entropy

MDPI

Article

Multilayer Perceptron Network Optimization for Chaotic Time Series Modeling

Mu Qiao [1,2], Yanchun Liang [3,4], Adriano Tavares [2] and Xiaohu Shi [3,4,*]

[1] School of Mathematics, Jilin University, Changchun 130021, China
[2] Department of Industrial Electronics, School of Engineering, University of Minho, 4800-058 Guimares, Portugal
[3] Key Laboratory of Symbol Computation and Knowledge Engineering of the Ministry of Education, College of Computer Science and Technology, Jilin University, 2699 Qianjin Street, Changchun 130012, China; ycliang@jlu.edu.cn
[4] School of Computer Science, Zhuhai College of Science and Technology, Zhuhai 519041, China
[*] Correspondence: shixh@jlu.edu.cn

Abstract: Chaotic time series are widely present in practice, but due to their characteristics—such as internal randomness, nonlinearity, and long-term unpredictability—it is difficult to achieve high-precision intermediate or long-term predictions. Multi-layer perceptron (MLP) networks are an effective tool for chaotic time series modeling. Focusing on chaotic time series modeling, this paper presents a generalized degree of freedom approximation method of MLP. We then obtain its Akachi information criterion, which is designed as the loss function for training, hence developing an overall framework for chaotic time series analysis, including phase space reconstruction, model training, and model selection. To verify the effectiveness of the proposed method, it is applied to two artificial chaotic time series and two real-world chaotic time series. The numerical results show that the proposed optimized method is effective to obtain the best model from a group of candidates. Moreover, the optimized models perform very well in multi-step prediction tasks.

Keywords: chaotic time series; multilayer perceptron network; generalized degrees of freedom; Akaike information criterion; maximal Lyapunov exponent

check for updates

Citation: Qiao, M.; Liang, Y.; Tavares, A.; Shi, X. Multilayer Perceptron Network Optimization for Chaotic Time Series Modeling. *Entropy* **2023**, *25*, 973. https://doi.org/10.3390/e25070973

Academic Editors: Augustine Wong and Xiaoping Shi

Received: 17 April 2023
Revised: 8 June 2023
Accepted: 16 June 2023
Published: 24 June 2023

1. Introduction

Some nonlinear dynamical systems present a seemingly irregular and random phenomenon, but they are actually produced by deterministic systems. For example, the famous "butterfly effect" concept describes how the flapping of a butterfly's wings in Brazil could ultimately lead to a tornado in Texas that would not have happened otherwise. In other words, very tiny changes in conditions can result in very large, unpredictable responses, but this follows basic aerodynamic laws. This characteristic of nonlinear dynamical systems is called chaos. Chaotic dynamical systems are ubiquitous in nature, so they have been extensively studied to solve practical problems in different fields, such as financial systems analysis [1,2], power system behavior [3,4], information security [5,6], and the control of nonlinear systems [7,8]. In general, chaotic dynamical systems do not have an explicit dynamical equation and can only be understood through the available time series. Therefore, chaotic time series modeling is of great significance, but it is very difficult due to its complex, chaotic, nonlinear dynamics, such as internal randomness, nonlinearity, and long-term unpredictability.

Various researchers have shown great interest in chaotic time series analysis, finding many results, which could be classified into symbolic regression, polynomial model, model simulation or decomposition, and neural networks. For symbolic regression, Brandejsky used a GPA-ES system to study symbolic regression of deterministic chaotic systems [9], and Senkerik et al. proposed a novel tool for symbolic regression and analytical programming for the evolutionary synthesis of discrete chaotic systems [10]. The polynomial model is a widely used method. Lainscsek et al. used a truncated polynomial expansion involving successive derivatives of the measured time series according to an Ansatz library [11]. This procedure was introduced and improved by [12]. Han et al. utilized genetic programming (GP) and multi-objective optimization to identify chaotic systems using the nonlinear auto-regressive moving average with exogenous inputs (NARMAX) model representing the basis of the hierarchical tree encoding in GP [13]. Model simulation or decomposition is another type of method. Karimov et al. constructed a chaotic circuit from data to identify chaos systems [14]. Yang et al. introduced the Hankel Alternative View of Koopman analysis to decompose chaotic dynamics into a linear model with intermittent forcing [15]. The most widely used methods of chaotic time series analysis are neural network-related methods, which are classified into artificial neural networks (ANN) [16–20], fuzzy neural networks (FNN) [21–24], optimization algorithms with ANN [25–28], and wavelet neural networks (WNN) [29–32]. One can refer to [33] for a comprehensive review. Although there have been many attempts at chaotic time series analysis by neural network methods, including deep learning [16,34–36], there are still many issues to be resolved, such as low prediction accuracy and difficulty in determining the network topologies. Therefore, some scholars have studied the network optimization of chaotic time series analysis models. Xie et al. proposed an enhanced grey wolf optimizer (GWO) by designing four distinctive search mechanisms and then developed the evolving convolutional neural network–long short-term memory (CNN-LSTM) networks for time series analysis [37]. Huang et al. proposed an improved differential evolution (IDE) algorithm to optimize the topology of DHNNs consisting of CNN and GRU, including CNNs' filter size and the number of hidden neurons of the GRU [38]. Focusing on the prediction of short-term traffic flow, Qian et al. found that the RBF neural network is better than the wavelet neural network, and they used the genetic algorithm to optimize the initial parameters [39]. Chen et al. presented a nonlinear ensemble of partially connected neural networks for short-term load forecasting, in which the genetic algorithm is used to generate diverse and effective neural networks, and a novel pruning method was developed to optimize the partially connected neural networks [40]. Kao et al. studied a Takagi-Sugeno-Kang (TSK)-type self-organizing fuzzy neural network, which not only generates and prunes the learning algorithm automatically but also adjusts the parameters of existing fuzzy rules [41]. Though all the above methods work on network structure optimization, they are empirical and have no strict theoretical basis.

The Akaike information criterion (AIC) [42] is a popular tool to evaluate a linear auto-regression model. The idea of AIC has been extended to nonlinear models, especially the neural networks based on the generalized degrees of freedom [43–45]. However, when employing the Akaike information criterion based on the generalized degrees of freedom (AIC_g) to evaluate the MLP network, the generalized degrees of freedom must be estimated. For a linear regression model, the degrees of freedom are the number of variables in the model, which is the sum of the sensitivities of each fitted value with respect to the corresponding observed values. For nonlinear models, Ye [45] defined a new quantity known as the generalized degrees of freedom (GDF) and proved that the GDF provides an unbiased estimation of the error variance. However, it is very difficult to calculate the GDF using the original definition. A Monte Carlo method is therefore suggested to calculate the GDF for a nonlinear model. This approach requires estimating the nonlinear model many times and is therefore time-consuming.

Recently, a new method for the estimation of the generalized degrees of freedom for the RBF-type network has been introduced [44]. Following that method, the generalized hat matrix for the nonlinear model is defined in this study; its trace is an estimation of the generalized degrees of freedom. The proposed estimation method simulates the model procedure once only and therefore is faster than the Monte Carlo method. Using the AIC_g, the performances of the MLP networks for different network topologies are evaluated, and the optimal MLP network for the chaotic time series is then selected. The model-selecting method is applied to four chaotic time series, two artificial time series and two real-world time series. By estimating the maximal Lyapunov exponents and forecasting the time series, the numerical results show that the MLP network selected by the proposed method behaves better in terms of the analytic and prediction capabilities than other models. The main contributions of the paper are the following three points:

1. Present an approximation method to calculate the generalized degree of freedom of MLP, and then obtain its $\mathrm{AIC_g}$.
2. Develop a multilayer perceptron network optimization method for chaotic time series analysis by designing the $\mathrm{AIC_g}$ as the loss function of MLP.
3. Propose an overall framework for chaotic time series analysis, including phase space reconstruction, model training, and model selection modules.

This paper is arranged as follows: In Section 2, some backgrounds are introduced, including phase space reconstruction of chaotic systems, the Akaike information criterion of nonlinear systems, and generalized degrees of freedom of MLP. In Section 3, our proposed modeling selection framework for MLP networks is described in detail. In Section 4, the proposed method is applied to four chaotic time series, and the results are presented and analyzed. The last section gives a brief conclusion.

