IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2017.Doi Number*

# Artificial intelligence algorithm for optimal time series data model

**Kang Wang**

School of Humanities and Media, Pingxiang University, Pingxiang Jiangxi, 337055, China.

Corresponding Author: Kang Wang, E-mail: kaobowang@163.com

**ABSTRACT** In order to solve the limitation of a large number of literatures on the study of modeling, simulation and prediction of time series data, there is no model selection, and a certain model is directly used for analysis. For three types of artificial intelligence models often applied to time series analysis: hidden Markov Carrier model, artificial neural network model and autoregressive moving average model are used to study model selection based on simulation comparison method. The study of nonlinear integration methods, using intelligent system methods to learn the weighting mode, has made the model's generalization ability and the degree is of fit to the sample data have been significantly improved. At the same time, numerical simulations are performed on various models, and the characteristics of the time series generated by various models are investigated. Based on the characteristics, the theory and algorithm of model selection are proposed. The model selection theory and algorithm in this paper is used for empirical analysis. For the artificial intelligence models commonly used in time series analysis such as autoregressive moving average model, artificial neural network model, hidden Markov model, etc., when selecting the research model, the method of simulation comparison can be used. The experimental results show that the time series data generated by various models have different mathematical and physical characteristics, which provide a basis for model selection. At the same time, the selection theory is practical. The model selected by the theory has a good fit and prediction effect. The generation of different models has different mathematical characteristics of time series data, which also provides a basis for selecting models.

**INDEX TERMS** Time series; data model; artificial intelligence algorithm; weight pattern; generalization ability

## I. INTRODUCTION

Time series prediction is an interesting and challenging research topic. The motivation of time series prediction is to discover the rules and patterns hidden in the sequence, and to make scientific estimates of the future development trend of the research object based on the learned knowledge. Artificial intelligence has universal approximation ability, and is an ideal rule and pattern learning machine that can be used to develop more advanced forms of predictors. This paper is devoted to improving the accuracy of prediction, and studying the scalability and advantages of artificial intelligence and its combination models on time series prediction problems.

Artificial intelligence technology has been widely used in robot systems, graphic recognition, knowledge engineering, and natural language processing, etc. [[1]. Many intelligent system technologies such as artificial neural network (ANN), support vector machine (SVM), genetic algorithm (GA), and genetic programming have also been widely used in the financial field [2, 3]. Especially the ANN method, in the past many years, ANN has been successfully applied to the research topics of time series analysis in the financial field [4-7]. A large number of studies have shown that feedforward neural networks (BPNN) with a hidden layer can usually approximate any non-linear function with arbitrary accuracy [8-10]. This feature also makes BPNN widely used to predict complex non-linear systems. In addition, compared with traditional models, the superior learning ability of neural networks for dynamic systems also makes it a more powerful tool for studying financial time

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



series. The prediction accuracy of multilayer perceptron (1VB, P), radial basis function network (RBF) and conditional heteroscedasticity model is empirically compared. The results show that in the prediction of exchange rate time series, neural network model and conditional heteroscedasticity model are each of them can give effective predictions [11-14], but it is clear that the overall performance of the neural network is better than the conditional heteroscedastic model. However, neural networks tend to fall into a local minimum in the time series prediction problem. In response to this defect, BPNN and Adaptive Differential Evolution Algorithm (ADE) are combined to present a hybrid model [15-19], namely ADE- BPNN to improve the goodness of fit of sample data for time series analysis. And using two real data sets, the operability and good performance of the proposed hybrid method are confirmed, and the proposed ADE-BPNN method can significantly increase the good fitting performance compared with separate models such as BPNN or ARIMA. Time series forecasting has always been a research problem of interest in many application fields, such as stock price forecasting, temperature forecasting, hydrological time series forecasting, power load forecasting, network traffic forecasting, and so on. Time series prediction is to predict the future data or trends from historical and current data by analyzing the rules or trends of time series over time. Time series prediction methods include classical time series analysis [20-23], neural networks [24-26], and expert systems [27-30]. Time series prediction can be performed by mining time series, discovering sequence rules, and using rule knowledge to predict. An algorithm for discovering sequence rules was proposed. The idea of Apriori algorithm in association analysis was used to mine sequence rules. Three algorithms, AprioriAll, AprioriSome, and DynamicSome, were proposed [31-33]. An algorithm for finding numerical association rules from multiple synchronized data streams is proposed [34]. The clustering method is used to symbolize the time series first, a sequence pattern mining algorithm is used to find the rules in the symbol [35, 36]. An evolutionary rule based on expert system was proposed, which combined fuzzy logic and rule inference for the analysis of stock market activities [37, 38]. Using the methods of fuzzy logic, ANN, and evolutionary computation, the trend of the Nasdaq-100 index value and the Nasdaq-100 index of six other companies were predicted [39]. A time series association rule discovery algorithm based on the cross-correlation succession tree model was proposed. The method of using sequence rules to make predictions is limited by the knowledge of domain experts and has certain limitations [40-42]. In general iterative methods in timing analysis, multi-step estimation is an iteration based on one-step estimation. However, even if the one-step prediction model is very accurate, repeating the iterative process of one-step prediction will accumulate prediction errors, resulting in poor prediction performance. In order to reduce the cumulative error in the iterative process and improve the robustness of the iterative estimation model, the multiple support vector regression (MSVR) model is used in the research of iterative time series analysis [43-46], and it is used in three benchmark data sets. The performance of the model is compared with SVR model and SVR direct model, which proves the validity of the MSVR model. A multi-output support vector regression based on the multi-input output strategy is proposed creatively [47-49], namely M-SVR based on the 1VBM0 strategy, and the effectiveness of the method is simulated with the help of simulated data sets and real data sets. In addition, from the perspective of prediction accuracy and calculation cost, the performance of three SVR models based on different strategies is compared and analyzed. The analysis results show that among the three models compared, the M-SVR based on the 1VBM0 strategy is acceptable. At the cost of computing, the best model accuracy can be achieved in multi-step timing analysis problems. Many application literatures directly apply one or more of the models to directly analyze time series data without establishing a more comprehensive model selection theory. Before establishing a time series model, analyze and compare which data is suitable for use. Class model. In fact, time series data are very different in terms of their own characteristics. For example, in terms of autocorrelation, there are short-term, medium-term, and long-term differences. A model that can only describe short-term correlation is obviously used to analyze time series with long correlation. The data is inappropriate.

When model-based data mining methods are used to process streaming data, the challenging problems are: automatic selection of models, easy updating of models, and training samples that appear in streaming form. In the process of streaming data, data-driven models have a lot of limitations, because the parameters of these models only appear as fitted parameters, without considering the internal mechanism that constitutes the data. Time series data are very different in terms of their own characteristics. For example, in terms of autocorrelation, there are short-term, medium-term, and long-term differences. A model that can only describe short-term correlation is obviously used to analyze time-series data with long correlation suitable. The purpose of this article is to answer what kind of time series the above three models can describe, so as to select the time series data analysis model based on this, and establish a computer intelligent time series model selection method. The selection theory of this model is helpful to guide researchers to better perform data pre-analysis and pre-processing, and improve the efficiency of establishing time series analysis models. Based on computer intelligence, the analysis of time series data is better promoted, making the modeling more targeted and the results more accurate.

## II. CHARACTERISTIC ANALYSIS OF TIME SERIES DATA GENERATED BY VARIOUS MODELS

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



### A. Three types of models

(1) Hidden Markov Model

Hidden Markov models have been successfully applied to speech recognition, DNA sequence analysis, text information extraction, network path analysis, time series analysis and other fields. It is essentially a double random process, consisting of a Markov chain random process (transition process) with hidden states and a random process of generating observations. A complete HMM model consists of 5 parts: a hidden state set (denoted as S), a probability transition matrix between hidden states (denoted as A), a phenotype set of the model output (denoted as V), and the corresponding state to generate these performance Type probability matrix (denoted as s) and the probability distribution of the initial hidden state (denoted as p). The topology is shown in Figure 1.

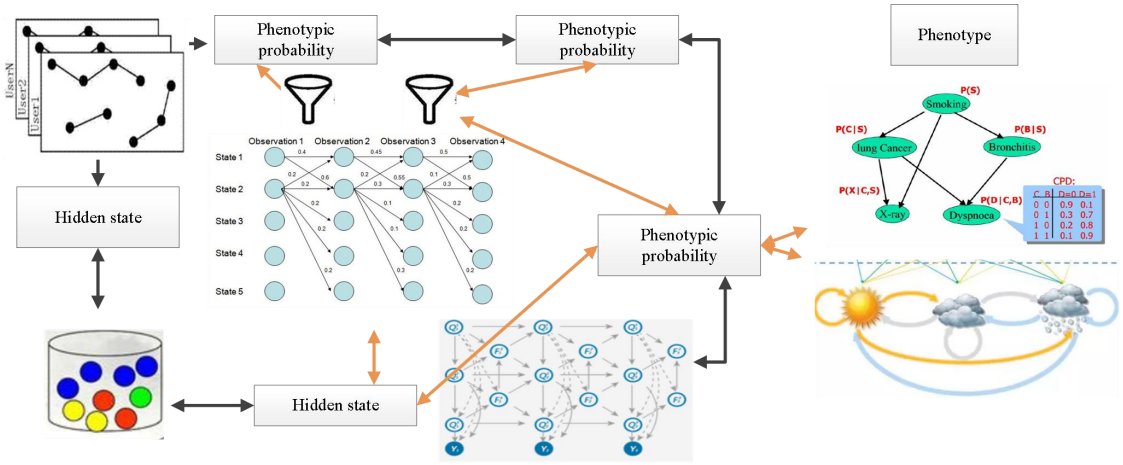


**FIGURE 1** Topological structure of hidden Markov model

(2) Artificial neural network model

Artificial neural networks have developed different network organization forms such as mapping networks, self-organizing networks, recursive networks, and time feedforward networks. Its training algorithm starts from the back propagation algorithm ((BP algorithm), and its basic structure is shown in Figure 2.

