# Air Combat Maneuver Trajectory Prediction Model of Target Based on Chaotic Theory and IGA-VNN 

Zhi-fei Xi ${ }^{(1)}$, An Xu ${ }^{(\mathbb{D}}$, Ying-xin Kou, Zhan-wu Li, and Ai-wu Yang<br>Air Force Engineering University, Aeronautics Engineering College, Xi'an 710038, China<br>Correspondence should be addressed to An Xu; 18292453682@163.com

Received 21 May 2020; Revised 3 August 2020; Accepted 22 October 2020; Published 29 November 2020
Academic Editor: Elisa Francomano
Copyright © 2020 Zhi-fei Xi et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.


#### Abstract

Target maneuver trajectory prediction is an important prerequisite for air combat situation awareness and threat assessment. Aiming at the problem of low prediction accuracy in traditional trajectory prediction methods, combined with the chaotic characteristics of the target maneuver trajectory time series, a target maneuver trajectory prediction model based on chaotic theory and improved genetic algorithm-Volterra neural network (IGA-VNN) model is proposed, mathematically deducing and analyzing the consistency between Volterra functional model and back propagation (BP) neural network in structure. Firstly, the C-C method is used to reconstruct the phase space of the target trajectory time series, and the maximum Lyapunov exponent of the time series of the target maneuver trajectory is calculated. It is proved that the time series of the target maneuver trajectory has chaotic characteristics, so the chaotic method can be used to predict the target trajectory time series. Then, the practicable Volterra functional model and BP neural network are combined together, learning the advantages of both and overcoming the difficulty in obtaining the high-order kernel function of the Volterra functional model. At the same time, an adaptive crossover mutation operator and a combination mutation operator based on the difference degree of gene segments are proposed to improve the traditional genetic algorithm; the improved genetic algorithm is used to optimize BP neural network, and the optimal initial weights and thresholds are obtained. Finally, the IGA-VNN model of chaotic time series is applied to the prediction of target maneuver trajectory time series, and the experimental results show that its estimated performance is obviously superior to other prediction algorithms.


## 1. Introduction

Maneuvering trajectory prediction is a process of learning and reasoning the inherent information contained in the target's historical trajectory and then making a reasonable prediction of the target's future trajectory. In modern air combat, it is of great significance to make a reasonable prediction of the maneuvering trajectory of enemy targets. According to the OODA (observation, orientation, decision, and action) cycle theory, the essence of winning an air is to form an OODA cycle prior to the enemy and accurately predict the target maneuver trajectory so as to achieve the purpose of preemption [1].

In recent years, the research on target maneuver trajectory prediction methods can be divided into two categories. One is the traditional method based on Kalman filter
algorithm, $(\alpha / \beta)$ filter algorithm, linear regression model, and particle motion model. In view of the complex and changeable characteristics of target motion patterns, a polynomial Kalman filter for the motion trajectory prediction algorithm is proposed for the change of the target motion mode [2]. Aiming at the problem of missing historical position information of the target, an improved Kalman filter algorithm with system noise estimation is proposed to predict the target maneuver trajectory [3]. Owing to general traditional fitting-based trajectory prediction algorithms cannot meet the requirements of high accuracy and real-time prediction, a dynamic Kalman filter for trajectory prediction approach is proposed [4]. In order to solve the problems of high nonlinear, difficult data processing, and low prediction accuracy in the high-order motion model of the target, an interactive multimodel
trajectory prediction algorithm with control variables is proposed [5]. According to the above analysis, the traditional prediction method is only applicable to the relatively simple trajectory prediction of the target motion characteristics. However, in the process of air combat, the target motion is often a highly nonlinear and complex time sequence process, which is also affected by many uncertain factors. The traditional trajectory prediction algorithms cannot reflect the target's maneuvering characteristics very well. In addition, the complexity of the trajectory prediction model is high, the adaptability of the prediction algorithm is poor, and the prediction accuracy cannot meet the requirements of air combat.

The other algorithm is a machine learning prediction algorithm based on intelligent algorithms, which are mainly based on big data to establish a target maneuvering trajectory prediction model. In view of GRNN's (generalized regression neural network) good nonlinear mapping ability, flexible network structure, and high fault tolerance and robustness, a trajectory prediction model based on GRNN neural network is proposed [6]; the target group trajectory is used to train the BP neural network, establishes a trajectory prediction model, and realizes the prediction of the flight trajectory [7]. However, the BP neural network has the disadvantages of being sensitive to the initial value and not having the global search ability. In order to solve these problems, the genetic algorithm and particle swarm optimization algorithm with global search ability are, respectively, used to optimize the neural network weights to improve the prediction accuracy [8, 9]. In addition, the essence of target maneuver trajectory prediction is the time series prediction problem, which is highly nonlinear and time-varying. The theory and practice show that the Volterra functional model can well represent the nonlinear system and be used to accurately predict the time series. However, the solution of the kernel function of the higher-order Volterra functional model is the bottleneck of its application. Due to the good adaptability, parallelism and fault tolerance of BP neural network, and the ability to approximate any nonlinear function, the Volterra functional model and BP neural network have the same structure [10-12]. Therefore, in this paper, a target maneuver trajectory prediction model based on BP neural network and Volterra series is proposed. In this paper, the theory of phase space reconstruction is used to reconstruct the time series of the target's maneuvering trajectory, and the small data method is used to prove that the time series of the target's maneuvering trajectory has chaotic characteristics. Then, the mathematical equivalence between the BP neural network model and the Volterra functional model is proved by theoretical analysis. Secondly, combining the chaotic characteristics of the target maneuver trajectory time series, an IGA-VNN target maneuver trajectory time series prediction model based on chaos theory is established. The model combines the advantages of the Volterra functional model and BP neural network and overcomes the difficulty of the Volterra series model in determining higher-order kernel function and the blindness of BP neural network modeling, and an improved genetic algorithm is proposed to optimize the weights and
thresholds of VNN networks simultaneously. Finally, the performance of the prediction model proposed in this paper is verified by simulation. The simulation results show that the model can accurately and rapidly predict the maneuvering trajectory of the target, which provides a new way to solve the problem of trajectory prediction.

## 2. Dynamics Theory of Chaotic System

2.1. Phase Space Reconstruction Theory. PSRT (phase space reconstruction theory) is an effective method to analyze nonlinear time series, and it is the basis for determining and predicting chaotic characteristics of time series. The basic idea of phase space reconstruction theory is to regard time series as the component generated by nonlinear dynamic series. According to Takens theorem [13, 14], for a chaotic time series, when the embedding dimension $m$ and attractor dimension $D$ of the phase space reconstruction meet the condition of $m \geq 2 D+1$, the attractor of the original time series can be replaced in the reconstructed phase space. At the same time, the time series after phase space reconstruction is topologically equivalent to the original system on the basis of differential homeomorphism [15]. If there is a time series $\{x(i), i=1,2, \ldots, N\}$, the time series reconstructed by phase space can be expressed as follows:

$$
\begin{align*}
\mathbf{X} & =\left[\begin{array}{llll}
\mathbf{X}_{1} & \mathbf{X}_{2} & \cdots & \mathbf{X}_{M}
\end{array}\right]^{T} \\
& =\left[\begin{array}{cccc}
x(1) & x(2) & \cdots & x(M) \\
x(1+\tau) & x(2+\tau) & \cdots & x(M+\tau) \\
\vdots & \vdots & \ddots & \vdots \\
x[1+(m-1) \tau] & x[2+(m-1) \tau] & \cdots & x[M+(m-1) \tau]
\end{array}\right] \tag{1}
\end{align*}
$$

where $N$ is the length of the time series; $m$ is the embedding dimension; $\tau$ is the delay time; and $M=N-(m-1) \tau$.

In the process of reconstructing phase space, the determination of time delay and embedding dimension plays a crucial role in the quality of phase space reconstruction. If the embedding dimension is too low, the self-interaction of attractors will appear; if the parameter is too high, the distance between points will be too far. If the time delay is too small, the correlation between the adjacent points of the reconstructed attractor is too strong, and the analysis of the attractor is easily interfered by noise; if the parameter is too large, the originally close vectors will also be far away, resulting in an uncertain system state [16]. Therefore, it is very important to determine reasonable time delay and embedding dimension for phase space reconstruction.
2.2. C-CMethod. The traditional calculation method regards the embedding dimension and time delay as two independent parameters, which need to be calculated independently. The research in recent years shows that the embedding dimension and time delay are interrelated. The determination of time delay should not be independent of the embedding dimension but should be determined by combining the time window $\tau_{w}=(m-1) \tau$.