2. Backgrounds

2.1. Phase Space Reconstruction of Chaotic Systems

Suppose that the chaotic system is an n-dimensional system given by

$$\begin{cases} x_1^{(i+1)} = f_1\left(x_1^{(i)}, x_2^{(i)}, \cdots, x_n^{(i)}\right) \\ x_2^{(i+1)} = f_2\left(x_1^{(i)}, x_2^{(i)}, \cdots, x_n^{(i)}\right) \\ \qquad\qquad \cdots \\ x_n^{(i+1)} = f_n\left(x_1^{(i)}, x_2^{(i)}, \cdots, x_n^{(i)}\right) \end{cases} \tag{1}$$

where $X = (x_1, x_2, \ldots, x_n)$ is an n-dimension vector.

In general, the real observed chaotic time series is only one or a few dimensional state sequences. In order to analyze the chaotic time series, it is necessary to reconstruct the phase space of the chaotic system from the observed low-dimensional time series and restore the chaotic attractor of the original phase space in this high-dimensional embedded space. Takens' theorem is the theoretical basis for phase space reconstruction [46].

Theorem 1. *(Takens' theorem* [46]*): Suppose M is an m-dimensional compact manifold, the map φ: $M \to M$ is a smooth differential isomorphism, y: $M \to R$ is a smooth function, and the map $\Phi_{(\varphi,\, y)}$: $M \to R^{2m+1}$ is defined by*

$$\Phi_{(\varphi,\, y)} = (y(x), y(\varphi(x)), y(\varphi^2(x)), \ldots, y(\varphi^{2m}(x))) \tag{2}$$

Then, $\Phi_{(\varphi,\, y)}$ is an embedding from M to R^{2m+1}.

This theorem states that for an m-dimensional chaotic dynamic system, as long as the embedding dimension is greater than or equal to $2m + 1$, the chaotic attractor can be restored in the embedding space; that is, a system with the same dynamic properties as the original chaotic system can be obtained in the embedding space. This mathematically guarantees that a phase space equivalent to the original system in the topological sense can be reconstructed from a lower-dimensional or even one-dimensional chaotic time series. The coordinate delay method is commonly used to reconstruct the phase space of the chaotic system. The essence is to construct m-dimensional phase space vectors from an observed one-dimensional time series $\{x_i\}$ $(i \in [1, N])$ according to a series with different time delays, namely, [47]

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{N-(m-1)\tau} \end{bmatrix} = \begin{bmatrix} x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(m-1)\tau} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-(m-1)\tau} & x_{N-(m-2)\tau} & \cdots & x_N \end{bmatrix} \tag{3}$$

where N is the time series length, m the embedding dimension, and τ the time delay. According to Takens' theorem, if the choice of the embedding dimension m and time delay τ is appropriate, the "orbit" of the reconstructed phase space in the embedded space is equivalent to the dynamics of the original system (Equation (1)) in the sense of differential isomorphism. This method is called the delay coordinate embedding approach. In this approach, the determination of the values of m and τ is the key to phase space reconstruction.

2.2. Akaike Information Criterion of a Nonlinear System

Assume that a nonlinear system is modeled by the function [44]

$$y = f\,(\mathbf{X}, \boldsymbol{\alpha}) + \varepsilon, \tag{4}$$

where $\mathbf{X} = (x_1, x_2, \ldots, x_d) \in \mathbf{R}^{\mathrm{d}}$ is the input, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_K) \in \mathbf{R}^{\mathrm{K}}$ is the model parameter, $\varepsilon \in \mathbf{R}$ is the random noise, and $y \in \mathbf{R}$ is the output. Assume also that we have N input samples $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N)$ and N output observers $\mathbf{Y} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N)$, and denote the function output of $\mathbf{X}_\mathrm{n}$ as $y_\mathrm{n} = f\,(\mathbf{X}_\mathrm{n}, \boldsymbol{\alpha}) + \varepsilon$. Then, the residual sum of squares (RSS) of the MLP network is [44]

$$RSS = \sum_{j=1}^{N} (y_n - \hat{y}_n)^2$$

For a nonlinear model, the quadratic loss of the model can be estimated by [30]

$$A = RSS - N\sigma^2 + 2n_g\sigma^2 \tag{5}$$

where σ is the variance of the observation error and n_g is the generalized degrees of freedom. The mean of $RSS/(N - n_g)$ was proven to be equal to the variance σ^2; hence, the quadratic loss can be approximated by [45]

$$\begin{aligned} A &\approx RSS - N\sigma^2 + 2n_g\tfrac{RSS}{N-n_g} \\ &= RSS\left(\tfrac{N+n_g}{N-n_g}\right) - N\sigma^2 \\ &= RSS\left(\tfrac{N}{N-n_g}\right)^2\left(1 - \left(\tfrac{n_g}{N}\right)^2\right) - N\sigma^2 \end{aligned}$$

Since n_g is much smaller than N and σ is a constant, another criterion G could be obtained by [48]

$$G = RSS\left(\frac{N}{N - n_g}\right)^2 \tag{6}$$

Taking the logarithm for Equation (6), we can obtain

$$\begin{aligned} \ln(G) &= ln(RSS) - 2\ln(1 - \tfrac{n_g}{N}) \\ &= \ln(RSS/N) + \ln(N) + 2n_g/N + (\tfrac{n_g}{N})^2 + \ldots \\ &\approx \ln(RSS/N) + 2n_g/N + \ln(N) \end{aligned}$$

It should be noted that the above derivation makes use of the Taylor expansion. Then, we arrive at the form of the Akaike information criterion for an MLP network,

$$\mathrm{AIC}_g = ln(RSS/N) + 2n_g/N \tag{7}$$

which is based on the concept of entropy and able to weigh the complexity of the estimated model and the goodness of the fitted data of the model.

2.3. Generalized Degrees of Freedom (GDF)

In order to calculate the Akaike information criterion of an MLP network according to Equation (7), the generalized degrees of freedom (GDF) n_g should be obtained in advance. Xu et al. developed a fast estimation method for GDF of a nonlinear system [44], which is briefly described in this section for the convenience of the reader.

GDF is defined as "the sum of sensitivity of each fitted value to the perturbation in the corresponding observed value", which can measure the complexity of a nonlinear model [45]. Xu et al. proposed an approximated method to obtain the GDF of a nonlinear system [44].

Theorem 2. *If a nonlinear model $f(\boldsymbol{X}, \boldsymbol{\alpha})$ is adequate in fitting the data and has bounded second order derivatives, the generalized degrees of freedom for the nonlinear system can be approximated by*

$$n_g \approx trace(\boldsymbol{H}) \tag{8}$$

where $\boldsymbol{H}$ is the generalized hat matrix of the nonlinear system, defined by

$$\boldsymbol{H} = \boldsymbol{M}^T \left(\boldsymbol{M}^T \boldsymbol{M}\right)^{-1} \boldsymbol{M} \tag{9}$$

where $\boldsymbol{M}$ is the gradient matrix of parameters.

$$\boldsymbol{M} = \begin{bmatrix} \frac{\partial f(\boldsymbol{X}_1,\boldsymbol{\alpha})}{\partial \alpha_1} & \frac{\partial f(\boldsymbol{X}_1,\boldsymbol{\alpha})}{\partial \alpha_2} & \cdots & \frac{\partial f(\boldsymbol{X}_1,\boldsymbol{\alpha})}{\partial \alpha_{k-1}} & \frac{\partial f(\boldsymbol{X}_1,\boldsymbol{\alpha})}{\partial \alpha_k} \\ \frac{\partial f(\boldsymbol{X}_2,\boldsymbol{\alpha})}{\partial \alpha_1} & \frac{\partial f(\boldsymbol{X}_2,\boldsymbol{\alpha})}{\partial \alpha_2} & \cdots & \frac{\partial f(\boldsymbol{X}_2,\boldsymbol{\alpha})}{\partial \alpha_{k-1}} & \frac{\partial f(\boldsymbol{X}_2,\boldsymbol{\alpha})}{\partial \alpha_k} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \frac{\partial f(\boldsymbol{X}_{N-1},\boldsymbol{\alpha})}{\partial \alpha_1} & \frac{\partial f(\boldsymbol{X}_{N-1},\boldsymbol{\alpha})}{\partial \alpha_2} & \cdots & \frac{\partial f(\boldsymbol{X}_{N-1},\boldsymbol{\alpha})}{\partial \alpha_{k-1}} & \frac{\partial f(\boldsymbol{X}_{N-1},\boldsymbol{\alpha})}{\partial \alpha_K} \\ \frac{\partial f(\boldsymbol{X}_N,\boldsymbol{\alpha})}{\partial \alpha_1} & \frac{\partial f(\boldsymbol{X}_N,\boldsymbol{\alpha})}{\partial \alpha_2} & \cdots & \frac{\partial f(\boldsymbol{X}_N,\boldsymbol{\alpha})}{\partial \alpha_{k-1}} & \frac{\partial f(\boldsymbol{X}_N,\boldsymbol{\alpha})}{\partial \alpha_K} \end{bmatrix} \tag{10}$$

Note that the MLP basis functions clearly have bounded second order derivatives. Hence, Theorem 1 can be applied to calculate the generalized degrees of freedom for an MLP model.