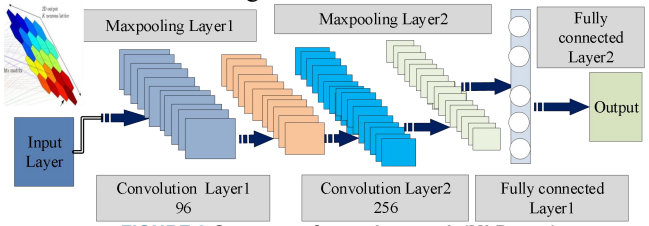


**FIGURE 2** Structure of neural network (MLP type)

Based on the network model, it has been applied to the research of numerical calculation problems, engineering control problems, and signal processing problems, including time series data modeling and prediction problems, and some practical results have been obtained.

(3) Autoregressive moving average model

Auto-regression and moving average (ARMA) is a model specifically designed for trend fitting and prediction of time series data. It was proposed by Box and Jenkins for the modeling of stationary time series. Its model is a combination of an autoregressive model (AR) and a moving average model (MA), which respectively describes the system's memory of past patterns and the memory of past noise. The basic model is given by:

$$y_t = c + \phi_1 y_{t-1} + ...\phi_p y_{t-p} + ...\theta_a \varepsilon_{t-a} \quad (1)$$

Where $\varepsilon_t (t = 1,2...)$ -white noise, $y_t (t = 1,2...)$ a data set for regression analysis (which can be in a multidimensional form), $\phi_i (i = 1,2..p), \theta_j (j = 0,1...q)$ represents the coefficient to be sought. For the calculation of coefficients in this model, the most widely used algorithms are the maximum likelihood estimation method and the moment method.

### B. Characteristic analysis of time series data

The method of simulation analysis is used to confirm the following conjectures about the characteristics of the time series data generated by the three models: Hidden Markov models are short-term related and are not stable; artificial neural networks will use historical information due to the trained algorithm, which is reflected in Among the weights, the results should show long-term correlation, and it is difficult to determine whether it is stable or not. Autoregressive moving average models have been theoretically proven that under certain parameters, it generates stable short-term correlation time series. Since the autoregressive moving average model has theories and a lot of research to reveal its data characteristics, the following is to design the model and verify the conjecture of the data characteristics of the first two models by simulation. Since the autoregressive moving average model has theories and a

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



lot of research to reveal its data characteristics, the following is to design the model and verify the conjecture of the data characteristics of the first two models by simulation.

(1) Features of time series data generated by hidden Markov models

Given a hidden Markov model with 3 hidden states, the transition probability is given by the following matrix:

$$\begin{bmatrix} 0.5, 0.3, 0.2 \\ 0.1, 0.7, 0.2 \\ 0.2, 0.4, 0.4 \end{bmatrix} \tag{2}$$

Let the initial probability distribution be $((0.3,0.3,0.4)^T$, and the probability distribution of the data generated by hidden state 1 satisfy the probability distribution with a mean of 1 and a standard deviation of 0.5, that is, N (-1,0.5). Also define The probability distribution of data generated by hidden state 2 satisfies N (0, 0.1), and the probability distribution of data generated by hidden state 3 satisfies N (1,0.25). A data generated by the above-mentioned hidden Markov model is shown in Figure 3.

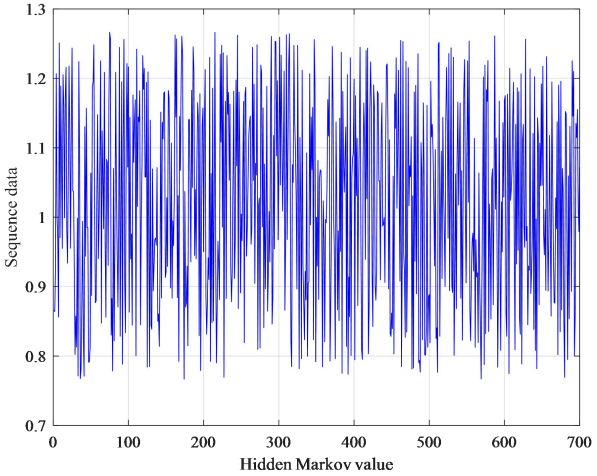


**FIGURE 3 Hidden Markov Model Generates Sequence Data**

(2) Characteristics of time series data by artificial neural network model

Construct a two-layer neural network with 3 input nodes, 2 hidden nodes, and 1 output node. The activation function of the hidden node is a hyperbolic tangent function, and the output node is a proportional linear function. White noise is with distribution N (0, 0.25). The two weight matrices take the following form:

$$w_{2\times3} = \begin{bmatrix} 0.2, -0.45, 0.1 \\ -0.3, 0.2, -0.6 \end{bmatrix}, w_{1\times2} = [2, 4] \tag{3}$$

The initial value is taken as $(0.3\ 0.2\ 0.9)^T$ to obtain the time series data of Figure. 4.

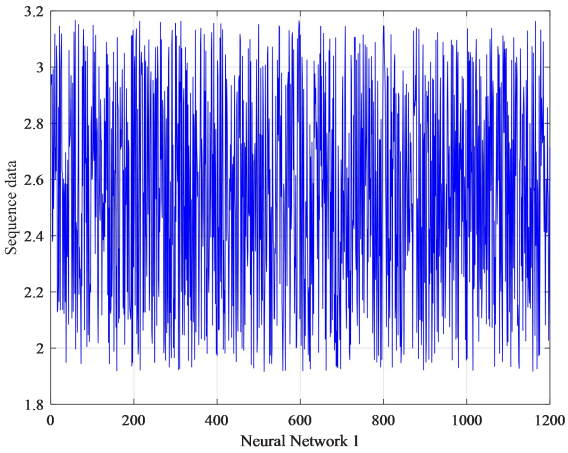


**FIGURE 4 Artificial Neural Network 1 Generates Time Series Data Connections**

Similar to the above analysis process, the autocorrelation, partial correlation analysis and stationary analysis are performed on the time series data, and the following conclusions are obtained: the data generated by the neural network model is long memory, showing long-range autocorrelation, and no attenuation trend. The same KPSS test was applied, which passed the stationary test at a confidence level of 0.1. However, if the above neural network weight matrix is changed to the following form.

$$w_{2\times3} = \begin{bmatrix} 0.2, +0.45, 0.1 \\ +0.3, 0.2, -0.6 \end{bmatrix}, w_{1\times2} = [2, 4] \tag{4}$$

The initial value is also taken as $(0.3\ 0.2\ 0.9)^T$. The connection diagram of the time series is shown in Figure 5.

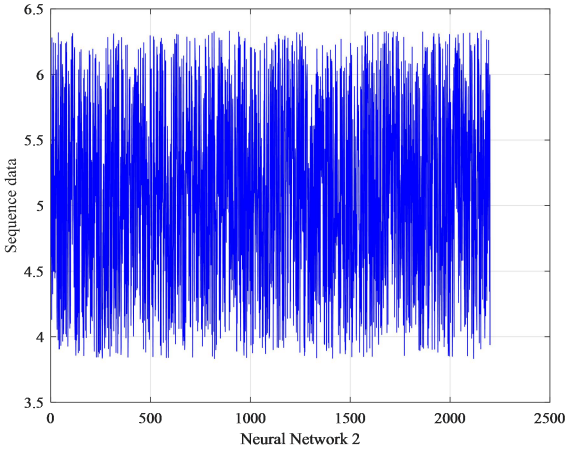


**FIGURE 5 Artificial Neural Network 2 connection for generating time series data**

Its autocorrelation graph and partial correlation graph are similar to the above. It pays attention to the stationarity test. It is found that the KPSS test does not meet the original hypothesis of stationarity at a confidence level of 0.1. It can be seen that the stationarity of the sequence generated by the neural network depends on the network structure.