In this paper, the $C-C$ method [15] is used to determine the time delay and embedding dimension. The C-C method uses the correlation integral of time series to form statistics, which reflects the correlation of nonlinear time series. Based on the relationship between time delay and statistics, the time window and time delay can be calculated simultaneously so that the embedding dimension can be determined. The correlation integral is defined as follows:

$$
\begin{gather*}
C(m, r, t)=\frac{2}{M(M-1)} \sum_{1 \leq i \leq j \leq M} H\left(r-d_{i j}\right),  \tag{2}\\
d_{i j}=\left\|X_{i}-X_{j}\right\|
\end{gather*}
$$

where $M$ is the number of phase points; $r$ is the size of neighborhood radius; $d_{i j}$ is the Euclidean distance between two phase points in phase space; and $H(z)$ is the Heaviside unit function, which is defined as follows:

$$
H(z)= \begin{cases}1, & z>0  \tag{3}\\ 0, & z \leq 0\end{cases}
$$

Dividing time series $\{x(i), i=1,2, \ldots, N\}$ into $t$ independent subtime series, the length of the subtime series is $\operatorname{INT}(N / t)$, where INT is an integer function. For $t$ disjoint subtime series, the test statistic $S(m, N, r, t)$ and $\Delta S(m, t)$ can be expressed as follows:

$$
\begin{align*}
S(m, N, r, t) & =\frac{1}{t} \sum_{s=1}^{t}\left[C_{s}\left(m, \frac{N}{t}, r, t\right)-C_{s}^{m}\left(1, \frac{N}{t}, r, t\right)\right], \\
\bar{S} & =\frac{1}{M \cdot J} \sum_{m=1}^{M} \sum_{j=1}^{J} S\left(m, r_{j}, t\right), \tag{4}
\end{align*}
$$

where $J$ is the number of $r$.
In order to express the relationship between $S(m, N, r, t)$ and $r$, the difference $\Delta S(m, t)$ between the maximum and minimum values of the two radius is expressed as follows:

$$
\begin{equation*}
\Delta S(m, t)=\max \left\{S\left(m, r_{j}, t\right)\right\}-\min \left\{S\left(m, r_{j}, t\right)\right\} \tag{5}
\end{equation*}
$$

According to Brock mathematical statistical conclusions [16], in general, we can obtain the range values of $m$ and $r$ as $m=2,3,4,5, r_{j}=(j \sigma / 2)$, and $j=1,2,3,4$, where $\sigma$ is the standard deviation of the time series. The three test statistical variables are defined as follows:

$$
\begin{align*}
\bar{S}(t) & =\frac{1}{16} \sum_{m=2}^{5} \sum_{j=1}^{4} S\left(m, r_{j}, t\right) \\
\Delta S(t) & =\frac{1}{4} \sum_{m=2}^{5} \Delta S(m, t)  \tag{6}\\
S_{\text {cor }}(t) & =\Delta \bar{S}(t)+|\bar{S}(t)|
\end{align*}
$$

where the time delay $\tau$ is the time corresponding to the first minimum of $\Delta \bar{S}(t)$ or the time corresponding to the first zero of $\bar{S}(t)$. When $S_{\text {cor }}(t)$ obtains the minimum value, the corresponding time is the time window $\tau_{w}$. The C-C method uses statistical theory to obtain the time delay and embedding
dimension. The calculation is simple, the calculation amount is small, and it has strong anti-interference ability.

Based on the theoretical analysis of PSR in Section 2.1, the C-C method is used to reconstruct the phase space of the three-dimensional coordinate time series of the target maneuver trajectory. The test statistics $\Delta \bar{S}(t), \bar{S}(t)$, and $S_{\text {cor }}(t)$ calculated by the C-C method are shown in Figures 1-3. It can be seen from Figure 1 that the first minimum point of $\Delta \bar{S}(t)$ is $\tau_{x}=19$, and the minimum point of $S_{\text {cor }}(t)$ is $\tau_{w}^{x}=126$. Then, based on the embedded time window equation $\tau_{w}=(m-1) \tau$, the optimal embedded dimension is $m_{\text {opt }}^{x}=7$. As can be seen from Figure 2, the first minimum point of $\Delta \bar{S}(t)$ is $\tau_{y}=25$, and the minimum point of $S_{\text {cor }}(t)$ is $\tau_{w}^{y}=181$. Then, based on the embedded time window equation $\tau_{w}=(m-1) \tau$, the optimal embedded dimension is $m_{y}^{\mathrm{opt}}=8$. Similarly, as can be seen from Figure 3, the first minimum point of $\Delta \bar{S}(t)$ is $\tau_{z}=21$, and the minimum point of $S_{\text {cor }}(t)$ is $\tau_{w}^{z}=157$. Then, based on the embedded time window equation $\tau_{w}=(m-1) \tau$, the optimal embedded dimension is $m_{\mathrm{opt}}^{z}=8$. The determination of the above parameters provides a basis for proving that the target maneuver trajectory prediction system has chaotic characteristics.
2.3. Calculation of Maximum Lyapunov Exponent. Butterfly effect is the basic form of the chaos system; that is, chaos is sensitive to the initial value. The trajectories generated by two initial values with little difference will show exponential separation with time. Lyapunov exponent is the eigenvalue of the chaotic system after averaging on the trajectories of the whole attractor [14]. Among them, the largest Lyapunov index can quantitatively describe the separation of orbits generated by two very close initial values as time goes by. It is theoretically proved in [17] that as long as the maximum Lyapunov exponent $\lambda$ satisfies $\lambda>0$, the system has chaotic characteristics.

The small data is one of the most commonly used methods to calculate the maximum Lyapunov exponent of the chaotic system. The specific calculation steps are as follows:

Step 1: perform fast Fourier transform on the time series $\{x(i), i=1,2, \ldots, N\}$ to determine the average period $P$.
Step 2: use the C-C method to determine time delay $\tau$ and embedding dimension $m$ and reconstruct phase space of time series $\left\{Y_{j} \mid j=1,2, \ldots, M\right\}$.
Step 3: find the closest phase point $Y \widehat{\jmath}$ to each phase point $Y_{j}$ in the phase space after time series reconstruction and limit the short-term separation, namely:

$$
\begin{equation*}
d_{j}(0)=\min \left\|Y_{j}-Y_{\hat{j}}\right\|, \quad|j-\widehat{j}|>p \tag{7}
\end{equation*}
$$

Step 4: for each phase point $Y_{j}$ in the phase space, calculate the distance $d_{j}(i)$ after $j$ discrete-time steps of the adjacent point pair:


Figure 1: Statistics curves of the C-C method to reconstruct the $X$ direction trajectory time series.


Figure 2: Statistics curves of the C-C method to reconstruct the $Y$ direction trajectory time series.

$$
\begin{equation*}
d_{j}(i)=\min \left\|Y_{j+i}-Y_{\widehat{j}+i}\right\|, \quad i=1,2, \ldots, \min (M-j, M-\widehat{j}) . \tag{8}
\end{equation*}
$$

Step 5: for each discrete time step $i$, calculate the average $y(j)$ of all $\ln d_{j}(i)$ :

$$
\begin{equation*}
y(j)=\frac{1}{q h} \sum_{i=1}^{q} \ln d_{j}(i) \tag{9}
\end{equation*}
$$



Figure 3: Statistics curves of the C-C method to reconstruct the Zdirection trajectory time series.
where $q$ is the number that $d_{j}(i)$ is nonzero. The least square method is used to make the regression line, and the slope of the line is the largest Lyapunov exponent $\lambda$. This method makes full use of the spatial evolution information of time series, which is more reliable for small data groups, less computation, and easy to operate.

Based on the theoretical analysis of the small data method in Section 2.3, the small data method is used to analyze the chaotic characteristics of the three-dimensional coordinate time series of the target maneuver trajectory. The least square fitting curve of the target maneuver trajectory time series and the corresponding slope changes are shown in Figures 4-6. Based on the small data method, the maximum Lyapunov exponent of the target maneuver trajectory in $X$-direction time series, $Y$-direction time series, and $Z$ direction time series is calculated as $L_{y}^{X}=0.0011$, $L_{y}^{Y}=0.0042$, and $L_{y}^{Z}=0.004682$, respectively. It can be seen from the above that the maximum Lyapunov exponents of the three-dimensional coordinate time series of the target's maneuvering trajectory are all positive numbers, which shows that the time series of the target's maneuvering trajectory has chaotic characteristics.

## 3. Volterra Functional Model of Nonlinear System

The Volterra filter is usually used in signal processing and model recognition of the nonlinear system [18], and the nonlinear expression ability is good, so the Volterra functional model can realize accurate prediction of chaotic time series. For nonlinear systems, the discretized Volterra functional model can be expressed as follows:


Figure 4: Fitted curve by the least square method and the corresponding slope changes of the $X$-direction trajectory time series.


Figure 5: Fitted curve by the least square method and the corresponding slope changes of the $Y$-direction trajectory time series.

$$
\begin{equation*}
y(k)=\sum_{i=1}^{k} \sum_{l_{1}, l_{2}, \ldots, l_{i}}^{\infty} h_{i}\left(l_{1}, l_{2}, \ldots, l_{i}\right) \cdot x\left(k-l_{1}\right) \cdot x\left(k-l_{2}\right) \cdots x\left(k-l_{i}\right) \tag{10}
\end{equation*}
$$

where $i, l_{i}, k \in R ; x\left(k-l_{i}\right)$ is the input of the system; $y(k)$ is the output of the system, which is the state of the system at the next moment; and $h_{i}\left(l_{1}, l_{2}, \ldots, l_{i}\right)$ is the Volterra kernel function of order $i$.

In theory, the discrete Volterra functional model can accurately predict the nonlinear time series. When the Volterra filter is used to predict the time series, because it is difficult for the functional model to solve the higher-order kernel function, the second-order or third-order truncation form of the Volterra functional model will be used in


Figure 6: Fitted curve by the least square method and the corresponding slope changes of the $Z$-direction trajectory time series.
general, but this will greatly reduce the prediction accuracy and performance of the model [19-21]. In addition, the Volterra functional model has limited memory ability. When time point $k-k_{0}$ is far away from the time point $k$, the input $x\left(k-k_{0}\right)$ will not affect the output. Based on the above analysis, the practical Volterra functional model adopted in this paper is as follows:

$$
\begin{align*}
y(k)= & h_{0}+\sum_{l_{1}=0}^{m-1} h_{1}\left(l_{1}\right) x\left(k-l_{1} \tau\right) \\
& +\sum_{l_{1}=0}^{m-1} \sum_{l_{2}=0}^{m-1} h_{2}\left(l_{1}, l_{2}\right) x\left(k-l_{1} \tau\right) x\left(k-l_{2} \tau\right)+\cdots+\sum_{l_{1}=0}^{m-1} \sum_{l_{2}=0}^{m-1} \cdots \\
& +\sum_{l_{m}=0}^{m-1} h_{m}\left(l_{1}, l_{2}, \ldots, l_{m}\right) x\left(k-l_{1} \tau\right) x\left(k-l_{2} \tau\right) \cdots x\left(k-l_{m} \tau\right), \tag{11}
\end{align*}
$$

where $m$ is the memory length of the Volterra functional system, that is, the minimum embedding space dimension of the phase space reconstruction of the target maneuver trajectory time series, and $\tau$ is the time delay.