3. Modeling Selection for MLP Networks

3.1. Pipeline of the Modeling Selection

MLP networks are effective tools for chaotic time series predictions. However, the network topology seriously affects the performance of algorithms. Therefore, a pipeline of the modeling selection for MLP networks is proposed in this paper, as shown in Figure 1. For a chaotic time series dataset, the phase space reconstruction should be performed in advance. Next, a candidate MLP structure pool is generated. Then, each structure is trained to determine its parameters. It should be noticed that the Akaike information criterion is used as the loss function for training. When the training process is completed, the MLP network structure with the least Akaike information criterion value is selected as the best one, that is, the final prediction model. The phase space reconstruction process is described in Section 3.2, and the Akaike information criterion computation is detailed in Section 3.3, respectively.

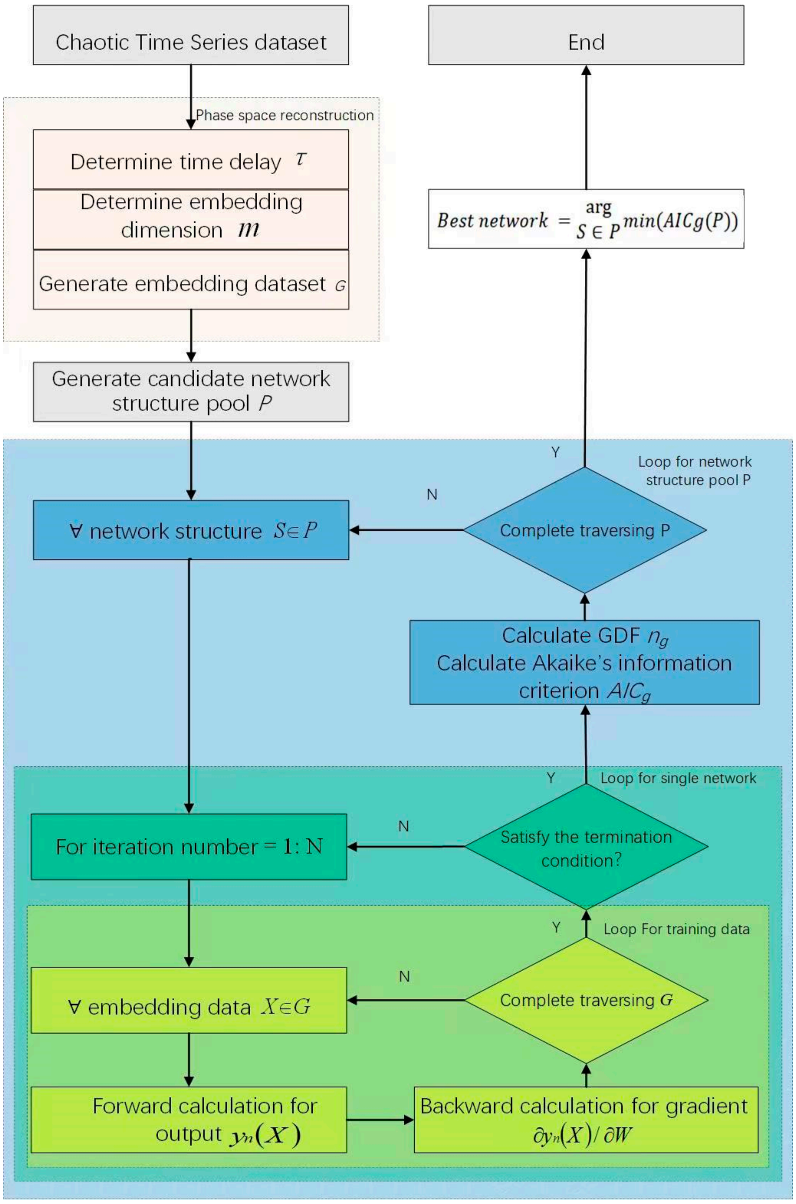

Chaotic Time Series dataset
End
Phase space reconstruction
Determine time delay τ
Determine embedding dimension m
Generate embedding dataset G
Best network = arg min(AICg(P)) S ∈ P
Generate candidate network structure pool P
Y
N
Loop for network structure pool P
∀ network structure S∈P
Complete traversing P
Calculate GDF n_g
Calculate Akaike's information criterion AIC_g
Y
Loop for single network
N
For iteration number = 1: N
Satisfy the termination condition?
Y
Loop For training data
N
∀ embedding data X∈G
Complete traversing G
Forward calculation for output $y_n(X)$
Backward calculation for gradient $\partial y_n(X)/\partial W$

Figure 1. Pipeline of the modeling selection.

3.2. Phase Space Reconstruction

According to Takens' theorem, the m-dimensional phase space of the chaotic system could be reconstructed from an observed one-dimensional time series by the delay coordinate embedding approach (Equation (3)). The key to the method is how to determine the embedding dimension m and time delay τ. To address this issue, many algorithms have been proposed: For example, the mutual information method [48] and the autocorrelation coefficients method [49] are two popular approaches to determine the time delay τ; the false nearest neighbor method [50] and its improved version the Cao method [51] are often used to determine the embedding dimension m; and Uzal et al. proposed a noise amplification approach able to handle these two tasks simultaneously [52]. In our pipeline, the time delay τ is determined by the mutual information method [48], and the embedding dimension m is selected by the Cao method [51].

We denote the observed one-dimensional time series $\{x_i\}$ as a discrete system S. The amount of information contained in this system can be characterized in terms of entropy:

$$H(S) = \sum_{s\in S} p_s log p_s \tag{11}$$

where p_s represents the probability of s occurring. We denote the delayed time series $\{x_{i+\tau}\}$ as another discrete system Q. Then, the joint entropy of the systems S and Q could be defined by.

$$H(S,Q) = \sum\nolimits_{s\in S,q\in Q} p_{sq} log p_{sq} \tag{12}$$

where p_{sq} represents the joint probability of s and q appearing in systems S and Q. Thereby, the mutual information between the two systems could be defined by

$$I(S,Q) = H(S) + H(Q) - H(S,Q) \tag{13}$$

which can be understood as the amount of overlapping information between these two systems. After fixing the original system S, system Q is determined by the time delay τ. Taking the time delay τ as a variable, the mutual information $I(S, Q)$ can be treated as a function of τ, denoted as $I(\tau)$. In order to make the correlation between the delayed time series and the original series as small as possible, the minimum point of $I(\tau)$ is taken as the final time delay:

$$\tau^* = \underset{\tau}{\mathrm{argmin}} I(\tau) \tag{14}$$

Considering that we want a small time delay, the first local minimum point is usually taken as the final result in the real applications. The pseudocode of the time delay determination is given by Algorithm 1.

Algorithm 1: Time delay determination

```
Input: one-dimensional time series $S = \{x_i\}$; time delay max value $\tau_{\max}$; iteration number $N$
Output: best time delay $\tau^*$
compute the entropy of $S: H(S) = \sum_{s\in S} p_s log\ p_s$
for i = 1: $N$
    time delay $\tau = i \times \tau_{\max}/N$
    time delayed system $Q = \{x_{i+\tau}\}$
    compute the entropy of $Q: H(Q) = \sum_{q\in Q} p_q log\ p_q$
    compute the joint entropy of $S$ and $Q: H(S,Q) = \sum_{s\in S,q\in Q} p_{sq} log\ p_{sq}$
    compute the mutual information of $S$ and $Q: I(\tau) = H(S) + H(Q) - H(S,Q)$
end for
$\tau^* = arg \min_{\tau} I(\tau)$
```

A chaotic time series is a projection in one- or lower-dimensional space of a deterministic dynamical system's trajectory in a high-dimensional phase space. The projection modifies the topological properties of the motion trajectory, resulting in "disorder" characteristic of chaotic systems. Thus, we can observe the change in the "nearest neighbor" distances when the embedding dimension increases, and it stops when it no longer changes.

