters of the BA are initialized, and all bats are iterated.
(3) The frequency, speed, and position of the bat are updated according to the formula.
(4) The fitness value is calculated. The frequency and loudness are updated until the termination condition is satisfied. The optimal parameters are output.
(5) The obtained parameters are input into the LSTM to build the GDP prediction model.

3 Experimental analysis

3.1 Experimental data

This article focuses on the GDP forecast of Sichuan Province. In 2020, the GDP of Sichuan Province reached 485.98 billion yuan, showing an increase of 3.8% over the previous year, the economic growth was fast, the industrial system was complete, and industries, such as nuclear power equipment and heavy combustion engines, ranked among the top in the country. The GDP data of Sichuan province between 1992 and 2020 provided by the National Bureau of Statistics are shown in Table 1. The GDP of the first 4 years was used as input, the GDP of the fifth year was used as output, and so on. The model was validated using the data between 2010 and 2020, as shown in Table 2.

Table 1: GDP of Sichuan Province between 1992 and 2020 (unit: billion yuan) [25]

Time	GDP
1992	1177.3
1993	1496.1
1994	2001.4
1995	2443.2
1996	2871.7
1997	3241.5
1998	3474.1
1999	3649.1
2000	3928.2
2001	4293.5
2002	4725.0
2003	5346.2
2004	6304.0
2005	7195.9
2006	8494.7
2007	10562.1
2008	12756.2
2009	14190.6
2010	17224.8
2011	21050.9
2012	23922.4
2013	26518.0
2014	28891.3
2015	30342.0
2016	33138.5
2017	37905.1
2018	42902.1
2019	46363.8
2020	48598.8

Table 2: Validation sample

Sample number	Model input	Model output
1	2007–2010 GDP	2011 GDP
2	2008–2011 GDP	2012 GDP
3	2009–2012 GDP	2013 GDP
4	2010–2013 GDP	2014 GDP
…	…	…
10	2016–2019 GDP	2020 GDP

In the evaluation of the prediction model, to have a better understanding of the prediction effect, it is generally expressed by the prediction error, i.e., the distance between the actual value and the predicted value. Commonly used indicators included mean absolute error, median absolute error, etc. The performance of the different models was analyzed using the following two indicators:

Mean absolute percentage error (MAPE): it normalizes the error of every point, reflecting prediction precision, and its calculation formula is as follows:

$$\mathrm{MAPE} = \frac{1}{N} \times \sum_{i=1}^{N} \left| \frac{y_i - y_i'}{y_i} \right| \times 100\%. \tag{30}$$

Root-mean-square error (RMSE): it reflects the degree of deviation between the predicted value and the actual value, and its calculation formula is as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N} \times \sum_{i=1}^{N} (y_i - y_i')^2}, \tag{31}$$

where y_i and y_i' are the actual and predicted values of GDP, respectively.

3.2 Experimental setup

As GDP belongs to normal time series, selecting a general three-layer structure was enough for the neural network model; therefore, the GDP of the first 4 years were used as the input, i.e., the number of input nodes was 4, and the GDP of the fifth year was used as the output, i.e., the number of output nodes was 1. The other settings are as follows.

(1) GA-BPNN model: according to the experimental data, the node N of the input layer in the BPNN was 4 and the node L of the output layer was 1, +1. A network with a 4–9–1 structure was obtained. The parameters that GA needed to optimize were determined as 23 according to $S = L \times (N + 1) + M \times (L + 1)$. The population size was 60. The maximum number of iterations was 70. The mutation probability was 0.1. The crossover probability was 0.7. The training number of the model was 100. The learning rate was 0.1.

(2) PSO-Elma and the node M of the hidden layer was determined as 9 using the empirical formula: $M = 2Nn$ NN model: referring to the BPNN model, the structure of Elman NN was also set as 4–9–1. Let the population size of PSO be 20, $r_1 = 1$, $r_2 = 0.5$, $c_1 = c_2 = 2$, and the maximum number of iterations be 1,000.

(3) BA-LSTM model: the node of the input layer was 4, the node of the output layer was 1, the node of the hidden layer was 10, the time step was 1, the maximum number of iterations was 1,000, the population size of BA was 30, and the frequency and loudness were 0.5.

3.3 Predicted results

The GDP prediction results of different models are shown in Figure 1.