Based on equation (11), it is possible to accurately predict the target maneuver trajectory. Through the above analysis, we can know that different time series have different characteristics, and the time delay and embedding dimension determined by the C-C method are also different. Therefore, the Volterra functional series prediction model used in the prediction of the three-dimensional coordinates of the target's maneuvering trajectory is different.

## 4. Prediction Model of Chaotic Time Series Based on VNN

4.1. Model Equivalence Analysis. The $m$ dimensional input, single hidden layer, and single output BP neural network are established, and its structure is shown in Figure 7. The input vector of the BP neural network is $\mathbf{X}^{T}(k)=\left[x_{k, 0}, x_{k, 1}, \ldots, x_{k, n}, \ldots, x_{k, m-1}\right]$, and the input is
obtained by $x(k)$ through phase space reconstruction theory, where $x_{k, m}=x(k-m)$. The output of the $l$ th unit of the hidden layer is as follows:

$$
\begin{equation*}
Z_{l, k}=S_{l}\left(U_{l, k}\right), \quad l=1,2, \ldots, L . \tag{12}
\end{equation*}
$$

The discrete convolution $U_{l, k}$ of system input time series is expressed by

$$
\begin{equation*}
U_{l, k}=\sum_{n=0}^{m-1} w_{l, n} x_{k, n} . \tag{13}
\end{equation*}
$$

The incentive function $S_{l}(\cdot)$ of the hidden layer of BP neural network is Sigmoid function and is expressed by

$$
\begin{equation*}
S_{l}\left(U_{l, k}\right)=\frac{1}{1-\exp \left(-\lambda\left(U_{l, k}-\theta_{l}\right)\right)}, \tag{14}
\end{equation*}
$$

where $w_{l, n}$ is the weight between the input layer and the hidden layer; $r_{l}$ is the weight between the hidden layer and the output layer; and $\lambda$ is a fixed value, and the parameter indicates the transfer slope from the input layer to the hidden layer.

The Taylor series expansion of each output of the hidden layer at the threshold $\theta_{l}$ can be expressed by

$$
\begin{equation*}
Z_{l, k}=S_{l}\left(U_{l, k}\right)=\sum_{i=0}^{\infty} d_{i}\left(\theta_{l}\right)\left(U_{l, k}\right)^{i}, \tag{15}
\end{equation*}
$$

where $d_{i}\left(\theta_{l}\right)$ is the coefficient after the output of the hidden layer is expanded into a series form. The coefficient is a function of the hidden layer threshold BB of the BP neural network and is related to the excitation function of the hidden layer. The output of BP neural network is expressed in the form of linear summation. Based on Taylor series, the output of BP neural network can be expressed as follows:

$$
\begin{align*}
y(k) & =\sum_{l=1}^{L} r_{l} Z_{l, k}=\sum_{l=1}^{L} r_{l}\left[\sum_{i=0}^{\infty} d_{i}\left(\theta_{l}\right) U_{l, k}^{i}\right] \\
& =\sum_{l=1}^{L} r_{l} \sum_{i=0}^{\infty} d_{i}\left(\theta_{l}\right) \sum_{n_{1}=0}^{m-1} \cdots \sum_{n_{i}=0}^{m-1}\left[w_{l, n_{1}} \cdots w_{l, n_{i}} x_{l, n_{1}} \cdots x_{l, n_{i}}\right] . \tag{16}
\end{align*}
$$

On the other hand, if the Volterra kernel function is expanded in the form of basis function $\left\{b_{n}(z)\right\}$, the deformed Volterra functional model can be obtained from the practical functional model, as shown in Figure 8.

In Figure $8, V_{l}(k)$ is the weighted sum of the input time series of the Volterra series model, that is, discrete convolution, which can be expressed as follows:

$$
\begin{equation*}
V_{l}(k)=\sum_{z=0}^{m-1} b_{l, z} x_{k, z} . \tag{17}
\end{equation*}
$$

At the same time, the output of the system can be expressed in the polynomial form as follows:

$$
\begin{align*}
y(k)= & f\left(V_{1}(k), \ldots, V_{l}(k), \ldots, V_{L}(k)\right) \\
= & c_{0}+\sum_{l_{0}=0}^{m-1} c_{1}\left(l_{0}\right) V_{1}(k)+\sum_{l_{0}=0}^{m-1} \sum_{l_{2}=0}^{m-1} c_{2}\left(l_{1}, l_{2}\right) \cdot V_{l_{1}}(k) V_{l_{2}}(k)+\cdots \\
& +\sum_{l_{0}=0}^{m-1} \sum_{l_{2}=0}^{m-1} \cdots \sum_{l_{m}=0}^{m-1} c_{m}\left(l_{1}, l_{2}, \ldots, l_{m}\right) \cdot V_{l_{1}}(k) V_{l_{2}}(k) \cdots V_{l_{m}}(k), \tag{18}
\end{align*}
$$

where $c_{m}\left(l_{1}, l_{2}, \ldots, l_{m}\right)$ is the expanded polynomial coefficient. Selecting the appropriate basis function $b_{l, z}$ can make the error between each coefficient of the polynomial $f\left(V_{1}(k), \ldots, V_{l}(k), \ldots, V_{L}(k)\right)$ and the corresponding kernel function of the model in the Volterra series approach zero. Based on this, the Volterra series kernel function can be expressed as follows:

$$
\left\{\begin{array}{l}
h_{1}(z)=\sum_{l=0}^{m-1} c_{1}(l) b_{l, z}  \tag{19}\\
h_{2}\left(z_{1}, z_{2}\right)=\sum_{l_{1}=0}^{m-1} \sum_{l_{2}=0}^{m-1} c_{2}\left(l_{1}, l_{2}\right) b_{l_{1}, z_{1}} b_{l_{2}, z_{2}}
\end{array}\right.
$$

Through the analysis of equations (13) and (17), we can see that there are implicit variables $\left(U_{l, k}, V_{l}(k)\right)$ in the expressions and they are all discrete convolutions of the input time series.

If the basis function $\left\{b_{n}(z)\right\}$ in the practical Volterra functional model can be found, equation (18) can be expressed as the linear weighting of the Sigmoid function:

$$
\begin{equation*}
f\left(V_{1}(k), \ldots, V_{l}(k), \ldots, V_{L}(k)\right)=\sum_{l=1}^{L} r_{l} S_{l}\left(V_{l}(k)\right) \tag{20}
\end{equation*}
$$

Comparing equations (16) and (17), it can be seen that there is an equivalent relationship between Volterra functional series model and BP neural network model, and the two models are completely consistent in theory. The multivariable function is linearly weighted by the univariate function $\left\{g_{l}\left(V_{l}(k)\right)\right\}$ and can be expressed as follows:

$$
\begin{equation*}
f\left(V_{1}(k), \ldots, V_{l}(k), \ldots, V_{L}(k)\right)=\sum_{l=1}^{L} r_{l} g_{l}\left(V_{l}(k)\right) \tag{21}
\end{equation*}
$$

where the univariate function $\left\{g_{l}\left(V_{l}(k)\right)\right\}$ can be expressed in any form.
4.2. Prediction Model of Target Maneuver Trajectory Based on VNN. In Section 4.1, the equivalence between BP neural network and practical Volterra functional model is proved theoretically. Based on the equivalence between the two models, combined with BP neural network and Volterra series functional model, a time series prediction model of target maneuvering trajectory based on VNN is proposed. The model structure is shown in Figure 9. In the prediction model, the input vector $\mathbf{X}^{T}=[x(k), x(k+\tau), \ldots, x(k+(m-1) \tau)]$ of the prediction model is obtained by PSR. The convolution $V_{l}(k)$ of target maneuvering trajectory time series can be expressed as follows:


Figure 7: The structure of three-layer neural networks.

$$
\begin{equation*}
V_{l}(k)=\sum_{j=0}^{m-1} w_{l, j} x(k+j \tau) \tag{22}
\end{equation*}
$$

In the model, the polynomial function $g_{l}(\cdot)(l=1,2, \ldots, L)$ is selected as the excitation function, and it can be expressed as follows:

$$
\begin{equation*}
g_{l}(\cdot)=a_{0, l}+a_{1, l} x+a_{2, l} x^{2}+\cdots+a_{i, l} x^{i}+\cdots \tag{23}
\end{equation*}
$$

where $a_{i, l}$ is the coefficient of the polynomial expression. The output expression of the VNN model of target maneuvering trajectory time series can be obtained as follows:

$$
\begin{align*}
y(k) & =\sum_{l=1}^{L} r_{l} g_{l}\left(V_{l}(k)\right)=\sum_{l=1}^{L} \sum_{i=0}^{\infty} r_{l} a_{i, l}\left(V_{l}(k)\right)^{i} \\
& =\sum_{l=1}^{L} \sum_{i=0}^{\infty} r_{l} a_{i, l}\left(\sum_{j=0}^{m-1} w_{l, j} x(k+j \tau)\right)^{i} \tag{24}
\end{align*}
$$

where $y(k)$ is the output of the system, which is the maneuver trajectory of the target at the next moment, and $x$ is the input of the system, which is the historical trajectory of the target.