Let x_i be a point of the original time series, and $X_i(m) = (x_i, x_{i+\tau}, \ldots, x_{i+(m-1)\tau})$ and $X_{i(m+1)} = (x_i, x_{i+\tau}, \ldots, x_{i+m\tau})$ be its mapping points in the m-dimensional and (m+1)-dimensional embedding spaces. Denote $X_{n(i,m)}$ (m) as the nearest neighbor to $X_i(m)$ in the embedding space, i.e.,

$$n(i,m) = \underset{j}{argmin} \parallel X_i^{(m)} - X_j^{(m)} \parallel_2 \tag{15}$$

Then, the change of the distance of this "nearest neighbor" pair when the embedding dimension increases from m to m + 1 could be represented by

$$a(i,m) = \frac{\parallel X_i^{(m+1)} - X_{n(i,m)}^{(m+1)} \parallel_\infty}{\parallel X_i^{(m)} - X_{n(i,m)}^{(m)} \parallel_\infty} \; i = 1,2,\cdots,n - m\tau \tag{16}$$

Thus, the average distance change of the "nearest neighbor" pairs could be expressed as

$$E(m) = \frac{\sum_{i=1}^{n-m\tau} a(i,m)}{n - m\tau} \quad m = 1, 2, \cdots . \tag{17}$$

It can be understood as the average scale of the nearest neighbor distance of all mapped points in m-dimensional embedding space after mapping to (m + 1)-dimensional space. Thus, we could define

$$E_1(m) = \frac{E(m+1)}{E(m)} \quad m = 1, 2, \cdots . \tag{18}$$

If the time series is not random, there exists an embedding dimension d^* satisfying the Takens' theorem, such that when $m > d^*$, $E_1(m)$ no longer changes; otherwise, $E_1(m)$ continues to increase.

To further ensure the reliability of the method, the following judgment criterion has been added:

$$E^*(m) = \frac{\sum_{i=1}^{n-d\tau} |x(i + m\tau) - x(n(i,m) + m\tau)|}{n - m\tau} \quad m = 1, 2, \cdots , \tag{19}$$

$$E_2(m) = \frac{E^*(m+1)}{E^*(m)} \quad m = 1, 2, \cdots . \tag{20}$$

Since there is no correlation between random sequence data, $E_2(m) \equiv 1$, and for deterministic sequences, there must exist m, such that $E_2(m) \neq 1$. The pseudocode of the embedding dimension determination is given by Algorithm 2.

Algorithm 2: Embedding dimension determination

```
Input: one-dimensional time series $S = \{x_i, \ldots, x_n\}$, time delay $\tau$;
          max embedding dimension $M$, threshold $\varepsilon$.
Output: embedding dimension $m^*$

for $k = 1 : \ n - m\tau$
for $m$ = 1: $M$
    $E(m)$ = 0; $E^*(m)$ = 0; $m^*$ = 0;
    for $i$ = 1: $n - m\tau$
        $n(i,m) = \underset{j}{arg\min} X_i^{(m)} - X_j^{(m)}{}_2$
        $a(i,m) = \frac{X_i^{(m+1)} - X_{n(i,m)}^{(m+1)}{}_\infty}{X_i^{(m)} - X_{n(i,m)}^{(m)}{}_\infty}$
        $E(m) = E(m) + \alpha(i, m)/(n - m\tau)$
        $E^*(m) = E^*(m) + |x(i + m\tau) - x(n(i,m) + m\tau)|/(n - m\tau)$
    end for
end for
$E_1(1) = E(2)/E(1)$
$E_2(1) = E^*(2)/E^*(1)$
for $m$ = 2: $M - 1$
    $E_1(m) = E(m + 1)/E(m)$
    $E_2(m) = E^*(m + 1)/E^*(m)$
    if ($|E_1(m) - E_1(m - 1)| < \varepsilon$) and ($|E_2(m) - 1| > \varepsilon$)
      $m^* = m$
    end if
end for
```

3.3. *Calculation of MLP Networks' Akaike Information Criterion (AICg)*

The Akaike information criterion ($\mathrm{AIC_g}$) offers a relative performance measure when nonlinear models are used to represent the process that actually generates the data. Hence, the $\mathrm{AIC_g}$ provides a means for model selection for an MLP network. In practice, several candidate MLP networks with different neuron sizes are built and trained using the same sample set. The residual sum of squares for every candidate network is then obtained. In order to apply the $\mathrm{AIC_g}$, the GDF for every candidate network is needed. According to Equation (8), the GDF can be estimated easily by the gradient matrix of the parameters. In this study, using the idea of back-propagation, we introduce a method to calculate the gradient matrix of the parameters for an MLP network.

Consider a multilayer perceptron network with one input layer, K hidden layers, and one output layer. There are M_k neurons in the kth hidden layer. In order to address the bias in the MLP network freely, we first expand the input vector and the neurons. Denote the expanding input vector by

$$\boldsymbol{U}^{(0)} = (x_{n,1}, x_{n,2}, \cdots, x_{n,d}, 1)^T \tag{21}$$

Then, the feed vector in the first hidden layer is

$$\boldsymbol{p}^{(1)} = \boldsymbol{w}^{(1)}\boldsymbol{U}^{(0)} \tag{22}$$

The parameter matrix $\mathbf{w}^{(1)}$ is given by

$$\boldsymbol{w}^{(1)} = \begin{pmatrix} w_{11}^{(1)} & w_{12}^{(1)} & \cdots & w_{1d}^{(1)} & \beta_1^{(1)} \\ w_{21}^{(1)} & w_{22}^{(1)} & \cdots & w_{2d}^{(1)} & \beta_2^{(1)} \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ w_{M_1 1}^{(1)} & w_{M_1 2}^{(1)} & \cdots & w_{M_1 d}^{(1)} & \beta_{M_1}^{(1)} \end{pmatrix} \tag{23}$$

where d is the dimension of the input vector, M_1 is the number of the first hidden layer nodes, and ${\beta_i}^{(1)}$ is the bias of the ith node, respectively.

In the same way, the input vector in the kth hidden layer can be extended by

$$\begin{aligned} \mathbf{U}(k-1) &= (o_1^{(k-1)}, o_2^{(k-1)}, \cdots, o_{M_{k-1}}^{(k-1)}, 1)T \\ &= (f(p_1^{(k-1)}, f(p_2^{(k-1)}, \cdots, f(p_{M_{k-1}}^{(k-1)}), 1)T \end{aligned} \tag{24}$$

The feed vector in the kth hidden layer is then given by

$$\boldsymbol{p}^{(k)} = \boldsymbol{w}^{(k)}\boldsymbol{U}^{(k-1)} \tag{25}$$

Finally, the input vector for the output layer is

$$\boldsymbol{U}^{(K)} = (o_1^{(K)}, o_2^{(K)}, \cdots, o_{M_{K-1}}^{(K)}, 1)^T \tag{26}$$

The feed value for the output layer is

$$\boldsymbol{p}^{(K+1)} = \boldsymbol{w}^{(K+1)}\boldsymbol{U}^{(K)} \tag{27}$$

The output of the MLP network is then given by

$$\hat{y}_n = g\left(\boldsymbol{p}^{(K+1)}\right) \tag{28}$$

Let the activation function $\boldsymbol{f}$ in the hidden layer be the sigmoid function

$$f(s) = \frac{1}{1 + exp(-s)} \tag{29}$$

Then

$$f'(s) = f(s)(1 - f(s)) \tag{30}$$

Let the activation function $\boldsymbol{g}$ in the output layer be the linear function

$$g(s) = s \tag{31}$$

Then

$$g'(s) = 1 \tag{32}$$

Denote the MLP network as a function

$$y_n = F\left(\boldsymbol{X}_n, \boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \cdots, \boldsymbol{w}^{(K+1)}\right) \tag{33}$$

Let $\mathbf{Q}^{(k)}(i,j)$ be a matrix with the same dimension as $\mathbf{w}^{(k)}$ where all its elements are zero except $q_{ij} = 1$. Furthermore, define matrix $\mathbf{O}^{(k)}$ by

$$\boldsymbol{O}^{(k)} = \begin{pmatrix} o_1^{(k)}\left(1 - o_1^{(k)}\right) & 0 & \cdots & 0 \\ 0 & o_2^{(k)}\left(1 - o_2^{(k)}\right) & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & o_{M_k}^{(k)}\left(1 - o_{M_k}^{(k)}\right) \\ 0 & 0 & \cdots & 0 \end{pmatrix} \tag{34}$$

Then, we have

$$\frac{d\boldsymbol{U}^{(k+1)}}{dy} = \boldsymbol{O}^{(k)} \frac{d}{dy}\left(\boldsymbol{w}^{(k)} \boldsymbol{U}^{(k-1)}\right) \tag{35}$$

Hence, the derivative of the MLP function with respect to $w_j^{(K+1)}$ is given by

$$\frac{\partial F}{\partial w_j^{(K+1)}} = \frac{\partial}{\partial w_j^{(K+1)}}\left(\boldsymbol{w}^{(K+1)} \boldsymbol{U}^{(K)}\right) = \boldsymbol{Q}^{(K+1)}(1,j)\boldsymbol{U}^{(K)},\ j = 1, 2, \cdots, M_K + 1 \tag{36}$$