### *C. Model comparison and model selection theory*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



Through discussion, we can find that the data generated by these three types of models have the following rules. The data generated by Hidden Markov is related to the first period. The KPSS stationarity test is applied, which is not stable. The model generated by the neural network is long-term related. The stability of the generated data depends on the structure of the model; the data generated by the autoregressive moving average model is short-term correlated and meets the stability (under certain parameter conditions). Based on the above description, the model selection theory here is based on the following premise: the data suitable for a model should be consistent with the characteristics of the data generated by the model. And this is a necessary condition, because if a time series data is long-term related, it cannot be generated by a hidden Markov model. If a hidden Markov model is established for such data, there will not be a good modeling effect and prediction. Therefore, the following model selection algorithm is proposed, as shown in Table 1:

TABLE 1 TIME SERIES DATA MODEL SELECTION PROGRAM PSEUDO CODE

(1) Input data;
(2) Perform correlation test and stationarity test on the data;
If the data is a period related then
If the data is not stable then
Select hidden Markov model;
Else if the data is smooth then
Choose an autoregressive moving average model;
End if
Else if the data is short-term related then
If the data is stationary then
Choose an autoregressive moving average model;
Else if data is not stable, then do data transformations, such as difference, into stable data, and apply autoregressive moving average model;
End if
Else if data is a long-term relevant then directly choose neural network model
End if
(3) End the model selection process

## III. Data model for optimal time series

A time series is a set of observations. These observations are the realization of the same variable at different time points in a certain period, such as $X:\{x_1,x_2...x_n\}$. They are arranged in chronological order, and the intervals between time points are equal. The basic idea of time series analysis can be summarized as follows: the ultimate goal is to estimate and infer future time series values, and this goal is to capture the laws behind the historical data of time series and use this law to predict the future [50, 51].

### A. Autoregressive moving average model

The ARMA model, the autoregressive moving average model, is the most widely used stationary time series analysis model. It has three basic forms, namely autoregressive model (AR model), moving average model (MA model), and hybrid model (ARMA model). In essence, the ARMA method is a linear model with limited parameters. The principle framework of stationary time series analysis that meets this condition has been perfected, and it is also widely used in many fields. The ARMA model uses a linear model with limited parameters to characterize the autocorrelation of time series. Not only is it conducive to sequence analysis and structure processing, but the finite parameter one-time model can also describe very common random phenomena, and the accuracy of the actual fitting can meet the needs of reality. In addition, the linear prediction theory can be extended from the structure of the primary model with limited parameters. Therefore, the study of timing analysis problems based on the ARMA model has a theoretically important position in the fields of signal processing, economic prediction, state estimation, control, and pattern recognition.

$$r_i = \varphi_0 + \sum_{i=1,R} \varphi_i r_{i-1} + \sum_{i=1}^{M} \varphi_i \varepsilon_{i-1} + \varepsilon_i \tag{5}$$

$$\varepsilon_i = u_i \sqrt{h_i} \tag{6}$$

$$h_i = k + \sum_{i=1}^{p} A_i \varepsilon_{i-1}^2 + \sum_{i=1}^{q} G_i h_{i-1} \tag{7}$$

The above model is referred to as the $ARMA(R,M)-GARCH(p,q)$ model. Where $r_i$ is the return on assets, $\varepsilon_i$ is the sequence of random error terms, $\varphi_i \phi_i$ is the coefficient term of the conditional mean equation, $h_t$ is the conditional variance, $u_t$ is the sequence of iid random variables, and the mean is 0, the variance is 1, and it is independent of $h_t$, k is the intercept term, $A_i$ is the coefficient of the ARCH term, and $G_i$ is the coefficient of the GARCH term $h_{i-1}^2$.

Based on the ordering theory given by Box and Jenkins, the specific operation of ordering with the help of autocorrelation coefficients and partial autocorrelation coefficients can be summarized as follows: When the ACF is tailed, and the PACF is truncated at the p-th order, consider using the lag order p Autoregressive model, namely AR (p) model. When PACF is tailing and ACF is truncating at q-th order, consider using the MA model and q-th order moving average model. For mixed-order definite order, EACF can be used to judge the lag order p and q of ARMA method.

### B. S-BPNN model

The principle of feedforward neural network can be summarized as follows: At the input layer of the neural network, a large amount of non-linear input information is received by many input network nodes. These input information are called input vectors. The input signal is transmitted, analyzed, and weighed in the neuron connection, and finally the output signal is generated. The hidden layer is all layers composed of all network nodes and connections between the input layer and the output layer. A neural

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



network can contain multiple hidden layers, and the number of nodes or neurons in the hidden layer is more or less, but the larger the number, the more obvious the nonlinearity of the neural network, and the more robust the network model.

For a neural network with a network structure of n-m-l, that is, it contains n input nodes, m hidden nodes, and l output nodes, in the forward propagation process, the input signal first flows into the input layer, and then passes along each hidden layer in turn. Finally, it is transmitted to the output layer, and an output signal is formed at the output end. If the output signal meets a given output requirement, the calculation ends; otherwise, it turns to the second stage of the training process, which is the back propagation error link. The calculation of the forward propagation process is as follows. For the hidden layer, there are:

$$\begin{aligned} net_j &= \sum_{i=1}^{n} v_{ij} x_i + \alpha_j, j = 1,2..m \\ y_j &= f(net_j), j = 1,2...m \end{aligned} \tag{8}$$

For the output layer, there is

$$\begin{aligned} net_k &= \sum_{j=1}^{m} w_{jk} y_j + \beta_k, k = 1,2..l \\ o_k &= f(net_k), k = 1,2...l \end{aligned} \tag{9}$$

The excitation functions corresponding to the hidden layer and the output layer are both unipolar Sigmoid functions, that is $f(x) = \frac{1}{1+e^{-x}}$. The Sigmoid function f (x) is continuously differentiable and $f'(x) = f(x)[1 - f(x)]$. When the actual output of the network model is not consistent with the expected output, Will produce an output error E, which has the following expression:

$$E = \frac{1}{2} \sum_{k=1}^{l} (d_k - o_k)^2 \tag{10}$$

Expanding from the error in the above expression to the hidden layer, there are:

$$E = \frac{1}{2} \sum_{K=1}^{l} [d_k - f(net_k)]^2 \tag{11}$$

The network error is a function of the weight $w_{jk}, v_{ij}$ of each layer, so the magnitude of the error E will change as the weight changes. When correcting the weights, the error should decrease at the fastest speed, so the weights should be corrected along the negative gradient direction of the weight, that is, the correction amount of the weight is proportional to the negative gradient direction of the error, that is,

$$\begin{aligned} \Delta w_{jk} &= -\eta \frac{\partial E}{\partial w_{jk}}, j = 1,2...m; k = 1,2..l \\ \Delta v_{ij} &= -\eta \frac{\partial E}{\partial v_{ij}}, i = 1,2...n; j = 1,2...m \end{aligned} \tag{12}$$

Where $\eta$ is the learning rate, which is a preset constant, usually $0 \prec \eta \prec 1$ is taken.

Suppose there are m p-dimensional vectors $(x_1, x_2 ... x_m)$. The set of all possible linear combinations of these vectors, namely $k_1 x_1 + ... k_m x_m$, forms a linear space, which is called the space of x sheets. Similarly, a vector composed of different lag orders of the original time series can also be expanded into a new vector space, which is called a set of basis that constitutes the prediction space. Proper selection of the basis of the prediction space can help solve many problems encountered in the application of neural network methods. Specifically, an appropriate set of basis can not only capture the potential characteristics of the input variables, but also avoid the computational difficulties caused by the non-uniqueness or multicollinearity of the parameters. Therefore, as a dimensionality reduction technique with the goal of finding several principal components that can explain most of the sample variance, principal component analysis (PCA) becomes a natural choice in this case. In PCA, the original variables are transformed into new variables that are orthogonal to each other, which is very advantageous for simplifying the calculation process, especially in those cases where the original variables are highly correlated. In addition, after the data is reduced in dimension, some noise may be reduced, and more information containing fundamental characteristics is identified, which is beneficial to the next data analysis.

About the first part of the BPNN model, that is, determining the connection weight between the input layer and the hidden layer, the PCA-based solution involves two aspects. First, the initial matrix of weights or loads contains the correlation of all variables and factors. These factor loads represent the degree of agreement between the variables and the principal components. Second, by associating a subset of the original variables with a principal component, the resulting variables will reflect the characteristics of the data formation process.

In addition, mapping raw data into a low-dimensional space can greatly improve the performance of pattern recognition or prediction in a high-dimensional space. Although this mapping may result in loss of information, it is the ultimate goal to build a new set of bases that minimize the number of variables while expanding into the original data space. It is more appropriate to use PCA to achieve this goal, because PCA, as a dimension reduction method, can achieve the purpose of dimension reduction while extracting the main features of the prediction space. In addition, based on the variation of the prediction space explained by each principal component, a natural ordering of the input variables is given, which allows the non-linear function of the original variables to be considered without losing overall degrees of freedom in the parameter estimation process.