Based on the above theoretical analysis, the $i$ th order Volterra series kernel function can be expressed as follows:

$$
\begin{equation*}
h_{i}\left(z_{1}, z_{2}, \ldots, z_{i}\right)=\sum_{l=1}^{L} r_{l} a_{i, l} w_{l, z_{1}} w_{l, z_{2}} \cdots w_{l, z_{i}} \tag{25}
\end{equation*}
$$

The weights $w_{l, n}$ and $r_{l}$ and thresholds $\theta_{l}$ of the VNN model are obtained by training, and $\left\{g_{l}\left(V_{l}(k)\right)\right\}$ is expanded by Taylor series at the threshold. Finally, the kernel functions of each order of the model can be obtained according to equation (25).

## 5. Chaos Adaptive Genetic Algorithm

5.1. Fitness Function. Fitness function is used to guide the evolutionary search process of the adaptive algorithm, and the design of fitness function is closely related to the convergence speed and accuracy of the algorithm. In general, the fitness function is derived from the objective function. In order to reduce the prediction error of the VNN model, in this paper, an improved GA is proposed to optimize the model parameters, and the optimization problem belongs to
the minimum optimization problem, so the fitness function of the algorithm can be defined as follows:

$$
\begin{equation*}
F(\mathrm{Ch})=\frac{1}{n} \sum_{l=1}^{n}\left(\widehat{y}_{l}-y_{l}\right)^{2} \tag{26}
\end{equation*}
$$

where $n$ is the number of training samples of the algorithm; $y_{l}$ is the predicted value of the $l$ th sample; $\widehat{y}_{l}$ is the expected value of the $l$ th sample; and $F(\mathrm{Ch})$ represents the average prediction error when the parameters decoded by chromosome Ch are used as model parameters.
5.2. Selection Operator. In the selection operation of the genetic algorithm, roulette and optimal individual retention strategies are used to select the mating group [22]. When the roulette method is used to perform the selection operation, the probability of each chromosome being selected is positively correlated with its function fitness. $p_{j}$ is the probability that the $j$ th individual is selected, which is expressed by

$$
\left\{\begin{align*}
f_{j} & =\frac{k}{F\left(\mathrm{Ch}_{j}\right)}  \tag{27}\\
p_{j} & =\frac{f_{j}}{\sum_{j=1}^{M} f_{j}}
\end{align*}\right.
$$

where $f_{j}$ is the function fitness value of the $j$ th chromosome; $M$ is the size of the population; $k$ is the correlation coefficient; and $F\left(\mathrm{Ch}_{j}\right)$ can be calculated based on the fitness function defined above.
5.3. Adaptive Crossover Operator Based on the Difference Degree of Chromosome Gene Fragments. Crossover operation is a crucial operator in traditional genetic algorithms, which directly affects the convergence speed and global convergence ability of genetic algorithms.

The crossover operation of traditional genetic algorithms is based on random selection mechanism. When there is a high degree of similarity between the selected chromosomal gene fragments, according to biological genetics theory, inbreeding will greatly reduce the probability of new individuals appearing; that is, it is most likely to be meaningless crossover operation, which causes the convergence speed of the algorithm to slow down and the phenomenon of premature emergence. In order to improve the convergence speed and global convergence ability of the algorithm, inspired by the principle of excellent gene fragment cloning, crossover operation is carried out based on the differences between gene fragments. Based on the degree of difference between chromosomal gene fragments, the crossover probability of chromosomal gene fragments is determined, and the gene fragments are selected for crossover operation so as to achieve the purpose of reduced inbreeding and invalid crossover operations.

In this paper, a continuous gene segment with a length $L$ in the chromosome is defined as a chromosome gene segment, and the starting position of the chromosome gene segment is defined as a cross point.


Figure 8: Series Volterra model after deformation.

During the crossover operation, the length of the chromosomal gene fragment needs to be determined. According to the length coefficient and the randomly determined length ratio of the gene fragment, the length of the gene fragment can be obtained:

$$
\begin{equation*}
L_{\text {fragment }}=u_{1} L_{\text {chrom }} I, \tag{28}
\end{equation*}
$$

where $I$ is the chromosomal gene segment length coefficient; $L_{\text {chrom }}$ is the total length of the chromosomal gene segment; and $u_{1}$ is the ratio of the chromosomal gene segment length, which is a random number in the range of $[0,1]$. Chromosome gene segment coefficient is a parameter that controls the length of gene segment from the whole. At the same time, based on the overall control, the random length ratio $u_{1}$ is introduced and effectively increases the diversity of crossover operations.

In the crossover operation based on the difference degree of gene fragments, the gene fragment length coefficient directly affects the convergence of the algorithm. In the early stage of the algorithm, the larger value of the gene fragment length coefficient is helpful to expand the search space of the population and avoid the premature algorithm; in the latter part of the algorithm, the algorithm should focus on local search to speed up the convergence rate of the algorithm, so the value of the gene fragment length coefficient should be smaller. Based on this, an adaptive gene fragment length coefficient is constructed, and it can be expressed as follows:

$$
\begin{equation*}
I=\left(I_{\min }-I_{\max }\right)\left[\left(\frac{t}{T_{\max }}\right)^{2}-2\left(\frac{t}{T_{\max }}\right)+1\right]+I_{\max } \tag{29}
\end{equation*}
$$

where $I_{\min }$ is the minimum value of the gene fragment length coefficient; $I_{\max }$ is the maximum value of the gene fragment length coefficient; $t$ is the current iteration number of the algorithm; and $T_{\max }$ is the maximum iteration number.

The coding length coefficient $I$ of the gene fragment decreases nonlinearly with the number of iterations of the algorithm. The specific steps of the crossover operation


Figure 9: Volterra neural network model of chaotic time series.
based on the improvement of the difference degree of the gene fragment are as follows:

Step 1: determine the maximum $I_{\max }$ and minimum $I_{\text {min }}$ length coefficients of gene fragments and generate a random number $u_{1} \in[0,1]$. According to the number of iterations of the current algorithm and the maximum number of iterations initially set by the algorithm, the gene fragment length is calculated by equation (29).
Step 2: using the roulette method to randomly select individual parents $x^{(1)}=\left(x_{1}^{(1)}\right.$, $\left.x_{2}^{(1)}, \ldots, x_{n}^{(1)}\right)$ and $x^{(2)}=\left(x_{1}^{(2)}, x_{2}^{(2)}, \ldots, x_{n}^{(2)}\right)$.
Step 3: calculate the degree of difference between gene fragments of length $L_{\text {fragment }}$ corresponding to parents $x^{(1)}$ and $x^{(2)}$, calculate the probability of crossover by determining the degree of difference between gene fragments, and select the intersection of gene fragments according to the probability of crossover of gene fragments. The gene fragment difference degree and crossover probability can be expressed as follows:

$$
\begin{align*}
& D(i)=\sqrt{\frac{1}{L} \sum_{j=i}^{(i+1) * L-1}\left(x_{i}^{(1)}-x_{i}^{(2)}\right)^{2}} ; \quad i=0,1, \ldots, \operatorname{INT}\left(\frac{n}{L}\right), \\
& P(i)=\frac{D(i)}{\sum_{j=0}^{\operatorname{INT}(n / L)} D(j)} ; \quad i=0,1, \ldots, \operatorname{INT}\left(\frac{n}{L}\right), \tag{30}
\end{align*}
$$

where $D(i)$ is the difference degree of the $i$-th gene fragment of the parent individual; $D(j)$ is the difference degree of the $j$-th gene fragment of the parent individual; $x_{i}^{(1)}$ and $x_{i}^{(2)}$ are the gene encoding at the $i$-th gene position of the parent individual; and INT is the rounding function.
Step 4: generate a random number $u_{2} \in[0,1]$ and get the individual $y^{(1)}=\left(y_{1}^{(1)}, y_{2}^{(1)}, \ldots, y_{n}^{(1)}\right)$ and $y^{(2)}=\left(y_{1}^{(2)}, y_{2}^{(2)}, \ldots, y_{n}^{(2)}\right)$ of the offspring from the parent $x^{(1)}=\left(x_{1}^{(1)}, x_{2}^{(1)}, \ldots, x_{n}^{(1)}\right) \quad$ and
$x^{(2)}=\left(x_{1}^{(2)}, x_{2}^{(2)}, \ldots, x_{n}^{(2)}\right)$ through the cross, which can be expressed as follows:

$$
\begin{align*}
& y_{i}^{(1)}= \begin{cases}u_{2}\left(x_{i}^{(2)}-x_{i}^{(1)}\right)+x_{i}^{(1)}, & i \in\left(i_{\text {start }} i_{\text {start }}+L_{\text {fragment }}-1\right), \\
x_{i}^{(1)}, & i \notin\left(i_{\text {start }} i_{\text {start }}+L_{\text {fragment }}-1\right),\end{cases} \\
& y_{i}^{(2)}= \begin{cases}u_{2}\left(x_{i}^{(1)}-x_{i}^{(2)}\right)+x_{i}^{(2)}, & i \in\left(i_{\text {start }} i_{\text {start }}+L_{\text {fragment }}-1\right), \\
x_{i}^{(2)}, & i \notin\left(i_{\text {start }} i_{\text {start }}+L_{\text {fragment }}-1\right),\end{cases} \tag{31}
\end{align*}
$$

where $i$ is the gene position encoded by the gene and $i_{\text {start }}$ is the gene position where the starting point on the gene fragment is located.
5.4. Mutation Operation. The mutation probability $p_{m}$ directly affects the mutation of population. The appropriate mutation of individuals can keep the population diversity and prevent the algorithm from falling into local optimum. However, if the mutation probability $p_{m}$ is too large, the algorithm is similar to random search and loses the genetic evolution characteristics. In this paper, the mutation probability is improved from two aspects of the genetic evolution algebra and the fitness function value of individual population, and the mutation probability $p_{m}$ is updated adaptively, which is computed by

$$
p_{m}^{i}= \begin{cases}p_{\max }+\left(p_{\max }-p_{\min }\right)\left(\frac{t}{2 T_{\max }}+\frac{\left(f_{i}-f_{\mathrm{avg}}\right)}{2\left(f_{\max }-f_{\mathrm{avg}}\right)}\right), & f_{i} \geq f_{\mathrm{avg}}  \tag{32}\\ p_{\min }, & f_{i}<f_{\mathrm{avg}}\end{cases}
$$

where $p_{\text {max }}$ is the maximum mutation probability; $p_{\text {min }}$ is the minimum mutation probability; $f_{i}$ is the fitness value of the $i$ th individual; and $f_{\text {avg }}$ is the average of individual fitness of the current population.

