Geoscience and Remote Sensing (2024)
Clausius Scientific Press, Canada

DOI: 10.23977/geors.2024.070102
ISSN 2523-2592 Vol. 7 Num. 1

Sunspot activity prediction based on adaptive hybrid algorithms

Chunxu Zhang[1,*,#], Yesheng Liang[1,#]

[1]*School of Accounting, Tianjin University of Finance and Economics, Tianjin, China*
[*]*Corresponding author*
[#]*These authors contributed equally to this work.*

Keywords: ARIMA Model, BP Neural Network, Hybrid ARIMA-BP Neural Network Model, Multivariate Nonlinear Regression, Differential Evolutionary Algorithm

Abstract: In this paper, by combining the ARIMA model and the BP neural network model, we establish an adaptive hybrid ARIMA-BP neural network model, which provides more accurate results for sunspot prediction. For solar activity prediction, in this paper, based on the multivariate nonlinear regression and BP neural network model, we utilize the differential evolutionary algorithm for model solving and obtain satisfactory hybrid model solving results. These results provide new perspectives and methods for solar activity prediction, and provide useful references and insights for research and practice in related fields.

1. Introduction

Solar activity, especially the appearance of sunspots on the surface of the sun, is a fascinating phenomenon of great significance for space weather forecasting and all aspects of the Earth's atmospheric conditions. Sunspots are temporary black spots on the solar sphere generated by a concentrated magnetic flux, leading to a local temperature reduction and convective suppression. Sunspots occur within active regions, often appear in pairs, have opposite magnetic poles, and exhibit periodic patterns consistent with a solar cycle of about 11 years.

At present, some scholars have carried out research in related fields. Yuan et al [1] proposed a PV power prediction method combining two techniques. The method uses a fast correlation filtering algorithm to extract meteorological features with a strong correlation with PV power generation. The full systematic empirical modal decomposition method with an adaptive noise model is used to decompose the data into high and low-frequency components, which reduces the volatility of the data. Then, the long-short-term neural network and deep confidence network are combined into a new prediction model for each component. Finally, the proposed combined PV power prediction method is analyzed by examples and compared with other prediction methods. The results show that the proposed combined prediction method has high prediction accuracy. Chen et al [2] proposed a hybrid ARIMA-LR algorithm based on a Bayesian combinatorial model, which demonstrated outstanding performance in targeting the prediction of air cargo volume. The algorithm is adaptive with respect to the movement of the series and reacts quickly to sudden changes. Moustafa et al [3] used three single and hybrid models, Long Short-Term Memory (LSTM), Autoregressive Integrated Moving Average (ARIMA), and Seasonal Autoregressive Integrated Moving Average (SARIMA), for

forecasting the maximum number of blacks for cycles 25 and 26. The hyperparameters of the singular models were optimized using a Bayesian optimization approach. The LSTM-ARIMA hybrid model gave the best performance. The outstanding results of the LSTM-ARIMA model show the potential of the hybrid approach in improving the overall performance. In addition, the ability of the LSTM model to outperform the ARIMA model demonstrates the ability of the LSTM network to learn from time-series data. Dang et al [4] compared three important non-deep learning models, four popular deep learning models, and their five integrated models for predicting sunspot numbers. In particular, an integrated model called XGBoost- dl is proposed which uses XGBoost as a two-level nonlinear integration method to combine deep learning models. The proposed XGBoost-DL obtains the best predictive performance in the comparison (RMSE and MAE) and outperforms the best non-deep learning model SARIMA (RMS) and MAE), outperforming the best non-deep learning model SARIMA (RMSE) The best deep learning model, Informer (RMSE and MAE) and MAE), the best deep learning model Informer (RMSE) and NASA's predictions (RMSE) and MAE). Our XGBoost-DL predicts a peak sunspot number of 133.47 in May 2025 for solar cycle 25 and 164.62 in November 2035 for solar cycle 26, which is similar to NASA's predictions of 137.7 in October 2024 and 161.2 in December 2034 Tabassum et al. [5] have estimated the sunspot number (SN) predictions over the recent solar cycle 24. To find the best model, moving average (MA), exponential smoothing (ES) and autoregression (AR) were used. In addition to this, in two other experiments, seasonal components were extracted using moving average (MA) and exponential smoothing (ES) and trend components were calculated with the help of simple regression analysis (RA). This exploration was solely to understand the differences between these models and the impact of these two components on the prediction of sunspots using moving average (MA) and exponential smoothing (ES). The forecast results reveal this difference and impact. Lessons are provided for other time series analysis (TSA) models to predict sunspot numbers.