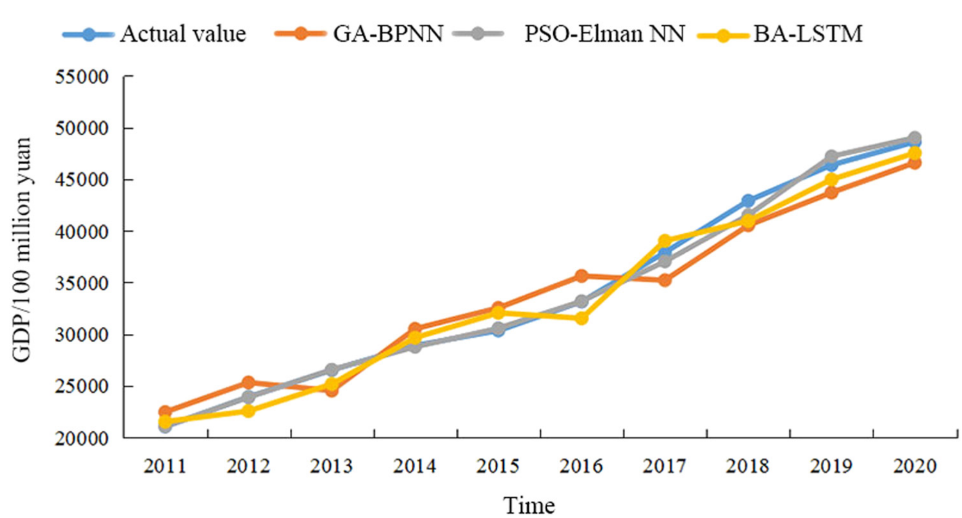

Figure 1: GDP prediction results of different models.

It is seen from Figure 1 that there were some differences in the prediction results of different models for the prediction of GDP. The results of GA-BPN and BA-LSTM models had large errors with the actual values and large data volatility, showing unstable prediction performance, while the prediction results of the PSO-Elman NN model had a good agreement with the actual values. To further understand the model performance, the MAPE and RMSE values were calculated and compared, and the results are shown in Figures 2 and 3.

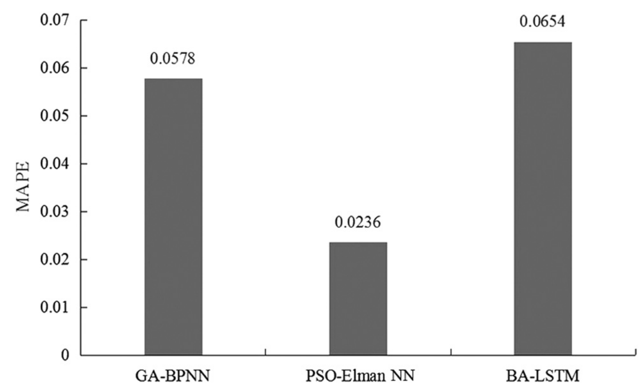

Figure 2: Comparison of MAPE values between different models.

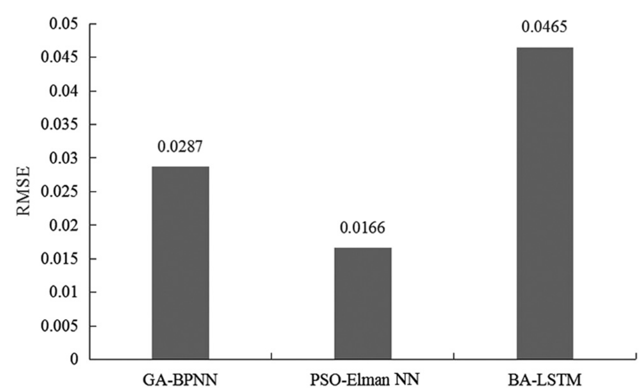

Figure 3: Comparison of RMSE values between different models.

It is seen from Figure 2 that the MAPEs of the three models were 0.0578, 0.0236, and 0.0654, i.e., the MAPE of the BA-LSTM model was the smallest, followed by GA-BPNN and PSO-Elman NN models. The MAPEs of the GA-BPNN and BA-LSTM models were above 0.05. The MAPE of the PSO-Elman NN model was 0.0236, the smallest, 0.0342 smaller than the GA-BPNN model and 0.0418 smaller than the BA-LSTM model. The above results indicated that the PSO-Elman NN model had a smaller prediction error, showing better performance in predicting GDP.