The existing mutation operators are mainly divided into two categories: one is the mutation operator with strong local search ability [23, 24], such as Gauss mutation operator; the other is the mutation operator with strong global search ability [24], such as Cauchy mutation operator. For the optimization problem with less extreme points, mutation operators with strong local search ability should be used; for the optimization problem with more extreme points, mutation operators with strong global search ability should be used, and if mutation operators with strong local search ability are used, it is easy to fall into local optimal value. In summary, it is difficult for a single mutation operator to take into account both global search and local search capabilities. Therefore, a combination mutation operator is proposed in this paper.

In this paper, three kinds of mutation operators are used: the local search ability of the first mutation operator increases with the increase in the number of iterations; the global search ability of the second mutation operator is strong; and the local search ability of the third mutation operator is strong. The combination of the three mutation
operators can make the mutation operation take into account both the global and local search ability. The combined mutation operator method can be described as follows:
(1) Adaptive mutation operator:

$$
\begin{equation*}
X_{i}^{\prime}=X_{i}+\lambda_{1} * \frac{\left(b^{T}-a^{T}\right)}{t} \tag{33}
\end{equation*}
$$

where $X_{i}^{\prime}$ is the offspring individual after mutation; $a$ and $b$ are the upper and lower limits of the variable values; and $\lambda_{1}=\left(\lambda_{11}, \lambda_{12}, \ldots, \lambda_{1 m}\right)$, where $\lambda_{11}, \lambda_{12}, \ldots, \lambda_{1 m}$ is a random number in the range of $[-1,1]$.
(2) Cauchy mutation operator:

$$
\begin{equation*}
X_{i}^{\prime}=X_{i}+\text { Cauchy }(0,1) \tag{34}
\end{equation*}
$$

where Cauchy $(0,1)$ is the standard Cauchy distribution.
(3) Gaussian mutation operator:

$$
\begin{equation*}
X_{i}^{\prime}=X_{i}+N\left(\mu, \delta^{2}\right) \tag{35}
\end{equation*}
$$

where $\mu$ is the mean value in the normal distribution, and generally the value is $\mu=X_{i} ; \delta^{2}$ is the variance of the normal distribution, and this parameter is calculated by

$$
\begin{equation*}
\delta^{2}=\frac{\left|X_{\text {best }}-X_{i}\right|}{6} \tag{36}
\end{equation*}
$$

where $X_{\text {best }}$ is the best individual in the population.
(4) Switching method of the combination mutation operator:

$$
\bmod (t)= \begin{cases}0, & \text { Adaptive mutation }  \tag{37}\\ 1, & \text { Cauchy mutation } \\ 2, & \text { Gaussian mutation }\end{cases}
$$

where mod is the function to find the remainder.
The first mutation operator has a strong global search ability at the beginning of the iteration; with the increase in the number of iterations, the local search capability of the algorithm gradually increases. But, when the number of iterations increases to a certain value, it almost loses the function of mutation. When the number of iterations is not very large, the effect of this mutation operator is obvious. The second mutation operator is Cauchy mutation. Compared with the normal mutation operator, the Cauchy
mutation operator will produce larger mutation step length, so it will make the algorithm have better global search ability. The third mutation operator is normal mutation. The normal mutation operator focuses on searching a local area near the original individual. The local search ability is better, but the ability to guide the individual to jump out of the local better solution is weak, which is not conducive to global convergence. Therefore, the combined mutation operator takes into account both the global exploration capability and the local search capability, which enables GA to quickly converge to the global optimal solution.
5.5. The IGA-VNN Model Learning Algorithm for Target Maneuver Trajectory Time Series. Firstly, based on the PSR theory, the C-C method is used to determine the optimal time delay $t$ and time window $\tau_{w}$, and then the optimal embedding dimension is determined by the relationship $\tau_{w}=(m-1) \tau$ between time delay and time window. Secondly, the small data method is used to calculate the maximum Lyapunov exponents of the target maneuvering trajectory time series, the chaotic characteristics of target maneuver trajectory time series is judged, and the time series of the target maneuver trajectory is reconstructed according to the obtained embedding dimension $\tau$ and delay time $m$. Then, the VNN model structure is determined. According to the embedding dimension obtained by the C-C method, the number of neurons in the input layer is determined as $m$; according to the empirical formula [25], the number of nodes in the hidden layer is determined as $L$; according to the sample characteristics, the number of neurons in the output layer is determined as one, and a chaotic time series IGA-VNN prediction model with $m-L-1$ network structure is established. Finally, the IGA-VNN model is constructed to obtain the kernel functions of each order of the Volterra series, and the specific execution steps are as follows:

Step 1: the C-C method is used to determine the time delay $\tau$ and embedding dimension $m$ of the phase space reconstruction. Based on this, the target maneuvering trajectory time series is reconstructed and $M=N-$ $(m-1) \tau$ phase space vectors are obtained.
Step 2: according to Figure 9, the input vector of the IGA-VNN prediction model with $m-L-1$ structure is $\mathbf{X}^{T}$, the output of prediction model is $y(k)$, and the weights of the hidden layer and output layer are $W=$ $\left(w_{i, j}\right)_{L \times m}(j=1,2, \ldots, m)$ and $r_{l}$, respectively.
Step 3: the improved genetic algorithm is used to optimize the network parameters in the IGA-VNN target maneuver trajectory time series prediction model. The specific optimization steps are as follows:

Firstly, according to the characteristics of the genetic algorithm, the network parameters including the connection weights and thresholds between input layer and hidden layer and hidden layer and output layer are real coded to form chromosomes, and the population size of the genetic algorithm is determined
simultaneously. Secondly, the fitness function is determined, and it can be expressed as follows:

$$
\begin{equation*}
F=k(e(k+1)-\tilde{x}(k+1)), \tag{38}
\end{equation*}
$$

where $e(k+1)$ is the expected output of the network and $\tilde{x}(k+1)$ is the predicted output of the network.
Then, selection, crossover, and mutation operations are carried out to generate a new generation of population. Finally, the fitness value of the evolved population is calculated, and the above operations are repeated until the condition $\left|F_{l+1}-F_{l}\right|<\varepsilon$ is met, and then the optimal initial weight and threshold value of the network are obtained; otherwise, the previous step is returned until the iteration end condition is met.
Step 4: based on the weights and thresholds obtained in Steps 1, 2, and 3, combined with equation (10), the IGA-VNN prediction model is used to predict the target maneuver trajectory time series.
Step 5: calculate the prediction error $E$ of the network, and it can be expressed as follows:

$$
\begin{equation*}
E=\frac{1}{2} \sum_{k=1}^{N^{\prime}}(y(k)-\tilde{y}(k))^{2}, \tag{39}
\end{equation*}
$$

where $y(k)$ is the real value and $\widetilde{y}(k)$ is the predicted value.
In this paper, we set the maximum error $E_{\text {max }}$ of network prediction as 0.025 . If the condition $E<E_{\max }$ is met, the calculation will be stopped, and the network weight matrix $\mathbf{W}=\left(w_{l, j}\right)_{L \times m}$ and weight coefficient $r_{l}$ obtained after training will be stored. The kernel functions $h_{i}\left(l_{1}, l_{2}, \ldots, l_{i}\right)$ of each order in the Volterra model are calculated with polynomial coefficients $a_{i, l}(i=1,2, \ldots, m)$; otherwise, the next step is taken.
Step 6: based on the gradient descent method, the local gradient $\delta_{l}(k)$ and the correction of the network weight $\Delta w_{l, j}(k)$ of the IGA-VNN prediction model of chaotic time series are calculated, respectively. This can be calculated as follows:

$$
\begin{align*}
\delta_{l}(k) & =-\frac{\partial E}{\partial y(k)} g_{l}^{\prime}\left(V_{l}(k)\right),  \tag{40}\\
\Delta w_{l, j}(k) & =\alpha \Delta w_{l, j}(k-1)+\eta \delta_{l}(k) y(k),
\end{align*}
$$