In this paper, sunspot numbers and periods are predicted by constructing adaptive ARIMA-BP neural networks and adaptive multiple nonlinear regression-BP neural network models.

2. Influence factor prediction based on adaptive ARIMA-BP neural network modeling

2.1. ARIMA modeling

The essence of the ARIMA model is the combination of the difference operation with the ARMA model, denoted as ARIMA (p,d,q). The ARIMA model can be formulated as:

$$\varphi(B)(1-B)^d y_t = \theta(B)\varepsilon_t \tag{1}$$

where y_t is a time series of historical observations, d is the order of the difference, p and q are the autoregressive model order and the moving average of previous observations, and ε_t is a sequence of independent and identically distributed white noise with zero mean and constant variance. b is the lag operator, and B satisfies the following expression:

$$B^n y_t = y_{t-n} \tag{2}$$

$$\varphi(B) = 1 - \varphi_1 B - \cdots - \varphi_p B^p \tag{3}$$

$$\theta(B) = 1 - \theta_1 B - \cdots - \theta_q B^q \tag{4}$$

The focus of building an ARIMA (p,d,q) model is on the selection of the three parameters of (p,d,q). d is the order of the difference, and the purpose of the difference is to change the original series of observations into a smooth time series. In this paper, Bayesian Information Criterion (BIC) is used to select p and q. The Bayesian Information Criterion can give a simple approximation of the logit model evidence as follows.

$$BIC = \text{Accuracy}(m) - \frac{p}{2}\log N \tag{5}$$

where p is the number of parameters and N is the number of data points.

2.2. BP neural network modeling

BP neural network is a multilayer feed-forward algorithm that consists of input, hidden and output layers. There is work signal and error signal propagation between layers. Figure 1 shows the neural network structure.

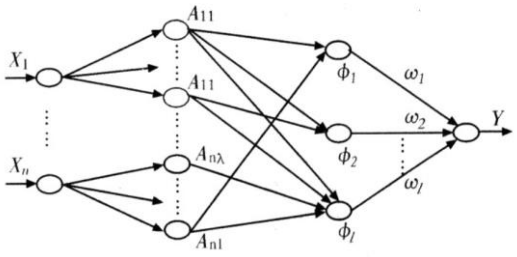

Figure 1: Schematic diagram of BP neural network

The principle of operation of the BP neural network is as follows:

Denote the training set as $\{(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots, (\boldsymbol{x}_m, \boldsymbol{y}_m)\}$, $\boldsymbol{x}_i \in R^d, \boldsymbol{y}_i \in R^l$, and the output as $\hat{\boldsymbol{y}}_k = (\hat{y}_1^k, \hat{y}_2^k, \ldots, \hat{y}_l^k)$. Then there are:

$$\hat{y}_j^k = f(\beta_j - \theta_j) \tag{6}$$

$$\beta_j = \sum_{i=1}^{n} w_{ij} x_{ij} \tag{7}$$

where w_{ij} is the connection weight of the ith neuron to the jth output. Remember that the error is when the network is on $(\boldsymbol{x}_k, \boldsymbol{y}_k)$:

$$E_k = \frac{1}{2}\sum_{j=1}^{l} \left(\hat{y}_j^k - y_j^k\right)^2 \tag{8}$$

When the neural network completes the forward computation, the error value is obtained by subtracting the predicted value from the actual value, followed by backpropagation to adjust the weight threshold of the neural network. The iterative update formula for $\boldsymbol{w}$ and $\boldsymbol{\theta}$ is given by:

$$\Delta\omega_{hj} = \eta g_j b_h \tag{9}$$

$$\Delta\theta_j = -\eta g_j \tag{10}$$

$$g_j = -\frac{\partial E_k}{\partial \hat{y}_j^k} \cdot \frac{\partial \hat{y}_j^k}{\partial \beta_j} = -\left(\hat{y}_j^k - y_j^k\right) \cdot (\hat{y}_j^k)' \tag{11}$$

Where b_h is the input data of this neuron. Based on this, the neural network constantly adjusts the weights and thresholds during its training process, so that the prediction error of the neural network is constantly approaching 0.