The BPNN model also aims to modify the model parameters with the goal of minimizing the sum of squared errors. For testing purposes, an extra Gaussian error structure is assumed during the parameter estimation process:

$$y_t = g(r_t, \theta) + \varepsilon_t \tag{13}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



In the stage of neural network model structure selection, it is necessary to evaluate the contribution of each basic variable from the perspective of explanatory ability. This can be achieved by choosing between two specific model settings, that is, the use of hypothesis testing to determine the optimal number of hidden nodes in the neural network:

$$\begin{aligned} H_0 &: y_t = g(r_t, \theta) + \varepsilon_t \\ H_1 &: y_t = g(r_t, \theta) + \tau h(r_t, \eta) + \varepsilon_t \end{aligned} \tag{14}$$

### C. EMD-LSSVM model

When implementing EMD decomposition, first of all, the following two prerequisites must be met: time series data must meet the following: first, the number of local extreme values of the time series data is the same as the number of zero points or the difference between the two is only 1; The local mean of the sequence is zero, that is, the time series signal is locally symmetric about the time axis, that is, the upper envelope generated by the local maximum and the lower envelope generated by the local minimum, respectively, and the mean of the two is zero. The data sequence x (t) (t = 1, 2… n) can be EMD decomposed according to the following process:

(1) Find all local extreme values in x (t). Use cubic spline interpolation to connect all local maxima to generate the upper envelope $x_u(t)$. Similarly, connect all local minima to produce the lower envelope $x_l(t)$.

(2) Calculate the average envelope value based on the upper and lower envelopes obtained in (1):

$$m_1(t) = \frac{x_u(t) + x_l(t)}{2} \tag{15}$$

(3) The original data sequence x (t) minus the average value of the upper and lower envelopes $m_1(t)$ will generate the first component $d_1(t)$:

$$d_1(t) = x(t) - m_1(t) \tag{16}$$

(4) Check whether $d_1(t)$ meets the requirements of the eigenmode function.

$$\sum_{t=1}^{n} \frac{[d_{k-1}(t) - d_k(t)]^2}{d_{k-1}^2(t)} \tag{17}$$

When the number of training samples is small, the confidence range will increase with the increase of the VC dimension of the learning machine, that is, based on the actual risk, the deviation from the empirical risk will gradually increase. Therefore, choosing a learning machine that is too complex, that is, a VC with too high a neural network, often fails to get good results. This "over-learning" phenomenon mainly occurs because in a small sample situation, once the network structure or algorithm is not designed properly, it will lead to a large confidence range. Even though the risk of experience may be small, the increased confidence will greatly reduce the ability to promote.

In the theoretical framework of the generalized world, a new strategy is proposed in the field of statistical learning, which is different from the empirical risk minimization criterion. Specifically, a function subset sequence is first constructed from a function set, and then the subset sequence is arranged according to the size of the VC dimension. Then based on the empirical risk, find the minimum value in each subset sequence, and finally select the subset with the smallest sum of the minimum empirical risk and confidence range among all the subset sequences. SVM transforms the problem of finding an optimal hyperplane between two different classes into a maximum classification interval problem. The maximum interval problem is actually a quadratic programming problem with inequality constraints.

### D. Nonlinear integrated prediction model

Assuming p training sample set $(x^u, y^u)(u = 1, 2..m)$, our goal is to find the most appropriate functional relationship f = f (x) for prediction. There are n separate prediction techniques that can be used to predict. For any x, the output of the i-th prediction technique is $f_i(x)$. Below we will combine n separate prediction techniques to perform integrated prediction, the general form of which can be expressed as follows:

$$f(x) = \sum_{i=1}^{n} w_i f_i(x) \tag{18}$$

Where f (x) is the integrated prediction result, and the weight of each individual prediction technology in the integrated prediction is $w_i (i = 1, 2..n)$.

The nonlinear integrated prediction model does not simply assign weights to the prediction results of each integrated prediction member, but rather learns the weight pattern by means of artificial intelligence to maximize the information contained in the data after fitting a single prediction model. Reflected in the weights learned, this is better than the linear integration model. To sum up, the nonlinear integrated prediction model has certain advantages over the other models mentioned above in terms of fitting accuracy and generalization ability. Following the above symbolic assumptions, the general steps of nonlinear integrated modeling can be summarized as follows:

① Take the data set: Obtain the expected output d (x) and the prediction results of the individual models constituting the integrated prediction model, that is, $f_i(x), i = 1, 2...n$, constitute the data set of the nonlinear integrated modeling, and divide the data set into two parts, the training set And test set;

② Training weight mode: Use the training set obtained in step ①, and use non-linear technology, such as the artificial intelligence method used in this article, including BPNN and SVM, to train the weight mode of the nonlinear integrated prediction model to determine each individual prediction The weight of technology, that is, $w_i(x), i = 1, 2...n$, and finally determine the optimal model structure;

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



③ Test model performance: Use the test set obtained in step ① and the optimal structure of the model obtained in step ② to test the performance of the nonlinear prediction model to quantify the prediction effect or performance of the nonlinear integrated model.

Generally, we can think of a nonlinear integrated prediction model as a nonlinear information processing system. Assuming that the prediction results of n individual prediction technologies is $y_i, i = 1, 2...n$ , the nonlinear integrated prediction model in this paper can be described by the following formula:

$$y = g(y_1, y_2 ... y_n) \tag{19}$$

Where g is a non-linear function and $y = g(y_1, y_2 ... y_n)$ is the input vector of the model. In the BPNN non-linear integrated prediction model, the weight of the integrated model is determined by the BPNN to realize this non-linear mapping. At this time, the input of the neural network is the prediction result $y_i$ of each individual prediction technology, and the model output is the result of the BPNN nonlinear integrated prediction. The expected output is the corresponding sample true value.

## IV. HMM-based time series artificial intelligence algorithm

Time series are different from static data, whose data change over time. Time series exist in a wide range of fields, from scientific computing, engineering, business, finance, economics, health care to government departments. Cluster analysis of time series has also been extensively studied. These studies include clustering analysis based on original time series, feature-based time series clustering, and model-based time series clustering.

### A. Time series clustering

Common time series clustering algorithms mainly include partitioning (dividing) method and layering method. Partition-based clustering randomly selects k objects as the initial class center, calculates the distance from each object to the class center, and assigns it to the nearest class, then recalculates the new class center, and so on until the object is not Change again. Hierarchical clustering method organizes data objects into a tree. According to whether the hierarchical structure is top-down or bottom-up, the method can be divided into two types: splitting and agglomeration. The split method treats all objects as belonging to the same class, and gradually splits down into more and smaller classes until each object becomes its own class or meets an end condition. The aggregation rule treats each object as an independent class, merging data objects from bottom to top until a certain end condition is met or all objects have been merged into one class. The data sequence space is mapped to the model space, and various existing clustering algorithms are applied in the model space. A partitioning and hierarchical combination clustering algorithm is proposed. Table 2 and Table 3 describe clustering algorithms based on HMM's time series hierarchical clustering, partitioning, and hierarchical combination.

TABLE 2 HIERARCHICAL CLUSTERING ALGORITHM BASED ON HMM

| |
|---|
| Input: $O = \{O_1, O_2 ... O_n\}$ |
| Output: results of clustering |
| Method: |
| 1) Train each sequence $O_i$ as an HMM $\lambda_i$ ; |
| 2) Construct the distance matrix $P(O_i \mid \lambda_i)$ by the likelihood $D = \{D(O_i, O_j)\}$ or the distance between the models; |
| 3) Use agglomerative hierarchical clustering algorithm to cluster by distance matrix D; |

TABLE 3 CLUSTERING ALGORITHM BASED ON HMM-BASED PARTITIONING AND LAYERING

| |
|---|
| Input: $O = \{O_1, O_2 ... O_n\}$ |
| Output: results of clustering |
| Method: |
| 1) Class division: set time series. Divided into k clusters; |
| 2) Train each cluster as a HMM $\lambda_i$ |
| 3) Construct the distance matrix $D = \{D(\lambda_i, \lambda_j)\}$ ; |
| 4) Use agglomerative hierarchical clustering algorithm to cluster by distance matrix D; |

Suppose G and C are data sets with k classes. The similarity measure of clustering is defined as:

$$Sim(G, C) = \frac{1}{k} \sum_{i=1}^{k} \max_{1 \le j \le k} Sim(G_I, C_j) \tag{20}$$

HBHCTS (HMM-Based Hierarchical Clustering Time-Series) algorithm is mainly divided into three parts: the formation of initial partitions, hierarchical aggregation and automatic selection of clustering results. The initial partition is formed by scanning the time series set in a single pass, comparing the currently accessed time series with the existing model (partition). If there is a suitable model, add it, otherwise create a new model. Judging whether the model is suitable is determined by the distance min value. We can obtain relevant prior knowledge by counting the distribution of this threshold, which is easier to determine than specifying the initial number of partitions in advance. After the initial partitions are formed, the hierarchical clustering is used to merge the partitions. The evaluation of the clustering results is similar to the Dunn Index method, and the largest one is the optimal clustering result. The algorithm flowchart is shown in Figure 6:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