where $\alpha \Delta w_{l, j}(k-1)$ is the introduced moving vector and $0<\alpha<1$ and $\eta$ is the learning step.
Step 7: correct the weights and thresholds of the network and train the prediction network again, calculate the error between the predicted output of the network and the true value of the sample, and repeat the training until the condition $E<E_{\max }$ is satisfied. After the training is completed, the weight matrix of the network is stored, and the Taylor series coefficients are decomposed at the threshold $\theta_{l}$ simultaneously, and then the Taylor expansion coefficient $d_{i}\left(\theta_{l}\right)$ can be
obtained. Because the output excitation function is in the form of polynomials, $d_{i}\left(\theta_{l}\right)=a_{i, l}$ can be obtained through mathematical analysis. At this time, the kernel functions of each order of the Volterra series can be obtained by substituting the coefficients of the Taylor expansion into equation (11).
5.6. The Framework of the Target Maneuver Trajectory Forecast Mode. The flowchart of the proposed target maneuver trajectory prediction model in this paper is depicted in Figure 10, which consists of four modules: (a) preprocessing module; (b) the IGA-VNN model learning algorithm module; (c) the whole prediction algorithm module; and (d) evaluation module. Furthermore, the detailed explanation is listed as follows:
(a) Preprocessing module: In order to determine if there is chaotic character of the target maneuver trajectory time series, the small data quantity method is used to calculate the maximum Lyapunov exponent. In addition, the C-C method is used to determine the input forms of the proposed target maneuver trajectory forecast model. Then, the normalization method is applied.
(b) The IGA-VNN model learning algorithm module: In view of the shortcomings of the genetic optimization algorithm, a strategy of the adaptive crossover mutation operator and combination mutation operator based on difference degree of gene fragments is proposed in this module. In addition, the improved genetic optimization algorithm is applied to optimize the parameter of the VNN model.
(c) The whole prediction algorithm module: combining phase space reconstruction and the Volterra adaptive filter, a novel target maneuver trajectory forecast model is proposed. The IGA-VNN model is developed in this paper to search for the optimal kernel functions of each order of the Volterra adaptive filter which is conducive to enhance the forecast accuracy.
(D) Evaluation module: four metrics which are $R_{\text {mse }}$ (root mean square error), $M_{\mathrm{ae}}$ (mean absolute error), $M_{\text {ape }}$ (mean absolute percent error), and $C_{\text {or }}$ (correlation coefficient) are used as evaluating indicators in this paper.

## 6. Simulation Experiment

6.1. Experimental Data Source and Simulation Scenarios. In order to verify the effectiveness and accuracy of the target maneuver trajectory prediction model, in this paper, 4000 consecutive sets of flight confrontation training data are intercepted from the air combat simulation system, and the data contain the target's three-dimensional coordinates and pitch angle and yaw angle data. A complete air combat confrontation is shown in Figure 11, the initial
confrontation scenario is shown in Table 1, and the threedimensional coordinate components of the target flight trajectory are shown in Figure 12.
6.2. Experimental Data Processing. When using the IGAVNN prediction model to predict the chaotic target maneuver trajectory time series, in order to obtain better prediction performance, it is necessary to normalize and inverse normalize the historical data [26]. Assuming that $\mathbf{X}=\left(x_{1}, x_{2}, \ldots, x_{N}\right), \mathbf{X} \in R^{N}$ is the original data of the target maneuver trajectory time series, and the normalization processing can be described as follows:

$$
\begin{equation*}
x(t)^{\prime}=\frac{x(t)-x_{\min }}{x_{\max }-x_{\min }}, \quad t=1,2, \ldots, N \tag{41}
\end{equation*}
$$

where $N$ is the length of the target maneuvering trajectory time series; $x(t)$ is the original value of the historical trajectory; $x(t)^{\prime}$ is the normalized value; $x_{\text {min }}$ is the minimum values in the historical trajectory; and $x_{\text {max }}$ is the maximum values in the historical trajectory.

Assuming that the output of the IGA-VNN target maneuvering trajectory prediction model is $\mathbf{Y}=\left(y_{1}, y_{2}, \ldots, y_{N}\right)$ and $\mathbf{Y} \in[0,1]$, the antinormalization processing can be expressed as follows:

$$
\begin{equation*}
y(t)^{\prime}=y(t)\left(y_{\max }-y_{\min }\right)+y_{\min }, \quad t=1,2, \ldots, N \tag{42}
\end{equation*}
$$

where $y(t)$ is the model output; $y(t)^{\prime}$ is the output after inverse normalization; $y_{\text {min }}$ is the minimum values in the model output; and $y_{\text {max }}$ is the maximum values in the model output.
6.3. Simulation Experiment Settings. In order to evaluate the performance of the target maneuvering trajectory prediction model, in this paper, the concept of multistep prediction is introduced [27]. Multistep prediction has a high use value in the aspects of perception situation and threat assessment. Multistep prediction of target maneuvering trajectory means that, based on the existing target historical trajectory data, the model can not only predict the target trajectory at the next moment but also predict the target trajectory at multiple moments in the future. The structure of the multistep prediction model is shown in Figure 13.

Based on the one-step prediction model, the single-step prediction result $x(k+1)$ is used to update the input vector $\mathbf{p}$ of the multistep prediction model, and $\mathbf{p}=\left[\begin{array}{llll}x(k+1) & x(k) & \cdots & x(k-m+1)\end{array}\right]^{T}$, and the prediction value at $k+2$ time can be obtained by calling the Volterra series prediction model. In this way, the input vector is updated repeatedly, and the Volterra one-step prediction model is called recursively, so the output value at any time in the future can be predicted theoretically.

In this paper, four prediction models are used to predict the target maneuvering trajectory time series in one step, two


Figure 10: Continued.

(d)

Figure 10: The flowchart of the proposed target maneuver trajectory prediction model.


Figure 11: A complete air combat trajectory map.

Table 1: The scenarios for maneuver trajectories.

| Scenario [1] | Initial condition | Velocity $(\mathrm{m} / \mathrm{s})$ | Pitch $(\mathrm{rad})$ | Roll $(\mathrm{rad})$ | Yaw $(\mathrm{rad})$ | Distance $(\mathrm{m})$ |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: |
| Fighter | $(0,0,8569)$ | 214.19 | -0.7837 | 2.5941 | 1.4825 |  |
| Target | $(-64259,-4360,9327)$ | 248.76 | 0.0017 | 0.4744 | 1.4971 |  |

steps, four steps, and six steps, respectively. The specific steps are as follows.

Firstly, $N-(m-1) \tau$ phase space vectors are obtained through phase space reconstruction, and $M$ vectors are selected as training samples from the phase space vectors. After the training, the network parameters obtained by the training are stored. Then, the chaotic time series IGA-VNN prediction
model, GA-VNN prediction model, RBF prediction model, second-order Volterra functional prediction model, and BP neural network prediction model are used to make $1,2,4$, and 6 step predictions on the target maneuver trajectory, respectively.

Finally, in order to quantify and compare the prediction accuracy and effectiveness of the four prediction algorithms, this paper uses root mean square error $R_{\text {mse }}$, mean absolute


Figure 12: Sample 3D data graph.


Figure 13: Multistep prediction model.
error $M_{\text {ae }}$, mean absolute percentage error $M_{\text {ape }}$, and correlation coefficient $C_{\text {or }}$ as evaluation indicators:

$$
\begin{aligned}
R_{\mathrm{mse}} & =\sqrt{\frac{1}{n} \sum_{t=1}^{n}(\widetilde{y}(k)-y(k))^{2}} \\
M_{\mathrm{ae}} & =\sqrt{\sum_{t=1}^{n} \frac{|\widetilde{y}(k)-y(k)|}{n}} \\
M_{\mathrm{ape}} & =\frac{1}{n} \sum_{t=1}^{n}\left|\frac{\widetilde{y}(k)-y(k) \mid}{\widetilde{y}(k)}\right| \\
C_{\mathrm{or}} & =\frac{\sum_{t=1}^{n}[\widetilde{y}(k)-\overline{\widetilde{y}}] \cdot[y(k)-\bar{y}]}{n \cdot \sigma(\widetilde{y}) \cdot \sigma(y)}
\end{aligned}
$$

where $n$ is the number of the sample, $\tilde{y}$ is the actual value, that is, the true position of the target at the next moment; $y$ is the predicted value, that is, the predicted position of the target at the next moment; $\sigma(y)$ is the standard deviation of $y$; $\bar{y}$ is the average value of $\tilde{y}$; and $\sigma(\tilde{y})$ is the standard deviation of $\tilde{y}$.

### 6.4. Analysis of Simulation Experiment Results

6.4.1. Predictive Performance Comparison of the Six Models. Based on PSR, phase space reconstruction is performed on the target maneuver trajectory samples to obtain the input and output vectors of the prediction model. In this paper, IGA-VNN prediction model, GA-VNN prediction model, RBF prediction model, second-order Volterra functional prediction model, and BP neural network prediction model are used to predict the three-dimensional coordinates of the
target's maneuvering trajectory in one step, two steps, four steps, and six steps, respectively. The prediction results are shown in Figures 14-16, and the comparison results of target maneuvering trajectory prediction are shown in Figures 17-20, and the comparison results of target maneuvering trajectory prediction error are shown in Figures 21-23.

In order to more intuitively see the prediction accuracy of each model, the forecasting evaluation indicators of each model are calculated as shown in Tables 2-4.
6.4.2. Results Analysis. By comparing and analyzing the prediction performance of different models in Tables 2-4, the following conclusions can be drawn:
(1) When performing multistep prediction on the target maneuver trajectory time series, the prediction performance of the chaotic time series IGA-VNN model is the best with the same prediction steps. Because the Volterra functional model can represent most nonlinear systems, it can accurately predict chaotic time series. However, the solution of highorder Volterra kernel function is the bottleneck of its application. In this paper, we establish the IGA-VNN model to determine the kernel functions of Volterra series, so that Volterra series can predict the target maneuver trajectory well.
(2) RBF neural network prediction model, second-order Volterra functional prediction model, and BP neural network prediction model show different prediction performance in different prediction steps and different coordinate directions of the target, which shows that the adaptability and robustness of these three prediction algorithms are poor, and the performance of the IGA-VNN prediction model is superior than that of these three prediction algorithms in this respect. In this paper, the phase space reconstruction theory is adopted to reconstruct the time series of target maneuver trajectory, which makes full use of the information contained in the time series of target maneuver trajectory so that the prediction accuracy is improved.
(3) Combined with the prediction error comparison charts in Figures 14-16, it can be seen that the same prediction algorithm's single-step prediction results for the three coordinates of the target maneuver trajectory are better than the 2 -step prediction results; the 2 -step prediction results are better than the 4 -step prediction results; and the 4 -step prediction results are better than 6 -step prediction results. Therefore, it can be seen that for the same prediction model, the prediction performance of the model will also decline with the increase in prediction steps. In
the multistep prediction of the target maneuver trajectory, as the number of prediction steps increases, the cumulative error of the prediction will continue to increase, so the prediction accuracy will also decrease.