2.3. Prediction model construction based on the GABP neural network

In this paper, the genetic algorithm was used to optimize the BP neural network, and the forward propagation process of the BP neural network was used to calculate the fitness of each individual in the iterative process, so as to improve the optimization efficiency of the algorithm. The design framework of the algorithm is shown in the following algorithm.

Algorithm: IGABP

Input: training set independent variable x_{train}, training set dependent variable y_{train}, test set independent variable x_{test}
Output: test set dependent variable $\hat{y}_{test}$
//Data normalization
//x'_{train} is the normalized x_{train}, y'_{train} is the normalized y_{train}
// x_maxmin and y_maxmin are normalization information, used for back-normalization.
[x'_{train},x_maxmin]=mapminmax(x_{train}); // mapminmax is the min-max normalization function
[y'_{train},y_maxmin]=mapminmax(y_{train}); // Parameter definition.
//Parameter definition
Set BP neural network parameters: number of neurons in the hidden layer num_{hidden}
Set the parameters of the genetic algorithm: iteration $iter$, crossover rate P_c, variation rate P_m.
Randomly initialize population $popu_{init}$
// Genetic algorithm part
$popu = popu_{init}$
for $d = 1 \rightarrow iter$ **do** // iterate over the population
  crossover
  mutation
  for $p = 1 \rightarrow size(popu, 1)$ **do** //traverse each individual in the current generation
    Direct computation of adaptation by forward propagation using activation functions
  end
  Selection to obtain new populations of $popu$
end
// BP neural network
$\hat{y}_{test}$=[ ]
for $p = 1 \rightarrow size(popu, 1)$ **do** // traverse each individual
  Decoding of popu(p,:)
  Initialize the weights and thresholds of the BP neural network using the values of popu(p,:)
  Run the BP neural network and calculate the predicted value y'
  $\hat{y}_{test} = [\hat{y}_{test}; y']$
end

In the coding part of the genetic algorithm, IGABP continuously represents the weights and thresholds of the neural network as a vector for constituting the expression of individual genes. Since the structure of the network has been determined during the running of the algorithm, and the number of weights and thresholds to be determined have been determined, the length of the chromosome remains constant during the iteration process.

In the part of the genetic algorithm that calculates the fitness of an individual, compared to GABP uses decoded individuals to initialize the neural network and then calculates the fitness based on the output of the training, IGABP starts from the principle of the process of forward propagation of the neural network and directly calculates the fitness of an individual, which eliminates the amount of computation required for the training and improves the optimization efficiency of the algorithm.

Encoding and Decoding

Assume that the neural network used in BAGP is shown in Figure 2:

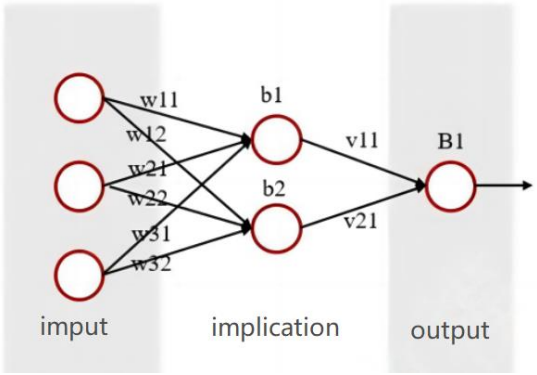

Figure 2: Neural network structure

Then, in determining the weights and thresholds of the network, the coding structure can be designed as:

w_{11}	w_{12}	w_{21}	w_{22}	w_{31}	w_{32}	b_1	b_2	v_{11}	v_{21}	B_1

In the chromosome designed in this paper, the gene loci are expressed in order: the weights between the input layer and the hidden layer, the threshold of the hidden layer, the weights between the hidden layer and the output layer, and the weights of the output layer, respectively. From this, the complete structure of a neural network can be determined.

2.4. Modeling of Adaptive Hybrid ARIMA-BP Neural Networks

For each prediction algorithm, some of the sequences are passed as a test set. The main idea of the hybrid algorithm designed in this article is that the better the performance in the past period's prediction the higher the weight in the future prediction and the higher the contribution to the predicted value.