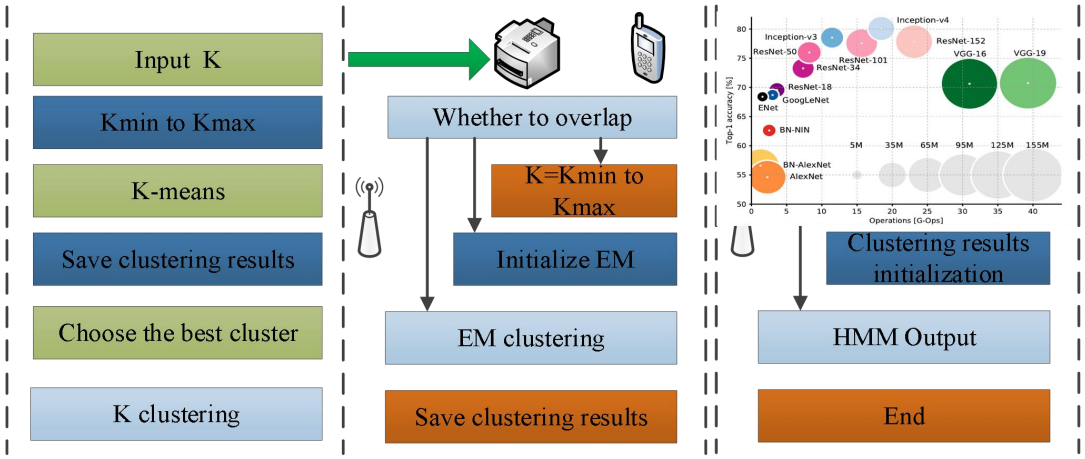


**FIGURE 6** HBHCTS algorithm flowchart

### B. Analysis of Time Series HBHCTS Algorithm

The advantages of the HBHCTS algorithm: (1) no need to specify the initial number of clusters and corresponding initialization; (2) combined with the CBIC method can automatically determine the number of HMM hidden states and corresponding initialization; (3) can clearly give the class's (4) is not sensitive to the length of the sequence; (5) when a new sequence is added, it just compares the new sequence with the existing set of models (models), and does not need to perform partition clustering for all sequences again. It is easy to implement incremental clustering, and initial partition clustering is suitable for time series stream data processing.

The sequences are $O_1 - O_{10}$ in the experiment are from HMM1, $O_{11} - O_{20}$ is from HMM2, and $O_{21} - O_{30}$ is from model 3. Table 4 shows the probabilities of these sequences in HMM 1 and HMM2, respectively:

TABLE 4 EXAMPLES OF PROBABILITIES OF 30 SEQUENCES IN HMM1 AND HMM2

| Sequence | HMM1 | HMM2 |
|---|---|---|
| $O_1$ | 1. 68E-160 | 2. 23E-168 |
| $O_2$ | 1. 66E-159 | 8. 88E-167 |
| $O_3$ | 1. 23E-160 | 1. 44E-167 |
| $O_4$ | 1. 25E-160 | 8. 98E-167 |
| $O_5$ | 1. 53E-159 | 9. 76E-166 |
| $O_6$ | 4. 05E-160 | 3. 91E-168 |
| $O_7$ | 1. 19E-160 | 9. 41E-167 |
| $O_8$ | 1. 13E-160 | 1. 73E-165 |
| $O_9$ | 7. 39E-160 | 1. 06E-165 |
| $O_{10}$ | 2. 05E-159 | 2. 04E-165 |
| $O_{11}$ | 2. 27E-165 | 9. 82E-159 |
| $O_{12}$ | 2. 96E-165 | 5. 06E-161 |
| $O_{13}$ | 9. 63E-169 | 2. 76E-160 |
| $O_{14}$ | 6. 52E-169 | 4. 67E-160 |
| $O_{15}$ | 5. 68E-168 | 7.38E-160 |
| $O_{16}$ | 9. 42E-167 | 2. 63E-160 |
| $O_{17}$ | 3. 72E-167 | 3. 23E-160 |
| $O_{18}$ | 1. 85E-165 | 1. 35E-160 |
| $O_{19}$ | 1. 83E-163 | 6. 55E-159 |
| $O_{20}$ | 1. 48E-167 | 8. 26E-160 |
| $O_{21}$ | 1. 48E-221 | 1. 65E-252 |
| $O_{22}$ | 1. 84E-214 | 5. 08E-216 |
| $O_{23}$ | 1. 91E-212 | 9. 75E-243 |
| $O_{24}$ | 4. 66E-230 | 1. 22E-262 |
| $O_{25}$ | 9.58E-230 | 1. 35E-263 |
| $O_{26}$ | 3. 22E-237 | 1. 84E-271 |
| $O_{27}$ | 1. 31E-213 | 3.84E-255 |
| $O_{28}$ | 3. 90E-242 | 6. 76E-264 |
| $O_{29}$ | 7. 70E-233 | 1. 81E-254 |
| $O_{30}$ | 3. 52E-212 | 1. 17E-255 |

In the experiments on HBHCTS, the HMM hidden state number is determined using two strategies. One uses the specified hidden state number, which ranges from 2 to 8, and is expressed by HBHCTS (2-8). The rate is expressed by HBHCTS (CBIC). For comparison, the experiment was repeated 20 times. The average correct rate of clustering is shown in Figure 7. From Figure 7, we can see that the HBHCTS (CBIC) method can reach the clustering results of the Hier-moHMMs method. In HBHCTS, the HMM hidden state number and initial partition number do not need to be specified in advance, and the Hier-moHMMs method needs to specify the hidden state number and initial partition number in advance, where the HMM hidden state number is specified as 5, and the initial partition number is specified Is 6, the accuracy of clustering is greatly affected by these two parameters.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



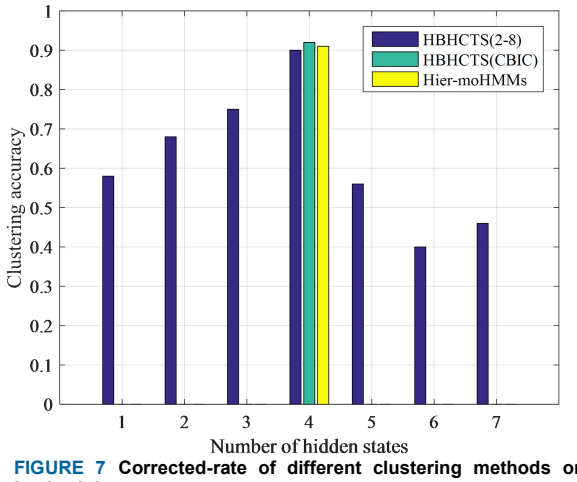


**FIGURE 7 Corrected-rate of different clustering methods on the synthesized data**

In the above experiment, the distance threshold sfit of HBHCTS was set to 0.08. This threshold can be used to decide whether to build a new model for the new sequence. This is achieved by calculating the distance between the sequence and the existing HMM model. If the threshold is set too large, sequences of different classes will be merged by mistake. The smaller the threshold is, the more the number of classes generated in the initial partition is, which increases the computational complexity of subsequent hierarchical clustering.

Experiments show that the distribution has the characteristics of a normal distribution. As shown in Figure 8, the statistical characteristics have a mean value of 0.007, a standard deviation of 0.025, a minimum and maximum value of -3.2 and 3.48, and a kurtosis value of 19.97.

According to the $3\sigma$ rule of normal distribution, the points that fall into the $(u-3\sigma, u+3\sigma)$ interval account for 99% of the entire distribution. Therefore, it can be considered that these points belong to a normal distribution with a 99% confidence level. Therefore, we set the threshold sfit to $3\sigma$ can merge the sequences as much as possible while ensuring the correctness of the initial partitioning of the clustering algorithm.

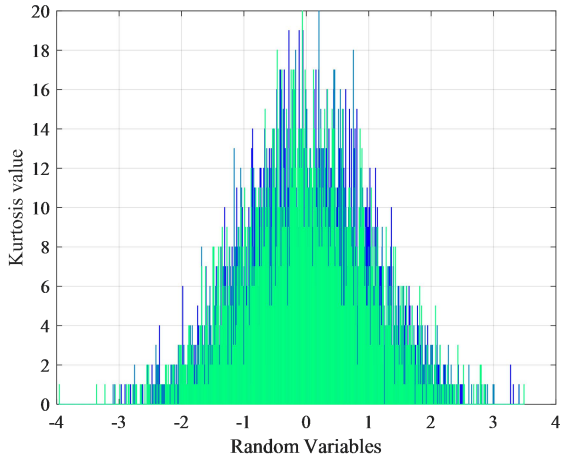


**FIGURE 8 Histogram of random variable x**

Regarding the distribution of the random variable X, we have also experimented with other models. Three models are selected. The first model is a HMM model with five hidden states, where the mean of the hidden state output distribution is randomly set between 0-5 and the variance is randomly set between 0-1. The second model is similar to the first model, except that the variance is randomly set between 0-10. The third model selects the previously mentioned HMM 1 model. Generate a random sequence for these models to count the distribution of the random variable X, calculate the kurtosis value of the random variable X, and repeat 10 random experiments. The experimental results are shown in Table 5. Their kurtosis values are all close to 3.0. It can be seen that, for the sequences of the three models, the random variable x approximately follows a normal distribution.