Furthermore, the derivative of the MLP function with respect to $w_{j,m}^{(K)}$ is given by

$$\frac{\partial F}{\partial w_{j,m}^{(K)}} = \boldsymbol{w}^{(K+1)} \frac{\partial}{\partial w_{j,m}^{(K)}} \boldsymbol{U}^{(K)} = \boldsymbol{w}^{(K+1)} \boldsymbol{O}^{(K)} \boldsymbol{Q}^{(K)}(j,m) \boldsymbol{U}^{(K-1)} \tag{37}$$

By the same discussion, we obtain the derivative with respect to $w_{j,m}^{(k)}$,

$$\frac{\partial F}{\partial w_{j,m}^{(k)}} = \left(\prod\nolimits_{n=k}^{K} \boldsymbol{w}^{(n+1)} \boldsymbol{O}^{(n)}\right) \boldsymbol{Q}^{(k)}(j,m) \boldsymbol{U}^{(k-1)} \tag{38}$$

Thus, the gradient matrix of parameters $\boldsymbol{M}$ is obtained, and then the generalized hat matrix $\boldsymbol{H}$ can be calculated according to Equation (9). The trace of the generalized hat matrix is an estimator of the GDF n_g for the MLP network according to Equation (8).

Using the GDF estimated by the generalized hat matrix, the AIC_g for each MLP candidate is calculated by

$$\text{AIC}_g = ln(RSS/N) + 2n_g/N = \ln(\sum\nolimits_{j=1}^{N} (y_n - \hat{y}_n)^2 / N) + 2n_g/N \tag{39}$$

The pseudocode of the calculation of MLP networks' Akaike information criterion is given by Algorithm 3.

Algorithm 3: Calculation of MLP networks' AIC$_g$

Input: ANN model ANN = $\{w_{i,m}{}^{(k)}\}$, embedded time series data G = $\{X_1, \ldots, X_N\}$, iteration number N_Max.
Output: MLP networks' Akaike's information criterion AIC$_g$

```
for i = 1: N_Max
  for n = 1: N
    $\hat{y}_n = \mathrm{ANN}(X_n) = g\left(\mathbf{p}^{(K+1)}\right)$
    $\frac{\partial F}{\partial w_{j,m}^{(k)}} = \left(\prod_{n=k}^{K} \mathbf{w}^{(n+1)} \mathbf{O}^{(n)}\right) \mathbf{Q}^{(k)}(j, m) \mathbf{U}^{(k-1)}$
    $\mathbf{H} = \mathbf{M}^T (\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}$
    $n_g \approx \mathrm{trace}(\mathbf{H})$
    $\mathrm{AIC}_g = \ln(\sum_{j=1}^{N} (y_n - \hat{y}_n)^2 / N) + 2n_g/N$
  end for
end for
```

4. Experiments

4.1. Benchmark Datasets

In order to verify the effect of the proposed method, it is applied to simulate four chaotic time series: two artificial chaotic time series and two real-world time series. The two artificial benchmarks are the Hénon map [53] and the Lorenz map [54], and the two real ones are sunspots time series [55] and SST data [56], respectively.

Hénon map: The Hénon map [53] is given as

$$y_{n+1} = 1 - a y_n + b y_{n-1} \tag{40}$$

It has a chaotic attractor when a = 1.4 and b = 0.3. A time series of length 1500 is generated with initial conditions y_{-1} = 1.005 and y_0 = −0.3032. The maximal Lyapunov exponent (MLE) of the Henon attractor is about 0.419.

Lorenz equation: The Lorenz map is [54] defined as

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = -xz + rx - y \\ \frac{dy}{dt} = xy - bz \end{cases} \tag{41}$$

It becomes chaotic for σ = 10, r = 28, and b = 8/3. Equation (41) is a three-dimensional differential system, which can be solved numerically using the fourth order Runge-Kutta method with a time step of 0.01 and initial conditions of

$$x(0) = 2.4151, y(0) = 3.6936, z(0) = 15.1426 \tag{42}$$

A time series of length 1500 is generated with sampling frequency 0.2,

$$x_n = x(0.2n),\ 1 \le n \le 1500$$

The MLE for the Lorenz attractor is about 0.91. Since the sampling frequency is 0.2, the maximal Lyapunov exponent for the time series is therefore about 0.182.

Sunspots time series: Sunspots are temporary phenomena on the surface of the Sun (the photosphere) that are visible as dark spots compared to the surrounding regions. The sunspots index has been used since 1749. The monthly averaging sunspots index considered in this study is from 1749 to 2009 with a sample size of 3132. The dataset can be downloaded from the website: psl.noaa.gov/gcos_wgsp/Timeseries/SUNSPOT/ (accessed on 5 June 2023). The sunspots data are a nonlinear time series, which have been discussed as a chaotic time series by many authors [55].

SST data: The sign and magnitude of equatorial Pacific sea-surface temperature anomalies (SSTA) provide a measure of the ENSO phase and strength. The El Niño phase produces sea-surface temperatures in the equatorial eastern Pacific that are anomalously warm with respect to the mean seasonal cycle, while La Niña conditions produce anomalously cold sea-surface temperature conditions. The S index of Wright [56] provides a continuous historical record (1881–1986) of SST anomalies averaged over an irregular region of the equatorial Pacific extending for 5° N–5° S and 150°–90° W. The dataset is obtained from Geographic Data Sharing Infrastructure, College of Urban and Environmental Science, Peking University (http://geodata.pku.edu.cn, (accessed on 8 April 2023)).

4.2. Experiment Setup

Generally, a three-layer MLP network structure is selected for experiments. The activation function in the hidden layer is given by the sigmoid function, while the activation function in the output layer is a linear function. Then, we focus on discussing the optimal neuron size in the hidden layer. The Akaike information criterion based on the generalized degrees of freedom ($\mathrm{AIC_g}$) is employed to evaluate the performance of the MLP networks. During the model training process, the SGD optimization algorithm is used; the momentum is set to 0.9, the weight decay is 0.0005, and the initial learning rate is 0.01.

In the following section, we first describe the results of phase space reconstruction and then give the results of the network model optimization. The performance of the models obtained is also analyzed in detail.

Two measurement methods, the maximum Lyapunov exponent (MLE) and the mean square error (MSE) of the time series, were considered in the experiment for evaluating the experimental results.

Maximum Lyapunov Exponents (MLE):

The Lyapunov exponent can quantitatively characterize the chaotic attractor by measuring the sensitivity of the chaotic orbit to the original condition. In general, a time sequence is considered chaotic only when the *MaxLE* is positive based on the phase track. Li and Xu proposed an estimation method of *MaxLE* by the local Jacobian matrix of the nonlinear system [57]. In the reconstructed delay phase space, the local Jacobian matrix at this point is rewritten in the form

$$\boldsymbol{A}_n = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ b_{1,n} & b_{2,n} & b_{3,n} & \cdots & b_{d,n} \end{bmatrix}$$

where

$$b_{k,n} = \frac{\partial f(\boldsymbol{X}_n)}{\partial x_{n,k}} \tag{43}$$

In this case, the nonlinear function is the three-layer MLP network, and Equation (43) can be rewritten as

$$b_{k,n} = \sum_{m=1}^{M} w_m^{(2)} o_{m,n}(1 - o_{m,n}) w_{m,k}^{(1)} \tag{44}$$

Mean square error (MSE)