For the actual observations are written as: $\boldsymbol{y_i} = \{y_1, y_2, y_3, \dots, y_t\}$. The predicted value of algorithm k is written as: $\hat{\boldsymbol{y}}_{\boldsymbol{k,i}} = \{\hat{y}_{k,1}, \hat{y}_{k,2}, \hat{y}_{k,3}, \dots, \hat{y}_{k,i}\}$. The prediction error can be denoted as:

$$wmape_{k,id} = \frac{\sum|y_i - \hat{y}_{k,i}|}{\sum y_i} \tag{12}$$

The total error of algorithm k can be formulated as:

$$wmape_k = \sum_{id} wmape_{k,id} \tag{13}$$

In a hybrid algorithm, if an algorithm performs better in the test set, the weight is higher. The weight of algorithm k in the hybrid algorithm can be denoted as:

$$\omega_k = \frac{1/mape_k}{\sum_k (1/wmape_k)} \tag{14}$$

where ω_k is the weight of algorithm k in the hybrid algorithm. In each calculation of the hybrid prediction value, it is necessary to combine the prediction value of the ARIMA algorithm and BP neural network algorithm. Its calculation formula can be expressed as:

2.5. Sunspot prediction results

The results of the model solution are shown in Figure 3:

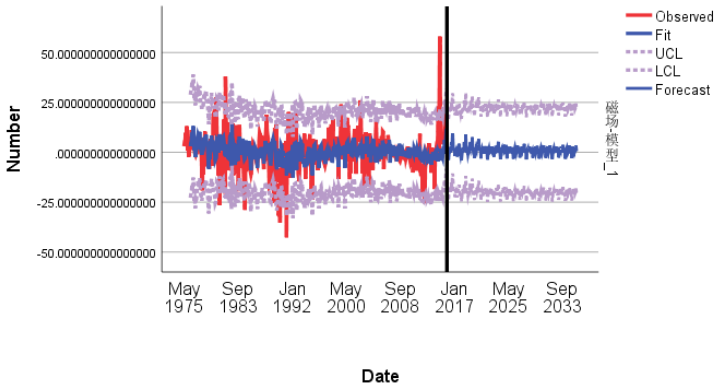

Figure 3: Sunspot predictions will result

From Figure 3, it can be found that the algorithm designed in this paper recognizes the periodic fluctuations better with better performance.

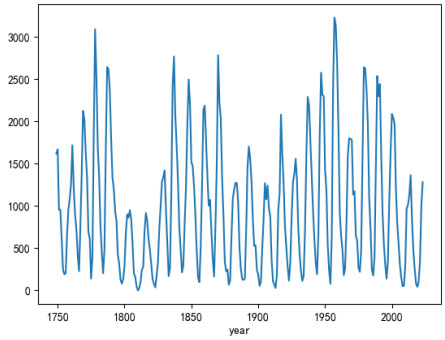

Figure 4: Total number of sunspots per year

Analyzing this in conjunction with the sunspot numbers shown in Figure 4, the solution is that the next solar cycle will begin in about 2031 and end in about 2042.

3. Solar Activity Prediction Based on Adaptive Multiple Nonlinear Regression-BP Neural Network Modeling

3.1. Modeling the relationship between time and sunspot number

In this paper, we can build a regression analysis model on the relationship between time and sunspot number, and quantify the relationship between time and sunspot number through regression analysis. When the time is determined, the sunspot number can be obtained, and the solar activity can be further inferred. We can express the relationship as:

$$y_i = f\left(x_1^i, x_2^i, \cdots, x_j^i, \theta_1, \theta_2, \cdots, \theta_p\right) + \sigma_i \varepsilon \ (i = 1,2, \cdots, n) \tag{15}$$

where y is the true value; i denotes the ith group of data; $f\left(x_1, x_2, \cdots, x_j, \theta_1, \theta_2, \cdots, \theta_p\right)$ is the multivariate nonlinear function, which denotes the deterministic part; $x_1, x_2, \cdots, x_j$ is the independent variable; $\theta_1, \theta_2, \cdots, \theta_p$ is the unknown model parameter of the multivariate nonlinear function; $\sigma_i \varepsilon$ is the stochastic part, ε is the random variable obeying N (0, 1) distribution; σ_i is the standard deviation of the random distribution of the ith set of data.