By comparing and analyzing the differences between the four models in the multistep prediction principle of target maneuver trajectory time series, the root cause of these differences can be further analyzed. BP neural network is based on sample data to train the network so as to realize the prediction of the future state. The important factors that affect the precision and prediction accuracy of the BP neural network are the subjectivity and blindness of its variable selection, which leads to the addition of extra irrelevant variables or ignoring important variables. In addition, BP neural network is sensitive to initial value and easy to fall into local minimum. The Volterra functional model essentially uses the chaotic characteristics of target maneuvering trajectory time series to predict the nonlinear system by training and fitting the trajectory of the chaotic attractor. The model is implemented by the multiplicationcoupled form of the second-order Volterra series, which avoids the difficulty of solving the kernel function caused by the higher-order Volterra series and improves the prediction accuracy of the model. In addition, the Volterra filter uses the linear adaptive algorithm to adjust the filter parameters, and this dynamic property can improve the adaptability and accuracy of prediction and reduce the training time.

The IGA-VNN prediction model based on chaos theory combines the discrete Volterra functional model, BP neural network, and the chaotic characteristics of the target maneuvering trajectory time series and uses the chaotic characteristics, causality, and memory function [25] of the target maneuvering trajectory time series to determine the truncation order and truncation term number of the Volterra functional model so as to establish an accurate Volterra functional model. At the same time, the equivalence of BP neural network and Volterra functional model is used to solve the problem of high-order kernel function of the Volterra functional model, and the global searching ability of the improved genetic algorithm is used to avoid falling into local minimum when optimizing the VNN model. In conclusion, the IGA-VNN prediction model based on chaos theory reasonably uses the advantages of the Volterra functional model and BP neural network and effectively solves the problem of subjectivity in BP neural network modeling and the difficulty in the Volterra functional model solving high-order kernel function, making the performance of the algorithm in multistep prediction better than that of the single BP neural network and Volterra functional model.


| -BP | - |
| :--- | :--- |
| - IGA-VNN |  |
| - | Volterra |
| -RBF | - |
| GA-VNN |  |
|  | Actual value |

(a)



- BP
- Volterra
_ RBF
- IGA-VNN
- GA-VNN
- Actual value
(b)

(d)
(c)
- Actual value
- IGA-VNN
- GA-VNN

Actual value

Figure 14: The curves of (a) 1-step, (b) 2-step, (c) 4-step, and (d) 6-step X-coordinate prediction results.


Figure 15: Continued.


| -BP | - IGA-VNN |
| :--- | :--- |
| - Volterra | - GA-VNN |
| RBF | - Actual value |

(c)


- BP
- Volterra
RBF
- GA-VNN
- Actual value
(d)

Figure 15: The curves of (a) 1-step, (b) 2-step, (c) 4-step, and (d) 6-step Y-coordinate prediction results.

-BP
$-\quad$ Volterra
$=\mathrm{RBF}$

- IGA-VNN
- GA-VNN
- Actual value
(a)



| - BP | IGA-VNN |
| :--- | :--- |
| - Volterra | - GA-VNN |
| - RBF | - Actual value |

(b)


| -BP | - |
| :--- | :--- |
| - IGA-VNN |  |
| - | - GAlterra |

(d)

Figure 16: The curves of (a) 1-step, (b) 2-step, (c) 4-step, and (d) 6-step Z-coordinate prediction results.


Figure 17: Comparison of one-step prediction results of target maneuvering trajectory.


Figure 18: Comparison of two-step prediction results of target maneuvering trajectory.


Figure 19: Comparison of four-step prediction results of target maneuvering trajectory.


Figure 20: Comparison of six-step prediction results of target maneuvering trajectory.


Figure 21: The curves of (a) 1-step, (b) 2-step, (c) 4-step, and (d) 6-step $X$-coordinate prediction error.


Figure 22: Continued.


Figure 22: The curves of (a) 1-step, (b) 2-step, (c) 4-step, and (d) 6-step Y-coordinate prediction error.


Figure 23: The curves of (a) 1-step, (b) 2-step, (c) 4-step, and (d) 6-step Z-coordinate prediction error.

Table 2: Target $X$-coordinate multistep prediction performance comparison of four different prediction methods.

| Prediction step | Algorithm | RMSE | MAE | MAPE | COR |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 1 | BP | 31.2054 | 5.2515 | 0.0011 | 0.9987 |
|  | IGA-VNN | 25.0554 | 4.2594 | $7.3741 * e-04$ | 0.9984 |
|  | GA-VNN | 46.6161 | 6.2959 | 0.0016 | 0.9983 |
|  | Volterra | 88.0949 | 9.2494 | 0.0035 | 0.9985 |
|  | RBF | 65.5883 | 7.4046 | 0.0023 | 0.9987 |
| 2 | BP | 58.2029 | 7.1255 | 0.0021 | 0.9972 |
|  | IGA-VNN | 43.1102 | 5.6694 | 0.0013 | 0.9969 |
|  | GA-VNN | 51.5107 | 6.6589 | 0.0018 | 0.9961 |
|  | Volterra | 103.4790 | 9.4192 | 0.0036 | 0.9920 |
|  | RBF | 103.9374 | 8.9800 | 0.0033 | 0.9888 |
| 4 | BP | 100.8075 | 9.0122 | 0.0032 | 0.9983 |
|  | IGA-VNN | 64.8278 | $7.2519$ | 0.0021 | 0.9952 |
|  | GA-VNN | 130.0170 | 10.6018 | 0.0046 | 0.9935 |
|  | Volterra | 227.3807 | 12.1373 | 0.0059 | 0.9816 |
|  | RBF | 171.4466 | 11.3985 | 0.0053 | 0.9636 |
| 6 | BP | 151.8142 | 11.9705 | 0.0058 | 0.9978 |
|  | IGA-VNN | 115.1146 | 9.7368 | 0.0038 | 0.9866 |
|  | Volterra | 200.2440 | 13.0592 | 0.0070 | 0.9980 |
|  | RBF | 238.1826 | 13.9321 | 0.0079 | 0.9531 |

Table 3: Target Y-coordinate multistep prediction performance comparison of four different prediction methods.

| Prediction step | Algorithm | RMSE | MAE | MAPE | COR |
| :--- | :---: | :---: | :---: | :---: | :---: |
|  | BP | 48.0217 | 3.9851 | 0.0069 | 0.9965 |
|  | IGA-VNN | 15.6534 | 3.4766 | 0.0022 | 0.9987 |
| 1 | GA-VNN | 30.7653 | 5.2296 | 0.0047 | 0.9970 |
|  | Volterra | 40.5759 | 5.9218 | 0.0061 | 0.9971 |
|  | RBF | 48.8712 | 6.3457 | 0.0073 | 0.9974 |
|  | BP | 99.3602 | 8.6527 | 0.0139 | 0.9913 |
|  | IGA-VNN | 23.0582 | 4.2662 | 0.0033 | 0.9980 |
| 2 | GA-VNN | 57.3526 | 6.6614 | 0.0078 | 0.9932 |
|  | Volterra | 59.8836 | 7.1030 | 0.0089 | 0.9948 |
|  | RBF | 70.9127 | 7.8841 | 0.0109 | 0.9899 |
|  | BP | 123.7001 | 10.4490 | 0.0188 | 0.9851 |
|  | IGA-VNN | 35.4839 | 5.5817 | 0.0056 | 0.9976 |
| 4 | GA-VNN | 119.9039 | 9.9738 | 0.0175 | 0.9752 |
|  | Volterra | 148.4173 | 10.9043 | 0.0221 | 0.9904 |
|  | RBF | 119.6835 | 9.2310 | 0.0159 | 0.9798 |
|  | BP | 161.1386 | 11.6505 | 0.0236 | 0.9685 |
|  | IGA-VNN | 44.4239 | 6.4155 | 0.0073 | 0.9952 |
|  | GA-VNN | 158.5973 | 11.4905 | 0.0242 | 0.9846 |
|  | Volterra | 189.8012 | 12.7370 | 0.0295 | 0.9880 |
|  | RBF | 159.7916 | 10.9816 | 0.0203 | 0.9618 |

Table 4: Target Z-coordinate multistep prediction performance comparison of four different prediction methods.