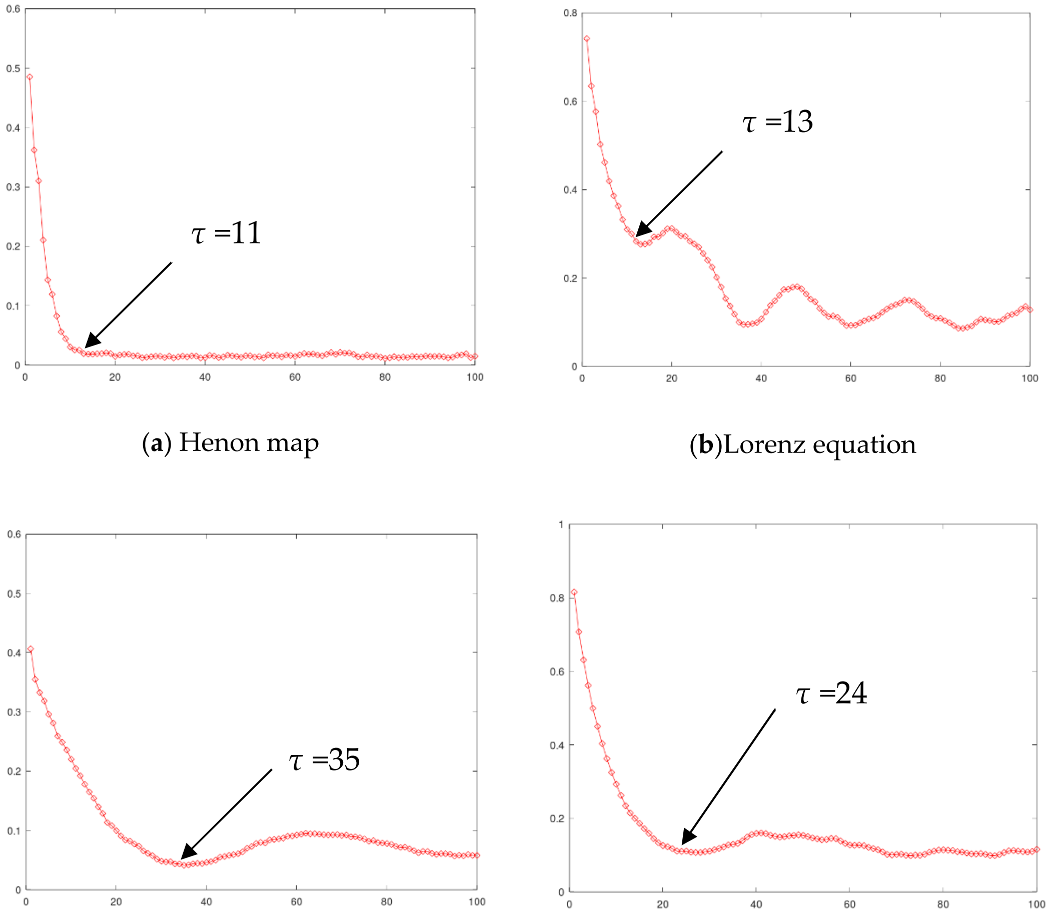
The mean square error (MSE) is defined by

$$MSE = \frac{1}{N} \sum_{j=1}^{N} (y_n - \hat{y}_n)^2 \tag{45}$$

where y_n and $\hat{y}_n$ are the MLP's output and expected output of the nth input, and N is the number of samples.

4.3. Phase Space Reconstruction

For phase space reconstruction, the two parameters of the delay interval τ^* and embedding dimension m^* should be determined. The delay interval τ^* is determined by the mutual information method, and the embedded dimension m^* is selected by the Cao method. Figure 2 shows the variation curve of mutual information with delay interval τ on four datasets. The first minimum point or inflection point of the four curves is selected as the final delay interval. In the Henon map dataset, when τ = 11, the curve enters the stationary region, so the time interval is selected as 11. In the Lorenz dataset, the curve reaches the first minimum point when τ = 13, so the time interval of the Lorenz dataset is selected as 13. In the sunspot dataset and the equatorial Pacific sea-surface temperature dataset, the first minimum is reached at τ = 35 and τ = 24, respectively, so the time interval is chosen as 35 and 24, respectively. Figure 3 shows the change of the two indicators E_1 and E_2 of the CAO method with the embedding dimension on four datasets, and we select the definite state where E_1 reaches a stationary state and E_2 does not. As shown in the figure, in the four datasets of the Henon map, Lorenz, sunspots, and equatorial Pacific sea-surface temperature, the embedding dimensions are selected as 5, 3, 7, and 4, respectively.



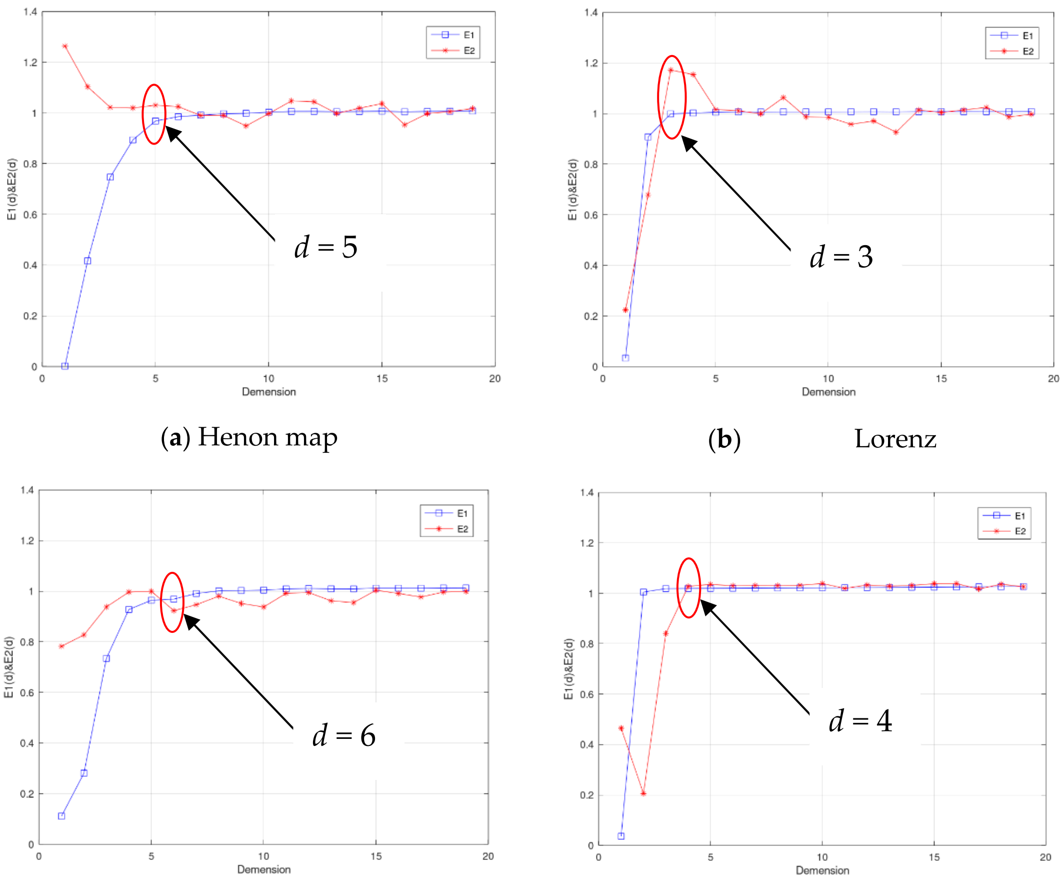
(**a**) Henon map

(**b**)Lorenz equation

(**c**) Sunspot

(**d**) SST

Figure 2. Variation curve of mutual information with delay interval τ on datasets of the (**a**) Henon map, the (**b**) Lorenz equation, (**c**) sunspot, and (**d**) SST.



(**a**) Henon map (**b**) Lorenz

(**c**) Sunspot (**d**) SST

Figure 3. The E_1 and E_2 indicators change curves with the embedded dimension d on datasets of the (**a**) Henon map, (**b**) Lorenz equation, (**c**) sunspot, and (**d**) SST.

4.4. Analysis for AIC_g

We trained different networks on the Lorentz dataset by randomly initializing the parameters (here, the same network topology is used but with different network weights). The number of nodes in the hidden layer of the network is set to 20, and 10 networks are obtained by training from 10 random initialized parameters. Table 1 shows the comparison results of generalized degrees of freedom (GDF), Akaike information criterion (AICg), and maximum Lyapunov exponents (MLE) for these 10 networks on the Lorenz dataset. The results show that the models with small AICg values prefer small MLE values. The scatterplots for MLE and AICg also show the same trend (Figure 4). In other words, AICg is a good criterion for selecting a good network for chaotic time series.

Table 1. Comparison results of $\mathrm{AIC_g}$, GDF, and MLE on 10 networks.