To observe the data distribution, a line graph is plotted as shown in Figure 5:

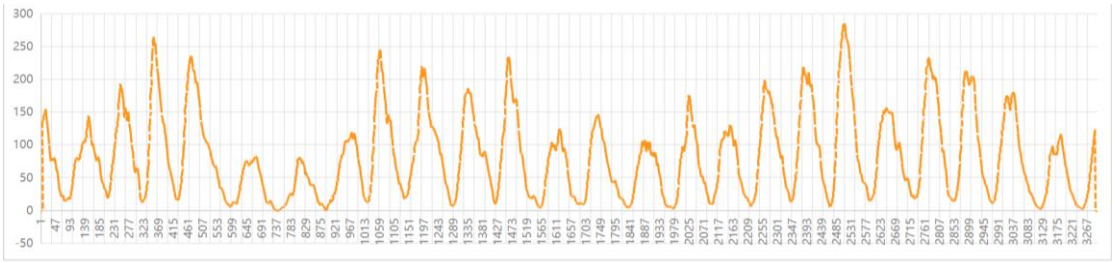

Figure 5: Total Number of sunspots per month

At present, most of the nonlinear regression models rely on empirical or experimental methods to select the regression model, but the empirical method will bring the problem of large errors, and the experimental method is time-consuming, but based on the experimental results can be better to select the correct model. Therefore, this paper determines the suitable regression model for the relationship through the experimental method:

$$y = p1 * Sin(p2 * x + p3) + p4 \tag{16}$$

3.2. Model solving based on differential evolutionary algorithm

According to the above regression model, we need to determine more parameters totaling 4. To solve this kind of multi-parameter optimization problem, we can generally use the gradient descent method or genetic algorithm. However, because the gradient descent method often easily falls into the local optimal solution, resulting in a large deviation from the final result, while the genetic algorithm's mutation operation is to try to find a better choice by generating a new solution, when it comes to the later stages of the optimization, the entire population may fall into the local optimum. At this time, the solution needs to be able to run out of the local optimal circle, and "ineffective" mutation can not achieve the purpose. Therefore, in this paper, we choose the differential evolutionary algorithm, which can better solve the global optimal problem, to optimize the solution of $\theta_1, \theta_2, \cdots, \theta_4$. The following are the solution steps:

Population initialization

The population size M is chosen as 100, and M individuals are randomly and uniformly generated in the solution space.

$$X_i(0) = \left(\mathrm{x}_{i,1}(0), x_{i,2}(0), x_{i,3}(0), \ldots, x_{i,n}(0)\right), i = 1,2,3, \ldots, M \tag{17}$$

Among them.

$$x_{i,j}(0) = L_{j\_\min} + \mathrm{rand}(0,1)\left(L_{j\_\max} - L_{j\_\min}\right), i = 1,2,3, \ldots, M, j = 1,2,3, \ldots, n \tag{18}$$

Mutation.

In the g-th iteration, three individuals $X_{p1}(g), X_{p2}(g), X_{p2}(g)$ are randomly selected from the population with $p1 \neq p2 \neq p3 \neq i$, generating a vector of variation:

$$H_i(g) = X_{p1}(g) + F \cdot \left(X_{p2}(g) - X_{p3}(g)\right) \tag{19}$$

where $\Delta_{p2,p3}(g) = X_{p2}(g) - X_{p3}(g)$ is the difference vector and F is the scaling factor.

The three randomly selected individuals in the variance operator are ranked from best to worst to obtain Xb, Xm, Xw, corresponding to the fitness fb, fm, fw, the variance operator reads:

$$V_i = X_b + F_i(X_m - X_w) \tag{20}$$

Also, the value of F varies adaptively according to the two individuals generating the difference

vector:

$$F_i = F_l + (F_u - F_l)\frac{f_m - f_b}{f_w - f_b}, F_l = 0.1, F_u = 0.9 \tag{21}$$

The mutation strategy is:

$$DE/\,rand\,/1: V_i(g) = X_{p1}(g) + F\big(X_{p2}(g) - X_{p3}(g)\big) \tag{22}$$

$$DE/\,best\,/1: V_i(g) = X_{\text{best}}(g) + F\big(X_{p1}(g) - X_{p2}(g)\big) \tag{23}$$

$$DE/\,current\ to\ best/1: V_i(g) = X_i(g) + F\big(X_{\text{best}}(g) - X_i(g)\big) + F\big(X_{p1}(g) - X_{p2}(g)\big) \tag{24}$$

$$DE/\,best\,/2: V_i(g) = X_{\text{best}}(g) + F\big(X_{p1}(g) - X_{p2}(g)\big) + F\big(X_{p3}(g) - X_{p4}(g)\big) \tag{25}$$

$$DE/\,rand\,/2: V_i(g) = X_{p1}(g) + F\big(X_{p2}(g) - X_{p3}(g)\big) + F\big(X_{p4}(g) - X_{p5}(g)\big) \tag{26}$$