TABLE 5 KURTOSIS VALUES OF THE RANDOM VARIABLE X OBTAINED FROM DIFFERENT SEQUENCES

| Experiment | Model 1 | Model 2 | Model 3 |
|---|---|---|---|
| 1 | 3. 0466 | 2. 9967 | 3. 0439 |
| 2 | 3. 0712 | 3. 1822 | 3. 1086 |
| 3 | 2. 9072 | 3. 0495 | 3. 0819 |
| 4 | 3. 1131 | 3. 1148 | 3. 0132 |
| 5 | 3. 0039 | 3. 0066 | 3. 0261 |
| 6 | 3. 0049 | 3. 1095 | 3. 0485 |
| 7 | 3. 0661 | 3. 0849 | 2. 9713 |
| 8 | 3. 0333 | 3. 0272 | 2. 9836 |
| 9 | 3. 0983 | 3. 0592 | 2. 9796 |
| 10 | 2. 9786 | 2. 9831 | 3. 0326 |

If in the initial partition, the sequences of different classes are divided into the same region, this introduces wrong partitioning. Because the hierarchical division clustering does not consider class splitting, such as HBHCTS and Hier-moHMMs methods, the initial partition It is important to ensure that sequences belonging to the same class are partitioned in the same region. In order to test the effectiveness of HBHCTS, we have performed experiments on the error rate of the partition sequence of the initial partition of HBHCTS. As shown in Figure 9, as the distance threshold increases, , The initial partition error rate also

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



increased, and after the distance threshold is greater than 0.08, the confidence level of the sequence merging to the same class decreases, resulting in a rapid increase in the initial partition error rate from about 10% to about 30%, which is mainly Because the sequences of HMM 1 and HMM2 are easily divided into the same region. It can be seen from Figure 9 that the number of partitions at this time is about 11. In addition, we can also see from Figure 9 that when the distance threshold is less than 0.08, the initial number of partitions exceeds 10, especially when the distance threshold is 0.03. The number of initial partitions actually exceeds 25. This is because there are many models of Model 3 in which each sequence is divided into a region, and it also shows that the model 3 sequence is the main factor affecting the accuracy of clustering.

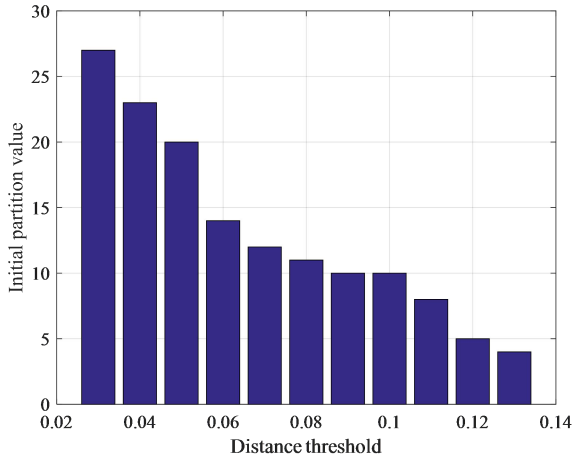


FIGURE 9 **Number of initial partitions at different distance thresholds**

## IV. Experimental verification

In the empirical research in this paper, the models used, ARMA-GARCH model, BPNN model, EMD-SVR model and nonlinear integrated prediction model are all realized by using matlab software developed by Mathworks Experimental Company.

According to the analysis results of autocorrelation analysis, partial autocorrelation analysis, and determination by the AIC criterion, in the ARMA model, the autoregressive order p is set to 4 and the moving average order q is set to 6. The ARCH order and GARCH order of the GARCH model are selected as 3 and 2, respectively. The autocorrelation diagram is shown in Figure 10 below, and Figure 11 is a partial autocorrelation diagram.

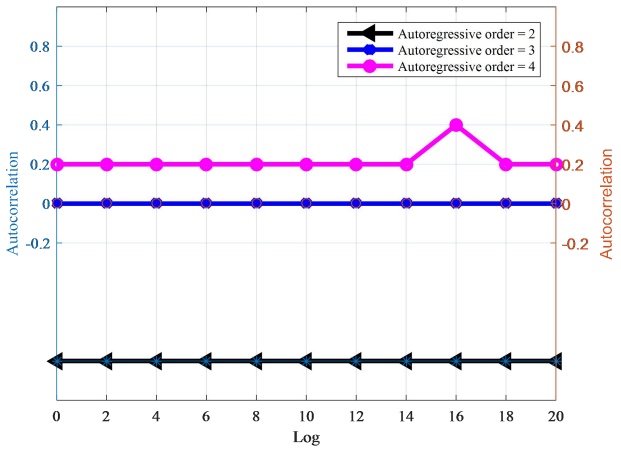


FIGURE 10 **Autocorrelation diagram of time series**

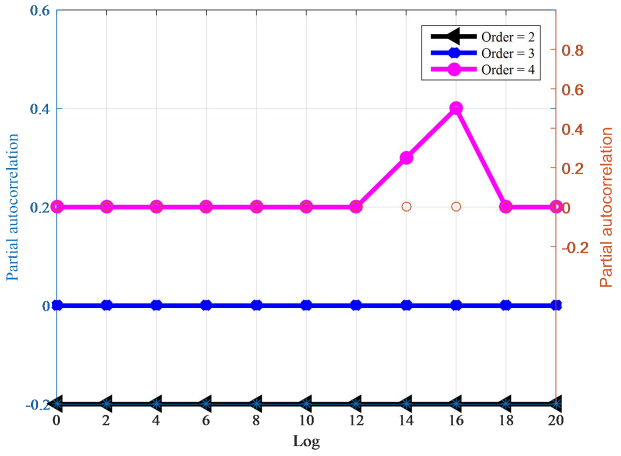


FIGURE 11 **Partial autocorrelation diagram of time series**

In the S-BPNN model, in order to avoid the subjectivity of artificially selecting the number of input nodes, the number of input nodes is set to 6 according to the autocorrelation analysis and partial autocorrelation analysis in the ARMA model modeling process, that is, 6 input variables are selected. PCA was applied to these 6 variables simultaneously, and 6 principal components were obtained, as shown below. The hypothesis test with a confidence level of $\alpha = 99\%$ described in the previous section is used to determine whether a principal component remains in the model. In the test statistic L of the hypothesis test, n = 2264, s = 1, k = 3, $SSE = \sum_{i=1}^{n}(d_i - o_i)^2$ . As a result, the two principal components remain in the network model of conditional mean prediction, in other words, the optimal network structure of the obtained neural network is 6-2-1.

By introducing PCA and hypothesis testing in the process of neural network model selection, the resulting BPNN model has several satisfactory features. First, the BPNN model does not need to make any assumptions about the functional relationship between lagging returns and future returns. Secondly, by orthogonalizing the input space, the possible multicollinearity is eliminated, and the uniqueness of hidden nodes is guaranteed. Third, the step-by-step

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



selection process selects a most streamlined model to ensure that the training data does not overfit. Finally, it reduces the computational cost required to find the best model structure.

The RBF kernel function has two hyperparameters, C and $\gamma$ , namely the penalty factor and the inverse of the Gaussian kernel bandwidth. For a specific problem, naturally, the optimal values of C and $\gamma$ cannot be determined in advance, so it is absolutely necessary to perform model selection, that is, the process of parameter $(C,\gamma)$ search. Choosing the best-performing parameter pair from the many optional parameter pairs is the ultimate goal of model selection, and the best-performing parameter pair refers to the parameter that enables the support vector classifier to make the most accurate prediction of the test data Correct. In the method to achieve this goal, the cross-validation method can avoid the over-fitting phenomenon, and control the variance of the model performance to ensure the stability of the model performance. Cross-validation methods are commonly used to determine tuning parameters and compare model performance.

In the nonlinear integrated prediction model, the first 205 prediction sample values are used as the training set, and the last 50 prediction sample values are used as the test set. Based on simple average integration, the S-BPNN and LSSVM are used to allocate the integrated time series analysis model. Baseline integrated prediction model, the arithmetic average of the results of simple average integrated model. The BPNN and SVM nonlinear integrated prediction models use neural network method and support vector machine technology to determine the weights of three separate prediction models in the integrated model. In addition, considering that the neural network method has the disadvantage of easily falling into a local optimum, when the S-BPNN nonlinear integrated modeling is performed, the average value of the results of running the program 100 times is taken as the final S-BPNN nonlinear integrated prediction model. The performance of the ARMA-GARCH model, the S-BPNN model, the EMD-LS S VM model, and the three integrated prediction models, the simple average integration model, the BPNN integrated time series analysis model, and the SVM integrated time series analysis model are shown in Table 6.

TABLE 6 MODEL PERFORMANCE OF EACH TIME SERIES PREDICTION MODEL

| Evaluation index | NMSE | Rank | $D_{stat}$ | Rank |
|---|---|---|---|---|
| ARMA(4,6)-GARCH(3,2} | 3.2243 | 6 | 48% | 4 |
| S-BPNN | 1.0086 | 5 | 46% | 5 |
| EMD-SSVM | 1.5324 | 4 | 54% | 2 |
| Simple average integrated prediction (benchmark) | 1.2026 | 3 | 51% | 3 |
| Nonlinear Integrated Prediction (BPNN) | 1.1129 | 2 | 52% | 6 |
| Nonlinear Integrated Prediction (SVM) | 0.7384 | 1 | 63% | 1 |

As can be seen from Table 6, three separate prediction models, namely the ARMA (4,6) -GARCH (3,2) model, the S-BPNN model, the EMD-LS S VM model, and three integrated prediction models, namely the simple average The integrated model, BPNN integrated prediction model, and SVM integrated prediction model. The prediction performance of the SVM integrated prediction model is the best among the six models. Not only is the NMSE the smallest, but the $D_{stat}$ is the largest.