	GDF	**AIC**	**MLE**
1	76.00	−9.22	0.230
2	76.00	−9.37	0.236
3	69.93	−8.31	0.237
4	74.01	−8.67	0.226
5	76.00	−8.99	0.244
6	73.60	−9.38	0.221
7	75.98	−8.64	0.234
8	76.00	−9.82	0.207
9	76.00	−8.63	0.200
10	76.00	−8.16	0.241

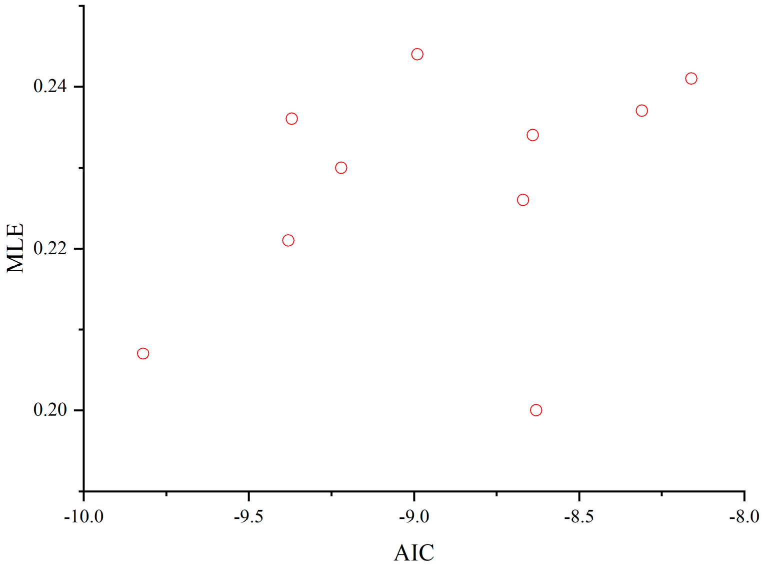


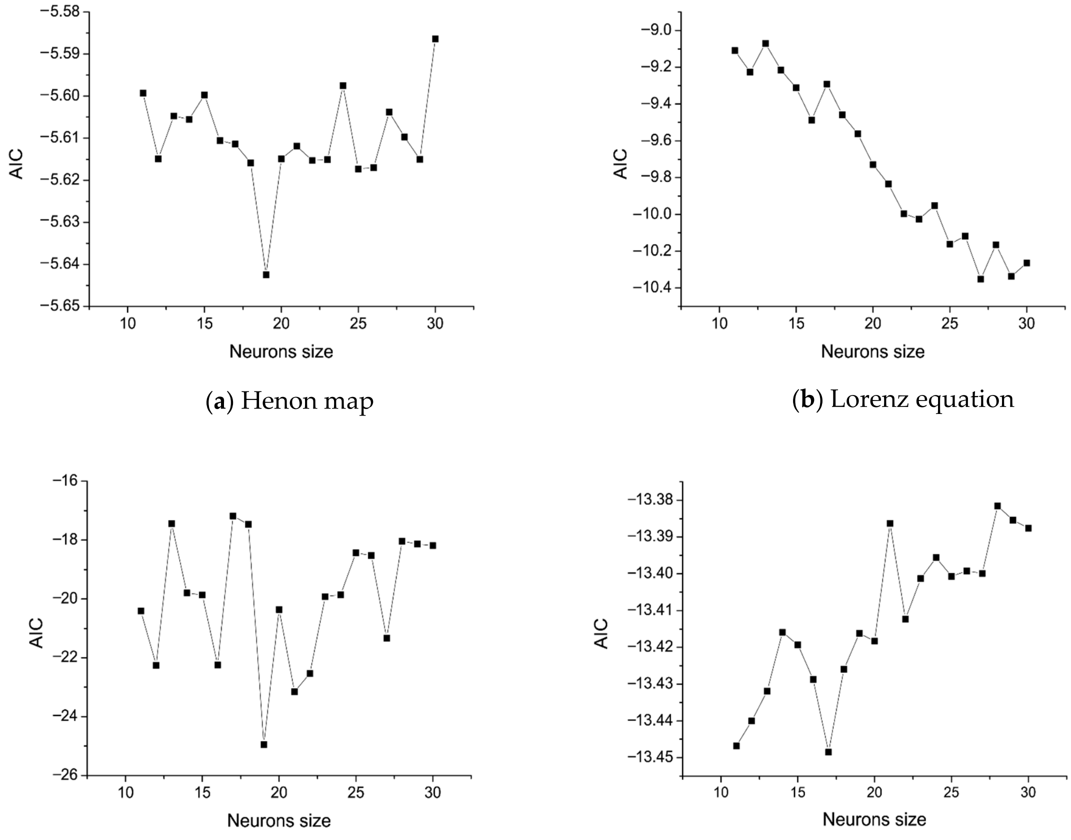
Figure 4. Scatterplot of AICg and MLE on 10 different networks.

4.5. Model Optimization

In order to find the optimal network topology for chaotic time series, we select 20 candidate network structures, that is, the number of neurons in the hidden layer from 11 to 30. Each network is trained by randomly selecting initial weights from 10 groups, and the model parameter with the smallest AIC_g was selected. Figure 5 shows the changes of AIC_g with hidden layer sizes on the Henon map, Lorenz equation, sunspot, and SST datasets, respectively. For example, as can be seen in Figure 5a, the smallest AIC_g value is obtained when the number of hidden layer nodes is 19, which is the selected optimal structure. Similarly, for the Lorenz equation, sunspots, and equatorial Pacific sea-surface temperature datasets, the smallest AIC_g values were obtained with the number of hidden nodes being 27, 19, and 17.

The maximal Lyapunov exponents (MLEs) are calculated by the above method for the four chaotic time series. Three MLP networks—one built by the optimal neuron size and the other two networks built by 20 and 25 neurons, respectively–are employed to estimate the MLE for the four datasets. The two compared networks are optimized by simply minimizing the loss function on MSE. The numerical results are listed in Figure 6. The numerical results show that the network model optimized by our proposed method is significantly better than the two network structures with a fixed number of hidden layer nodes of 20 and 25 on the Lorenz and sunspot datasets, slightly better than the other two types of networks on the Henon map dataset, roughly equivalent to the other two networks on the SST dataset, slightly better than the 25 hidden layer node network, and slightly worse than the 20 hidden layer node network results.

In order to further compare the performance of our proposed method, we investigate the multi-step prediction effect of the above three types of networks on the four datasets, namely, the optimized network by our proposed method, and two MLP networks with 20 and 25 neurons in the hidden layer, respectively. They are trained by the same dataset. The comparison results are shown in Figure 7. In Figure 7a, we show that the optimized MLP network provides significantly better prediction results than the other two MLP networks on the Henon map dataset; the optimized MLP network performs the best on all time steps. It also indicates that the AIC_g performs well in choosing a good MLP network for a chaotic time series. Figure 7a shows the results of the Lorenz dataset. Except for the fifth time step, the optimized MLP network performs the best among the three networks. The comparison results on two real-world time series are shown in Figure 7c,d. For the first five steps, the MLP networks can predict the time series quite well, but after five steps, the predicting errors become larger. The MLP network selected by the AIC_g also behaves much better than the other two MLP networks in nearly all cases. It also demonstrates the benefit of using the AIC_g to choose a good MLP network for a chaotic time series.



(**a**) Henon map

(**b**) Lorenz equation

(**c**) Sunspot

(**d**) SST

Figure 5. AIC_g curves of hidden layer scales on the datasets of the (**a**) Henon map, (**b**) Lorenz equation, (**c**) sunspot, and (**d**) SST.