| Prediction step | Algorithm | RMSE | MAE | MAPE | COR |
| :--- | :---: | :---: | :---: | :---: | :---: |
|  | BP | 3.8168 | 1.8070 | $8.7264 * e-04$ | 0.9959 |
|  | IGA-VNN | 1.4914 | 1.0943 | $3.2070 * e-04$ | 0.9984 |
| 1 | GA-VNN | 3.5207 | 1.7437 | $8.1077 * e-04$ | 0.9958 |
|  | Volterra | 2.0636 | 1.3294 | $4.7240 * e-04$ | 0.9937 |
|  | RBF | 5.0411 | 2.0680 | 0.0011 | 0.9720 |
|  | BP | 7.6877 | 2.6837 | 0.0019 | 0.9881 |
|  | IGA-VNN | 4.9893 | 2.1206 | 0.0012 | 0.9968 |
| 2 | GA-VNN | 4.9718 | 2.2086 | 0.0013 | 0.9957 |
|  | Volterra | 6.2689 | 2.4025 | 0.0015 | 0.9479 |
|  | RBF | 7.9969 | 2.3284 | 0.0018 | 0.9572 |

Table 4: Continued.

| Prediction step | Algorithm | RMSE | MAE | MAPE | COR |
| :---: | :---: | :---: | :---: | :---: | :---: |
| 4 | BP | 11.9821 | 3.2566 | 0.0028 | 0.8768 |
|  | IGA-VNN | 6.4740 | 2.3860 | 0.0015 | 0.9757 |
|  | GA-VNN | 8.0413 | 2.7889 | 0.0021 | 0.9978 |
|  | Volterra | 8.1816 | 2.5405 | 0.0017 | 0.9316 |
|  | RBF | 14.0847 | 3.396 | 0.0032 | 0.9044 |
| 6 | BP | 14.1209 | 3.2762 | 0.0029 | 0.8273 |
|  | IGA-VNN | 9.8469 | 2.8852 | 0.0022 | 0.9677 |
|  | GA-VNN | 10.9808 | 2.9773 | 0.0024 | 0.9585 |
|  | Volterra | 10.1371 | 2.9463 | 0.0023 | 0.8627 |
|  | RBF | 20.9611 | 4.4327 | 0.0052 | 0.8633 |

## 7. Conclusions

In this paper, a VNN network model is designed by analyzing the chaotic characteristics of the target maneuver trajectory time series, and the IGA-VNN prediction model of target maneuver trajectory based on chaos theory is proposed, and then the target maneuver trajectory time series is predicted in multiple steps. The prediction method proposed in this paper combines the advantages of the Volterra functional model and BP neural network algorithm, overcomes the blindness in building BP neural network, and solves the difficulties in determining the higher-order kernel function of the Volterra functional model, which has certain theoretical application value.
(1) In this paper, the C-C method is used to reconstruct the phase space of the target maneuver trajectory, and the chaotic characteristic of the target maneuver trajectory time series is proved by the method of small data volume.
(2) In this paper, the IGA-VNN target trajectory prediction model based on chaos theory is constructed. The model solves the problem of solving the higher-order kernel function of the Volterra functional model and realizes the accurate prediction of target maneuvering trajectory. In multistep prediction, the prediction performance of the model is better than that of the single BP neural network and Volterra functional prediction model.
(3) The IGA-VNN prediction model based on chaos theory proposed in this paper combines the chaotic characteristics of the target maneuver trajectory time series to determine the dimension of the input vector of the prediction network. It makes the truncation term and truncation order of the Volterra functional model have clear physical meaning, avoids the blindness brought by subjective determination of parameters, and improves the practical value of the model.
(4) In addition, in order to realize the application value of the IGA-VNN prediction model of chaotic time series, the realization of its hardware is also the focus of future research.

## Data Availability

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

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Authors' Contributions

Zhifei Xi was responsible for conceptualization, data analysis, investigation, methodology, software, visualization, and original draft preparation. An Xu acquired funding and supervised the study. Zhanwu L and Yingxin Kou were responsible for project administration. Aiwu Yang validated the study and reviewed and edited the manuscript.

## References

[1] Y. X. Kou, Z. W. Li, X. D. Chen et al., "The important role of fire control system in air combat-soul of fighters," Electronics Optics \& Control, vol. 20, no. 12, pp. 1-5, 2013.
[2] B. C. Jiang, J. Guan, W. Zhou et al., "Vessel trajectory prediction algorithm based on polynomial fitting Kalman filtering," Journal of Signal Processing, vol. 35, no. 5, pp. 741-746, 2019.
[3] S. Zhao, C. Tang, S. Liang et al., "Track prediction of vessel in controlled waterway based on improved Kalman filter," Journal of Computer Applications, vol. 32, no. 11, pp. 32473250, 2012.
[4] S. J. Qiao, N. Han, X. W. Zhu et al., "A dynamic trajectory prediction algorithm based on Kalman filter," Acta Electronica Sinica, vol. 46, no. 2, pp. 418-423, 2018.
[5] D. L. Zhai, H. M. Lei, J. Li et al., "Trajectory prediction of hypersonic vehicle based on adaptive IMM," Acta Aeronautica et Astronautica Sinica, vol. 37, no. 11, pp. 3466-3475, 2016.
[6] B. Yang and Z. H. He, "Hypersonic vehicle track prediction based on GRNN," Computer Applications and Software, vol. 32, no. 7, pp. 239-243, 2015.
[7] K. Qian, Y. Zhou, L. J. Yang et al., "Aircraft target track prediction model based on BP neural network," Command Information System and Technology, vol. 8, no. 3, pp. 54-58, 2017, in Chinese.
[8] W. Tan, B. C. Lu, and M. L. Huang, "Track prediction based on neural networks and genetic algorithm," Journal of Chongqing Jiaotong University, vol. 291, no. 1, pp. 147-150, 2010.
[9] X. S. Gan, J. S. Duanmu, Y. B. Meng et al., "Aerodynamic modeling from flight data based on WNN optimized by particle swarm," Acta Aeronautica et Astronautica Sinica, vol. 33, no. 7, pp. 1209-1217, 2012.
[10] A. Y. Yakubov, "On nonlinear Volterra equations of convolution type," Differential Equations, vol. 45, no. 9, pp. 1326-1336, 2009.
[11] S. Murakami and P. H. A. Ngoc, "On stability and robust stability of positive linear Volterra equations in Banach lattices," Central European Journal of Mathematics, vol. 8, no. 5, pp. 966-984, 2010.
[12] Y. V. Bibik, "The second Hamiltonian structure for a special case of the Lotka-Volterra equations," Computational Mathematics and Mathematical Physics, vol. 47, no. 8, pp. 1285-1294, 2007.
[13] D. Y. Chen, Y. Liu, and X. Y. Ma, "Parameter joint estimation of phase space reconstruction in chaotic time series based on radial basis function neural networks," Acta Physica Sinica, vol. 61, no. 10, pp. 22-31, 2012.
[14] S. Wang, Y. Y. Sui, Y. Dong et al., "LAN traffic flow prediction using improved small data method," Journal of Jilin University (Engineering and Technology Edition), vol. 46, no. 4, pp. 1254-1260, 2016.
[15] L. J. Wang, X. Z. Liao, S. Gao et al., "Chaos characteristics analysis of wind power generation time series for a grid connecting wind farm," Transactions of Beijing Institute of Technology, vol. 27, no. 12, pp. 1077-1080, 2007.
[16] H. B. Zhang, X. D. Sun, and Y. L. He, "Analysis and prediction of complex dynamical characteristics of short-term traffic flow," Acta Physica Sinica, vol. 63, no. 4, Article ID 0405051, 2014.
[17] Y. M. Qin, X. Y. Li, J. Ren et al., "Prediction of the adiabatic film cooling effectiveness influnenced by multi parameters based on BP neural network," Journal of Engineering Thermophysics, vol. 32, no. 7, pp. 1127-1130, 2011.
[18] Z. B. Lu, Z. M. Cai, and K. Y. Jiang, "Prediction of chao tic time series using singular value decomposition Volterra filter," Journal of Wuhan University of Technology (Transportation Science \& Engineering), vol. 31, no. 4, pp. 672-675, 2007.
[19] R. Y. Liu and L. Huang, "Wind power forecasting based on dynamic neural networks," Automation of Electric Power Systems, vol. 36, no. 11, pp. 19-37, 2012, in Chinese.
[20] J. S. Zhang and X. C. Xiao, "Prediction low-dimensional chaotic time series using Volterra adaptive filers," Acta Physica Sinica, vol. 49, no. 3, pp. 404-408, 2000, in Chinese.
[21] J. L. Yu, Y. Zhang, and J. Zhou, "Continuous attractors of Lotka-Volterra recurrent neural networks with infinite neurons," IEEE Transactions on Neural Networks, vol. 21, no. 10, pp. 1690-1695, 2010.
[22] Y. Mei, G. Z. Tan, and Z. T. Liu, "Improved brain emotional learning algorithm for fast classification," Tien Tzu Hsueh Pao/Acta Electronica Sinica, vol. 45, no. 11, pp. 2663-2670, 2017.
[23] X. H. Wang, X. Y. Liu, and M. H. Bai, "Migration algorithm with Gaussian variation and steepest descent operator," Journal of Computer Engineering and Applications, vol. 45, no. 20, pp. 57-60, 2009.
[24] J. J. Li and Y. M. Dai, "Frog leaping algorithm based on adaptive mixed variation," Journal of Computer Engineering and Applications, vol. 49, no. 10, pp. 58-61, 2013.
[25] L. S. Yin, Y. G. He, X. P. Dong et al., "Research on the multistep prediction of Volterra neural network for traffic flow," Acta Automatica Sinica, vol. 40, no. 9, pp. 2066-2072, 2014.
[26] Y. Y. Guo, Terminal Departures Passenger Traffic Prediction Based on Chaotic Time Series, Harbin Institute of Technology, Harbin, China, 2013.
[27] Y. C. Jiang, B. J. Zhang, F. F. Xing et al., "Super-short-term multi-step prediction of wind power based on GA-VNN model of chaotic time series," Power System Technology, vol. 39, no. 8, pp. 2160-2166, 2015.