Crossover:

$$v_{i,j} = \begin{cases} h_{i,j}(g), \text{rand}(0,1) \le cr \\ x_{i,j}(g), \text{ else} \end{cases} \tag{27}$$

where cr ∈ [0,1] is the crossover probability, taken as $cr = 0.7$.
Select:

$$X_i(g+1) = \begin{cases} V_i(g), f(V_i(g)) < f(X_i(g)) \\ X_i(g), \text{ else} \end{cases} \tag{28}$$

The model parameter settings are shown in Table 1.

Table 1: Differential evolutionary algorithm parameter settings

Parameter	value
Number of populations	100
Crossing rate	0.7
Variation rate	0.85
Allowable error of convergence	10^{-10}
Convergence tolerance judgment number	1000
Maximum allowable number of iterations	30000

When the iteration is over, the better combination of $p_1, p_2, \cdots, p_p$ can be found.

Through the above iterative optimization, we found the optimization results for four parameters and plotted the fitted graphs after the optimization results. The results are:

$$p1 = 26.9884001822349 \tag{29}$$

$$p2 = 0.0444929670892593 \tag{30}$$

$$p3 = -1.90076018991413 \tag{31}$$

$$p4 = 81.9846295969476 \tag{32}$$

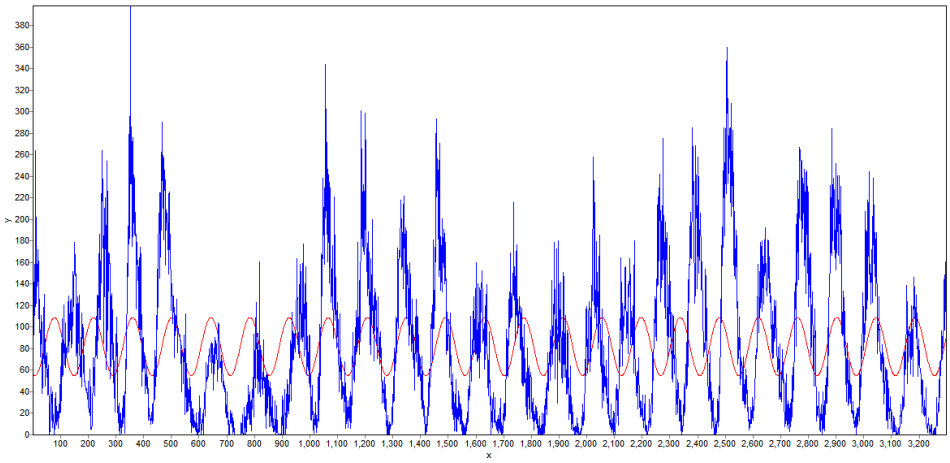

Figure 6: Fitting of regression model results based on differential evolutionary algorithm solution

In Figure 6, the blue line shows the predicted values and the red line shows the fitted values. From the figure, we can analyze the model results solved by this algorithm.

3.3. Hybrid modeling results

The results of the solution are shown in Figure 7:

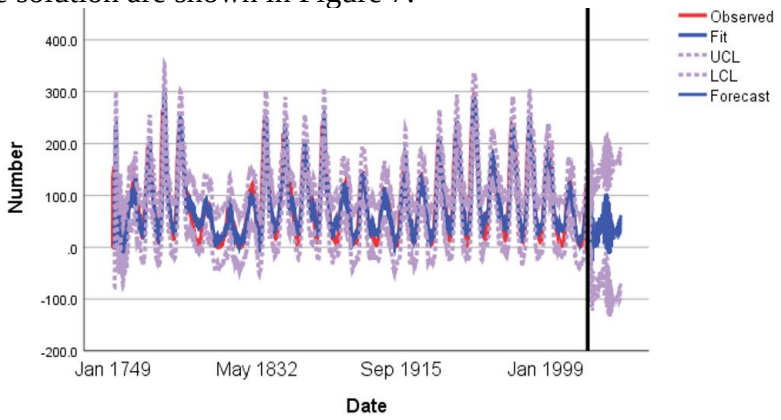

Figure 7: Monthly sunspot totals

Draw a localized map for analysis:

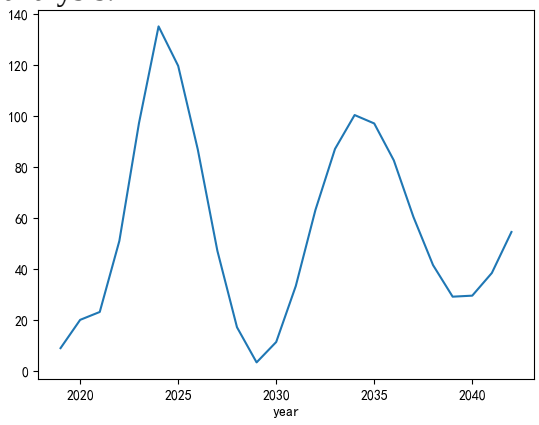

Figure 8: Bureau of Sunspot Quantity Prediction

From Figure 8, it can be seen that the maximum occurs in April 2034, which corresponds to a black volume of 100.5.

4. Conclusions

The aim of this study is to explore the prediction of influencing factors based on the adaptive ARIMA-BP neural network model and the prediction of solar activity based on the adaptive multiple nonlinear regression-BP neural network model. Through the establishment of ARIMA and BP neural network models as well as the prediction model construction of the GABP neural network, we successfully established the adaptive hybrid ARIMA-BP neural network model, which provides a new solution for sunspot prediction.

In Chapter 2, we delve into the ARIMA model building and BP neural network model building. The construction of these models lays a solid foundation for the subsequent prediction models. Through the construction of the prediction model based on the GABP neural network, we combined the neural network technology with the prediction model and achieved satisfactory results. Finally, we built the adaptive hybrid ARIMA-BP neural network model, which provides more accurate prediction results for sunspot prediction.

Chapter 3 focuses on solar activity prediction based on an adaptive multiple nonlinear regression-BP neural network model. We first modeled the relationship between time and the number of sunspots and applied a differential evolutionary algorithm to solve the model. Subsequently, we constructed a BP neural network prediction model and performed hybrid model solving. The successful completion of these steps provides a new perspective and method for solar activity prediction.

Through this study, we not only explored the prediction model of solar activity in depth but also successfully combined the traditional ARIMA model and the BP neural network model, which brought new ideas and methods for sunspot prediction and solar activity prediction. Our results provide useful references and insights for research and practice in related fields and also point out new directions for future research.

In summary, this study has achieved useful results in the field of solar activity prediction and provided new ideas and methods for the improvement and optimization of prediction models. We are satisfied with the research results of the adaptive ARIMA-BP neural network model and adaptive multiple nonlinear regression-BP neural network model, and we are looking forward to future in-depth exploration in this field.

References

[1] J. Yuan, Y. Gao, B. Xie, H. Li, and W. Jiang, "Prediction method of photovoltaic power based on combination of CEEMDAN-SSA-DBN and LSTM," Science and Technology for Energy Transition (STET), vol. 78, 20231.

[2] B. Chen, J. Liu, Z. Ruan, M. Yue, H. Long, and W. Yao, "Freight traffic of civil aviation volume forecast based on hybrid ARIMA-LR model," in International Conference on Smart Transportation and City Engineering (STCE 2022), M. Mikusova, Ed., Chongqing, China: SPIE, Dec. 2022.

[3] S. S. R. Moustafa and S. S. Khodairy, "Comparison of different predictive models and their effectiveness in sunspot number prediction," Physica Scripta, vol. 98, no. 4, 2023, doi: 10.1088/1402-4896/acc21a.

[4] Y. Dang, Z. Chen, H. Li, and H. Shu, "A Comparative Study of non-deep Learning, Deep Learning, and Ensemble Learning Methods for Sunspot Number Prediction," Applied Artificial Intelligence, vol. 36, no. 1, 2022.

[5] A. Tabassum, M. Rabbani, and S. B. Omar, "An approach to study on ma, es, ar for sunspot number (sn) prediction and to forecast sn with seasonal variations along with trend component of time series analysis using moving average (ma) and exponential smoothing (es)," in 1st International Conference on Advances in Electrical and Computer Technologies, ICAECT 2019, April 26, 2019 - April 27, 2019, in Lecture Notes in Electrical Engineering, vol. 672. Coimbatore, India: Springer Science and Business Media Deutschland GmbH, 2020, pp. 373–380.