In the sample, the difference between the fitted data and the real data is shown in Figure 12, and it can be found that the effect is better, the maximum difference is 0.8, and the average fluctuates within the range of ± 0.8.

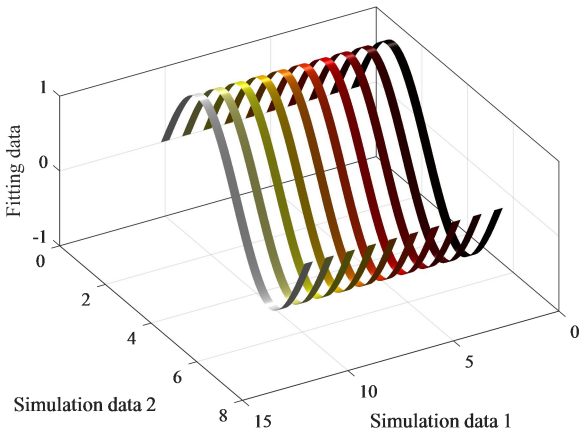


**FIGURE 12 Differences between model data and real data**

The following is a comparison group. Based on the data used as the test set for the fitting set above, the model mechanism for establishing an autoregressive moving average is as follows. First, the model order is determined. Generally, the smaller AIC and BIC are selected as the model according to the AIC and Schwarz criteria the lag order is shown in Table 7 in several cases.

TABLE 7 SELECTION OF LAG ORDER OF ARMA MODEL

| Numbering | AR | MA | AIC | Schwarz |
|---|---|---|---|---|
| 1 | 6 | 3 | -2.813 | -2.661 |
| 2 | 5 | 3 | -2.782 | -2.656 |
| 3 | 4 | 2 | -2.772 | -2.651 |
| 4 | 3 | 2 | -2.749 | -2.690 |
| 5 | 2 | 1 | -2.712 | -2.660 |

The following examines the effect of model extrapolation, and the application model examines the remaining 19 data differences (using real data minus the data generated by the model) as shown in Figure 13.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2020.2981488, IEEE Access



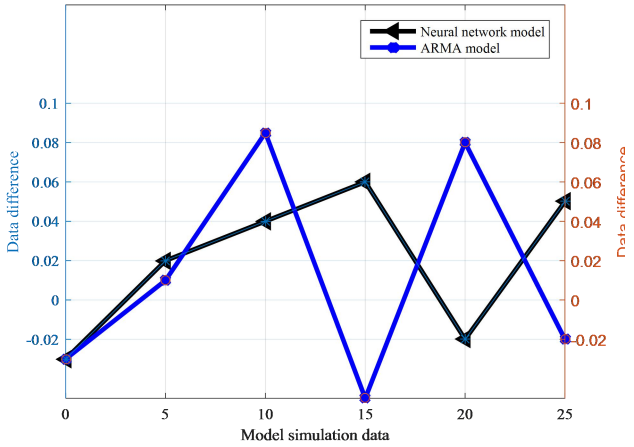


**FIGURE 13 Differences between model data and real data**

In general, the neural network model has a good prediction effect, especially the first 4 predictions, with a small deviation, the magnitude is $10^{-2}$ , and the maximum deviation is 0.06. The maximum difference between the real data and the predicted data of the ARMA model is 0.08, and there are 3 All of them exceed 0.06, and the accuracy of the model fitting and prediction is not as good as the neural network model. By comparing these two models, we can find that the neural network extrapolation prediction has a better effect. From this example, we can prove that the model selection theory in this paper has its rationality.

## V. Conclusion

This paper points out that the three models are suitable for modeling different types of time series data. As a result, the question arises of how to select a suitable model for time series data analysis and prediction. Based on the full analysis of various models, through the pre-analysis and pre-processing of the data, the intelligent technology of the computer is applied to automatically complete the selection process of the corresponding model, forming an artificial intelligence method of model selection. Through empirical analysis of the method, it is found that the method is practical and helps to select a suitable model in a targeted manner to achieve high prediction accuracy. Empirical results show that: from the perspective of normalized mean square error or direction change statistics, the prediction performance of the EMD-LS S VM model is the best among three separate yield time series prediction models; integrated time series The overall performance of the analysis model is better than that of the single model. Among the three integrated time series analysis models, the SVM nonlinear integrated prediction model performs most prominently. In the next step, based on the full analysis of various models, through the pre-analysis and pre-processing of the data, the intelligent technology of the computer is used to automatically complete the selection process of the corresponding model to form an artificial intelligence method of model selection. It helps to select a suitable model in order to achieve high prediction accuracy.

## REFERENCES

[1] Antonio M. Durán-Rosal, Pedro A. Gutiérrez, Salcedo-Sanz S , et al. Dynamical memetization in coral reef optimization algorithms for optimal time series approximation[J]. Progress in Artificial Intelligence, vol.8, pp.253-262, March 2019.

[2] Kewen Li, Lu Liu, Jiannan Zhai. The improved grey model based on particle swarm optimization algorithm for time series prediction:[J]. Engineering Applications of Artificial Intelligence, vol.55, pp.285-291, July 2016.

[3] Aihua Y , Yine C . Data Mining Technology Based Financial Time Series Forecasting Algorithm[J]. Journal of Computational & Theoretical Nanoscience, vol.12, no.12, pp.6296-6303, Dec 2015.

[4] Talaverallames R L , PérezChacón, Rubén, MartínezBallesteros, Maria, et al. A Nearest Neighbours-Based Algorithm for Big Time Series Data Forecasting.[J]. Lecture Notes in Computer Science, vol.14, pp.174-185, April 2016.

[5] Zuhaimy Ismail, Riswan Efendi, Mustafa Mat Deris. Application of Fuzzy Time Series Approach in Electric Load Forecasting[J]. New Mathematics & Natural Computation, vol.11, no.3, pp.229-248, May 2015.

[6] Yongquan Zhou, Rui Wang. An Improved Flower Pollination Algorithm for Optimal Unmanned Undersea Vehicle Path Planning Problem[J]. International Journal of Pattern Recognition & Artificial Intelligence, vol.30, no.4, pp.135-148, Feb 2016.

[7] Weihang S , Nan L . A Joint Classification Learning Algorithm for Feature Sequences of Time-series Data[J]. Computer Engineering, vol.42, no.6, pp.196-200, June 2016.

[8] Wang W , Pedrycz W , Liu X . Time series long-term forecasting model based on information granules and fuzzy clustering[J]. Engineering Applications of Artificial Intelligence, vol.41, pp.1016-1028, May 2015.

[9] Nesrine Baklouti, Ajith Abraham, Adel M. Alimi. A Beta basis function Interval Type-2 Fuzzy Neural Network for time series applications[J]. Engineering Applications of Artificial Intelligence, vol.7, pp.259-274, May 2018.

[10] Tank, Alex, Foti, Nicholas, Fox, Emily. Bayesian Structure Learning for Stationary Time Series[J]. Statistics, vol.24, no.5, pp.70-78, Aug 2015.

[11] Stopar L , Skraba P , Grobelnik M , et al. StreamStory: Exploring Multivariate Time Series on Multiple Scales[J]. IEEE Transactions on Visualization & Computer Graphics, vol.3, pp.78-98, April 2018.

[12] Stuart Edward Lacy, Stephen Leslie Smith, Michael Adam Lones. Using Echo State Networks for Classification : A Case Study in Parkinson's Disease Diagnosis[J]. Artificial Intelligence in Medicine, vol.86, pp.53-59, March 2018.

[13] Yang Zhena, Zhang Wanpeng, Liu Hongfu. Real-time Strategy Game Tactical Recommendation Based on Bayesian Network[J]. Journal of Physics Conference Series, vol. 1168, pp.32018-32029, June 2019.

[14] A. Patra, S. Das, S. N. Mishra. An adaptive local linear optimized radial basis functional neural network model for financial time series prediction[J]. Neural Computing & Applications, vol.28, no.1, pp.101-110, Aug 2015.

[15] Jalalkamali, A, Moradi, M, Moradi, N. Application of several artificial intelligence models and ARIMAX model for forecasting drought using the Standardized Precipitation Index[J]. International Journal of Environmental Science & Technology vol.12, no.4 ,pp.1201-1210, Nov 2014.