It is seen from Figure 3 that the RMSE values of all the models were above 0.01; the RMSE value of the PSO-LSTM model was the smallest, followed by GA-BPNN and BA-LSTM models. The RMSE of the BA-LSTM model was 0.0465, the largest, and the RMSE of the PSO-Elman NN model was 0.0166, which was 0.0121 smaller than the GA-BPNN model and 0.0299 smaller than the BA-LSTM model. The smaller the RMSE value was, the smaller the difference between the prediction result and actual value was, i.e., the higher the degree of fitting was. The results indicated that the PSO-Elman NN model had better performance in GDP prediction and its prediction results were closer to the actual values; thus, it could achieve better applications in practice.

4 Discussion

A neural network is a way to process data by simulating the information processing mechanism of the human brain, which can realize large-scale parallel processing. Neural networks have strong adaptive and self-organizing abilities. The neural network model consists of basic neurons, each of which is independent but interconnected, so it has good performance in processing complex information and has been very widely used in industrial control [26], pattern recognition [27], and prediction estimation [28]. This article mainly compared three neural network models, BPNN, Elman NN, and LSTM models, and optimized every model to achieve a better prediction of GDP.

It is seen from the data in Table 1 that the GDP of Sichuan Province showed a trend of steady growth. To be specific, in the primary industry, the farming system in Sichuan Province is three times a year, so the yields of food crops and cash crops are high and in a steady state of improvement; in the secondary industry, Sichuan Province has a full range of industrial sectors, strategic emerging industries, such as information technology and new energy, are in rapid development, and the development of the construction industry is also fast; in the tertiary industry, Sichuan Province has a wide range of financial institutions and a high degree of openness, and domestic trade and foreign economy are also developing rapidly. It is seen from Figure 1 that the prediction results of GA-BPNN and BA-LSTM models had large errors with the actual values, and fluctuations of the values were also significant, which indicated that the stability of the models was relatively general, but the prediction results of the PSO-Elman NN model almost coincided with the actual values. Then, it is seen from the comparison of MAPE and RMSE values (Figures 2 and 3) that the MAPE value of the PSO-Elman NN model was 59.17% smaller than that of the GA-BPNN model and 63.01% smaller than that of the BA-LSTM model; the RMSE value of the PSO-Elman NN model was 42.16% smaller than that of the GA-BPNN model and 64.3% smaller than the BA-LSTM model.

The experimental results verify the accuracy of the PSO-Elman NN model in GDP prediction, and the model can be applied in practical cases. For example, it can predict the future GDP trend of provinces and cities to help governments to make timely and accurate responses to economic changes and provide scientific guidance for the next development decision. Changes in GDP are affected by indicators, such as residents' consumption, cash circulation, gold reserves, foreign investment, and taxation. According to the forecast results of GDP, in the future development, Sichuan Province can further implement policies favorable to industrial development, promote residents' consumption, and flexibly apply the forecast model to grasp the GDP trend of Sichuan Province in time and make adjustments to relevant policies.

5 Conclusion

This article compared several different models, GA-BPNN, PSO-Elman NN, and BA-LSTM models, for the prediction problem of GDP in Sichuan Province, and trained and tested the models using the GDP data of Sichuan Province from 1992 to 2020. It was found that the PSO-Elman NN model had an MAPE value of 0.0236 and an RMSE value of 0.0166, showing the best prediction performance. The PSO-Elman NN model can be further studied and applied to realize the accurate prediction of GDP and stable and healthy development of the economy. However, this study also has some limitations, such as the limited comparison of experimental data and insufficient comparison and optimization of neural network models. In future research, experiments will be conducted on larger-scale data, more neural network models will be investigated, and more in-depth research on the improvement of PSO will also be conducted to further improve the performance of the models.

Conflict of interest: Author states no conflict of interest.