[16] Ling Teng, Junwu Zhu, Bin Li. A Voting Aggregation Algorithm for Optimal Social Satisfaction[J]. Mobile Networks & Applications, vol.23, pp.344-351, Sep 2017.
[17] R. Arunkumar, V. Jothiprakash, Kirty Sharma. Artificial Intelligence Techniques for Predicting and Mapping Daily Pan Evaporation[J]. Journal of the Institution of Engineers, vol.98, no.3, pp.219-231, Aug 2017.
[18] Mahmud M S , Meesad P . An innovative recurrent error-based neuro-fuzzy system with momentum for stock price prediction[J]. Soft Computing, vol.20, no.10, pp.4173-4191, June 2015.
[19] Andrzej Janusz, Marek Grzegorowski, Marcin Michalak. Predicting seismic events in coal mines based on underground sensor measurements[J]. Engineering Applications of Artificial Intelligence, vol.64, pp.83-94, Sep 2017.
[20] Mabel González, Christoph Bergmeir, Isaac Triguero. Self-labeling techniques for semi-supervised time series classification: an empirical study[J]. Knowledge & Information Systems, vol.55, pp.493-518, Aug 2017.
[21] Ozoegwu C G . The solar energy assessment methods for Nigeria: The current status, the future directions and a neural time series method[J]. Renewable & Sustainable Energy Reviews, vol.92, pp.146-159, Sep 2018.
[22] Selmo Eduardo Rodrigues Júnior, Ginalber Luiz de Oliveira Serra. A novel intelligent approach for state space evolving forecasting of seasonal time series[J]. Engineering Applications of Artificial Intelligence, vol.64, pp.272-285, Sep 2017.
[23] Diana M. Sánchez-Silva, Héctor G. Acosta-Mesa, Tania Romo-González. Semi-Automatic Analysis for Unidimensional Immunoblot Images to Discriminate Breast Cancer Cases Using Time Series Data Mining[J]. International Journal of Pattern Recognition and Artificial Intelligence, vol.32, no.1, pp.18604-18621, March 2018.
[24] Pritpal Singh. Neuro-Fuzzy Hybridized Model for Seasonal Rainfall Forecasting: A Case Study in Stock Index Forecasting[J]. Studies in Computational Intelligence, vol.611, pp.361-385, Aug 2016.
[25] Xuemin Xing, Debao Wen, Hsing-Chung Chang. Highway Deformation Monitoring Based on an Integrated CRInSAR Algorithm — Simulation and Real Data Validation[J]. International Journal of Pattern Recognition & Artificial Intelligence, vol.32, no.8, pp.185036-185051, May 2018.
[26] Xiao-Xia Yin, Sillas Hadjiloucas, Yanchun Zhang. Pattern identification of biomedical images with time series: Contrasting THz pulse imaging with DCE-MRIs[J]. Artificial Intelligence in Medicine, 2016, vol.67, no.3, pp.1-23, Feb 2016.
[27] Wanjawa B W . Evaluating the Performance of ANN Prediction System at Shanghai Stock Market in the Period [J]. vol.5, no.3, pp.124-145, Dec 2016.
[28] Hirata T , Kuremoto T , Obayashi M , et al. Deep Belief Network Using Reinforcement Learning and Its Applications to Time Series Forecasting[J]. Lecture Notes in Computer Science, 2016, vol.6, no.8, pp.30-37, Sep 2016.
[29] Atencia M, Sandoval F, Prieto A. Advances in computational intelligence: Selected and improved papers of the 12th International Work-Conference on Artificial Neural Networks (IWANN 2013)[J]. vol.164, no.21, pp.1-4, Sep 2015.
[30] Patty Kostkova, Jane Mani-Saada, Gemma Madle. Agent-Based Up-to-date Data Management in National electronic Library for Communicable Disease[J]. Concurrency Practice and Experience, vol.34, pp.105-124, May 2018.
[31] Wen J , Kow Y M , Chen Y . Online games and family ties: Influences of social networking game on family relationship.[J]. Lecture Notes in Computer Science, vol.6948, pp.250-264, Sep 2017.
[32] Felix G , Gonzalo Nápoles, Falcon R , et al. A Review on Methods and Software for Fuzzy Cognitive Maps[J]. Artificial Intelligence Review, vol.52, pp.1707-1737, Aug 2017.
[33] Miquel L. Alomar, Vincent Canals, Nicolas Perez-Mora. FPGA-Based Stochastic Echo State Networks for Time-Series Forecasting[J]. Computational Intelligence & Neuroscience, 2016, vol.2016, pp.892-901, Aug 2016.
[34] Nikolaos Kariotoglou, Maryam Kamgarpour, Tyler H. Summers. The Linear Programming Approach to Reach-Avoid Problems for Markov Decision Processes[J]. Mathematics, 2017, vol.60, pp.263-285, Oct 2016.
[35] Haiyang Yu, Zhihai Wu, Dongwei Chen. Probabilistic Prediction of Bus Headway Using Relevance Vector Machine Regression[J]. IEEE Transactions on Intelligent Transportation Systems, 2017, vol.18, no.7, pp.1772-1781, Jan 2016.
[36] Nema M K , Khare D , Chandniha S K . Application of artificial intelligence to estimate the reference evapotranspiration in sub-humid Doon valley[J]. Applied Water Science, 2017,vol.7, no.5, pp.3903-3910, Mar 2017.
[37] Philipp Grohs, Fabian Hornung, Arnulf Jentzen. A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations[J]. Papers, vol.2, no.2, pp.1314-1325, Sep 2018.
[38] Tawfek Mahmoud, Zhao Yang Dong, Jin Ma. Advanced method for short-term wind power prediction with multiple observation points using extreme learning machines[J]. Journal of Engineering, vol.1, no.1, pp.29-38, Oct 2017.
[39] Priya Nayar, Bhim Singh, Sukumar Mishra. Neural Network based Control of SG based Standalone Generating System with Energy Storage for Power Quality Enhancement[J]. Journal of the Institution of Engineers, 2016, vol.98, no.4, pp.405-413, Sep 2016.
[40] Albrecht S V , Beck J C , Buckeridge D L , et al. Reports on the 2015 AAAI Workshop Series[J]. Ai Magazine, vol.36, no.2, pp.90-101, June 2015.
[41] Shikha Gupta, Nikita Basant, Premanjali Rai. Modeling the binding affinity of structurally diverse industrial chemicals to carbon using the artificial intelligence approaches[J]. Environmental Science & Pollution Research International, vol.22, no.22, pp.17810-17821, July 2015.
[42] Wuhui Chen, Incheon Paik, Zhenni Li. Tology-Aware Optimal Data Placement Algorithm for Network Traffic Optimization[J]. IEEE Transactions on Computers, vol.65, no.8, pp.2603-2617, May 2016.
[43] Raul Cristian Scarlat, Georg Heygster, Leif Toudal Pedersen. Experiences With an Optimal Estimation Algorithm for Surface and Atmospheric Parameter Retrieval From Passive Microwave Data in the Arctic[J]. IEEE Journal of Selected Topics in Applied Earth Observations & Remote Sensing, vol.3, pp.1-14, Sep 2017.
[44] Mohammad Altaher, Omaima Nomir. Euler–Lagrange as Pseudo-metric of the RRT algorithm for optimal-time trajectory of flight simulation model in high-density obstacle environment[J]. Robotica, vol.5, pp.1-19, Dec 2015.
[45] Ankita Sinha, Prasanta K. Jana. MRF: MapReduce based Forecasting Algorithm for Time Series Data[J]. Procedia Computer Science, vol.132, pp.92-102, Dec 2018.
[46] Goran Klepac, Robert Kopal, Leo Mršić. REFII Model as a Base for Data Mining Techniques Hybridization with Purpose of Time Series Pattern Recognition[J].

Studies in Computational Intelligence, vol.611, pp.237-270, Dec 2016.

[47] Wang T , Xia Y , Muppala J , et al. Achieving Energy Efficiency in Data Centers Using an Artificial Intelligence Abstraction Model[J]. IEEE Transactions on Cloud Computing, vol.5, pp.35-48, Dec 2015.

[48] Adit B. Sanghvi, Erastus Z. Allen, Keith M. Callenberg. Performance of an artificial intelligence algorithm for reporting urine cytopathology[J]. Cancer Cytopathology, vol.127, no.10, pp.241-154, Aug 2019.

[49] Chen, Xin, Xing, Pei, Luo, Yong. Surface temperature dataset for North America obtained by application of optimal interpolation algorithm merging tree-ring chronologies and climate model output[J]. Theoretical & Applied Climatology, vol.127, pp.533-549, Oct 2015.

[50] Zhang Qi, Hu Yupeng, Ji Cun. Edge Computing Application:Real-Time Anomaly Detection Algorithm for Sensing Data[J]. Journal of Computer Research & Development, vol.3, pp.1009-1021, Jan 2018.

[51] H. Zheng, H. Li, J. Xiao. Research on model and algorithm for optimal power flow of large-scale high voltage direct current transmission system[J]. Zhongguo Dianji Gongcheng Xuebao/proceedings of the Chinese Society of Electrical Engineering, vol.35, no.9, pp.2162-2169, Sep 2015.

[52] P. Sengottuvelan, N. Prasath. BAFSA: Breeding Artificial Fish Swarm Algorithm for Optimal Cluster Head Selection in Wireless Sensor Networks[J]. Wireless Personal Communications, vol.94, no.4, pp.1979-1991, April 2016.

**Kang Wang**, Graduated from Wuhan University in 2016. Currently works at the School of Humanities and Media, Pingxiang University. His research interests include Cognitive Science, Philosophy of Mind and Artificial Intelligence.