References

[1] Lakštutienė A. Correlation of the indicators of the financial system and gross domestic product in European Union Countries. Eng Econ. 2015;32:7–18.
[2] Suzuki Y, Kaneda Y, Mineno H. Analysis of support vector regression model for micrometeorological data prediction. Gastroenterology. 2015;3:37–48.
[3] Rau H, Fuad A, Hsu CY, Wei LM, Lin YA, Hsu MH, et al. Development of a web-based liver cancer prediction model for type II diabetes patients by using an artificial neural network. Comput Method Prog Bio. 2016;125:58–65.
[4] Bo Z, Zhou M, Zhang J. Forecasting the energy consumption of China's manufacturing using a homologous grey prediction model. Sustainability. 2017;9:1975.
[5] Xu K, Hou R, Ding X, Tao Y, Xu Z. Short-term time series data prediction of power consumption based on deep neural network. IOP Conf Ser Mater Sci Eng. 2019;646:012027.
[6] Jiang X, Coffee M, Bari A, Wang J, Jiang X, Huang J, et al. Towards an artificial intelligence framework for data-driven prediction of coronavirus clinical severity. Comput Mater Con. 2020;62:537–51.
[7] Ingle V, Deshmukh S. Ensemble deep learning framework for stock market data prediction (EDLF-DP) – ScienceDirect. Glob Transit Proc. 2021;2:47–66.
[8] Noori AM, Mikaeil R, Mokhtarian M, Haghshenas SS, Foroughi M. Feasibility of intelligent models for prediction of utilization factor of TBM. Geotech Geol Eng. 2020;38:3125–43.
[9] Mikaeil R, Shaffiee Haghshenas S, Sedaghati Z. Geotechnical risk evaluation of tunneling projects using optimization techniques (case study: the second part of Emamzade Hashem tunnel). Nat Hazards. 2019;97:1099–113.
[10] Salemi A, Mikaeil R, Haghshenas SS. Integration of finite difference method and genetic algorithm to seismic analysis of circular shallow tunnels (Case study: Tabriz urban railway tunnels). KSCE J Civ Eng. 2018;22:1978–90.
[11] Guido G, Haghshenas SS, Haghshenas SS, Vitale A, Astarita V, Haghshenas AS. Feasibility of stochastic models for evaluation of potential factors for safety: a case study in Southern Italy. Sustainability. 2020;12:7541.
[12] Arthur CK, Temeng VA, Ziggah YY. Performance evaluation of training algorithms in backpropagation neural network approach to blast-induced ground vibration prediction. Ghana Mining J. 2020;20:20–33.
[13] Li M, Wu H, Wang Y, Handroos H, Carbone G. Modified Levenberg–Marquardt algorithm for backpropagation neural network training in dynamic model identification of mechanical systems. J Dyn Syst Meas Control. 2017;139:031012.
[14] Puspita JW, Jaya AI, Gunadharma S. Classification of epileptiform and wicket spike of EEG pattern using backpropagation neural network. AIP Conference Proceedings. 2017;1825:020018.
[15] Stevanovic A, Martin P, Stevanovic J. VisSim-based genetic algorithm optimization of signal timings. Transp Res Rec. 2015;2035:59–68.
[16] Wu J, Long J, Liu M. Evolving RBF neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm. Neurocomputing. 2015;148:136–42.
[17] Liu H, Tian HQ, Liang XF, Li YF. Wind speed forecasting approach using secondary decomposition algorithm and Elman neural networks. Appl Energ. 2015;157:183–94.
[18] Wang D, Tan D, Lei L. Particle swarm optimization algorithm: an overview. Soft Comput. 2018;22:387–408.
[19] Khan SU, Yang S, Wang L, Liu L. A modified particle swarm optimization algorithm for global optimizations of inverse problems. IEEE Trans Magn. 2016;52:1–4.
[20] Sundermeyer M, Ney H, Schluter R. From feedforward to recurrent LSTM neural networks for language modeling. IEEE/ACM Trans Audio Speech. 2015;23:517–29.
[21] Fan H, Zhu L, Yang Y. Cubic LSTMs for video prediction. Proc AAAI Conf Artif Intell. 2019;33:8263–70.
[22] Lam PD, Ahn H, Kim K, Kim K. Process-aware enterprise social network prediction and experiment using LSTM neural network models. IEEE Access. 2021;1.
[23] Raghavan S, Sarwesh P, Marimuthu C, Chandrasekaran K. Bat algorithm for scheduling workflow applications in cloud. 2015 International Conference on Electronic Design, Computer Networks and Automated Verification, 29–30 Jan 2015. Shillong, India: IEEE; 2015. p. 139–44.
[24] Ashok KN, Rao GS, Lavanya B, Novel BAT. Algorithm for position estimation of a GPS receiver located in coastal region of Southern India. Proc Comput Sci. 2018;143:860–7.
[25] https://data.stats.gov.cn/easyquery.htm?cn=E0103&zb=A0201&reg=510000&sj=2020.
[26] El-Sousy F, Abuhasel KA. Adaptive nonlinear disturbance observer using a double-loop self-organizing recurrent wavelet neural network for a two-axis motion control system. IEEE Trans Ind Appl. 2018;54:764–86.
[27] Esteva A, Kuprel B, Novoa RA, Ko J, Swetter SM, Blau HM, et al. Dermatologist-level classification of skin cancer with deep neural networks. Nature. 2017;542:115–8.
[28] Alanis AY. Electricity prices forecasting using artificial neural networks. IEEE Lat Am Trans. 2018;16:105–11.