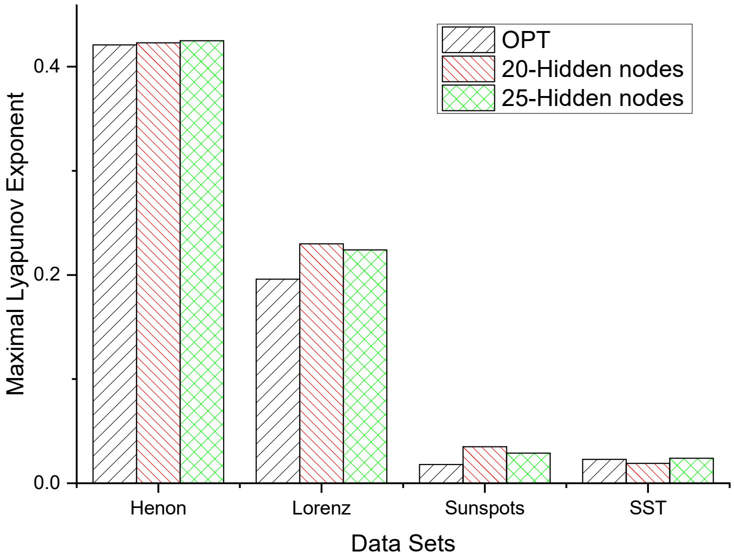
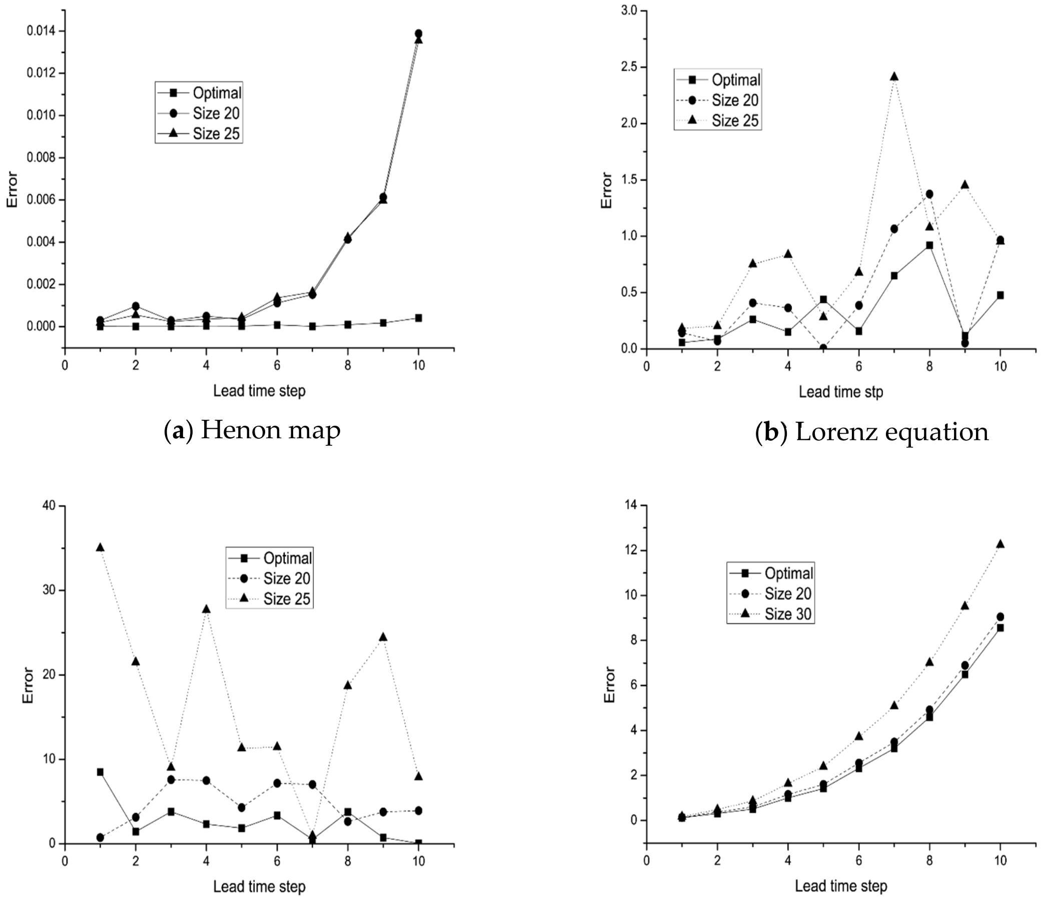


Figure 6. Lyapunov exponent estimation using the MLP model.



(**a**) Henon map

(**b**) Lorenz equation

(**c**) Sunspot

(**d**) SST

Figure 7. MSE error comparisons on datasets of the (**a**) Henon map, (**b**) Lorenz equation, (**c**) sunspot, and (**d**) SST.

5. Conclusions

Focusing on chaotic time series modeling, this paper proposes an MLP network-based framework. First, the generalized degree of freedom approximation method of MLP networks is derived, and then the Akaike information criterion ($\mathrm{AIC_g}$) can be calculated. Next, a multilayer perceptron network optimization method for chaotic time series analysis is designed according to $\mathrm{AIC_g}$. Finally, this paper proposes an overall framework for chaotic time series analysis. To verify the effectiveness of the proposed method, it is applied to four chaotic time series datasets, including two artificial datasets and two actual datasets. Experimental results show that the proposed method can effectively optimize the MLP network, and the selected model has obvious performance advantages. At the same time, the results show that the model selected in this paper can obtain good prediction performance on all four datasets. In future work, this approach could be extended to deep neural network models.

Author Contributions: Conceptualization, X.S. and Y.L.; methodology, X.S.; software, M.Q.; validation, M.Q.; formal analysis, A.T. and M.Q.; investigation, A.T. and X.S.; writing—original draft preparation, M.Q.; writing—review and editing, Y.L., A.T. and X.S.; visualization, M.Q.; funding acquisition, Y.L. and X.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the NSFC grant numbers 61972174 and 62272192, the Science-Technology Development Plan Project of Jilin Province grant number 20210201080GX, the Jilin Province Development and Reform Commission grant number 2021C044-1, the Guangdong

Universities' Innovation Team grant number 2021KCXTD015, and Key Disciplines Projects grant number 2021ZDJS138.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The open-access data that support the findings of this study are made publicly available at psl.noaa.gov/gcos_wgsp/Timeseries/SUNSPOT/ (accessed on 5 June 2023) and http://geodata.pku.edu.cn (accessed on 8 April 2023).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Alzaid, S.S.; Kumar, A.; Kumar, S.; Alkahtani, B.S.T. Chaotic behavior of financial dynamical system with generalized fractional operator. *Fractals* **2023**, *31*, 2340056. [CrossRef]
2. Yao, Q.; Jahanshahi, H.; Batrancea, L.M.; Alotaibi, N.D.; Rus, M.-I. Fixed-Time Output-Constrained Synchronization of Unknown Chaotic Financial Systems Using Neural Learning. *Mathematics* **2022**, *10*, 3682. [CrossRef]
3. Zhang, Y.; Li, P.; Li, H.; Zu, W.; Zhang, H. Short-Term Power Prediction of Wind Power Generation System Based on Logistic Chaos Atom Search Optimization BP Neural Network. *Int. Trans. Electr. Energy Syst.* **2023**, *2023*, 6328119. [CrossRef]
4. Wang, M.H.; Lu, S.D.; Liao, R.M. Fault Diagnosis for Power Cables Based on Convolutional Neural Network With Chaotic System and Discrete Wavelet Transform. *IEEE Trans. Power Deliv.* **2022**, *37*, 582–590. [CrossRef]
5. Abuturab, M.R. Multiple-information security system using key image phase and chaotic random phase encoding in Fresnel transform domain. *Opt. Lasers Eng.* **2020**, *124*, 105810. [CrossRef]
6. Liu, Z.Y.; Xia, T.C.; Wang, J.B. Fractional two-dimensional discrete chaotic map and its applications to the information security with elliptic-curve public key cryptography. *J. Vib. Control* **2017**, *24*, 4797–4824. [CrossRef]
7. Yang, G.C.; Yao, J.Y.; Dong, Z.L. Neuroadaptive learning algorithm for constrained nonlinear systems with disturbance rejection. *Int. J. Robust Nonlinear Control* **2022**, *32*, 5811–6215. [CrossRef]
8. Yang, G.C. Asymptotic tracking with novel integral robust schemes for mismatched uncertain nonlinear systems. *Int. J. Robust Nonlinear Control* **2023**, *33*, 1407–2507. [CrossRef]
9. Brandejsky, T. Specific modification of a GPA-ES evolutionary system suitable for deterministic chaos regression. *Comput. Math. Appl.* **2013**, *66*, 106–112. [CrossRef]
10. Senkerik, R.; Oplatková, Z.K.; Zelinka, I.; Chramcov, B.; Davendra, D.; Pluhacek, M. Utilization of analytic programming for the evolutionary synthesis of the robust multi-chaotic controller for selected sets of discrete chaotic systems. *Soft Comput.* **2014**, *18*, 651–668. [CrossRef]
11. Bezruchko, B.P.; Smirnov, D.A. Constructing nonautonomous differential equations from experimental time series. *Phys. Rev. E* **2000**, *63*, 016207. [CrossRef]
12. Lainscsek, C.S.M.; Letellier, C.; Schurrer, F. Ansatz library for global modeling with a structure selection. *Phys. Rev. E* **2001**, *64*, 016206. [CrossRef] [PubMed]
13. Han, P.; Zhou, S.L.; Wang, D.F. A Multi-objective Genetic Programming/NARMAX Approach to Chaotic Systems Identification. In Proceedings of the 6th World Congress on Intelligent Control and Automation (2006), Dalian, China, 21–23 June 2006; pp. 1735–1739.
14. Karimov, A.; Rybin, V.; Kopets, E.; Karimov, T.; Nepomuceno, E.; Butusov, D. Identifying empirical equations of chaotic circuit from data. *Nonlinear Dyn.* **2023**, *111*, 871–886. [CrossRef]
15. Yang, J.; Zhao, J.; Song, J.; Wu, J.; Zhao, C.; Leng, H. A Hybrid Method Using HAVOK Analysis and Machine Learning for Predicting Chaotic Time Series. *Entropy* **2022**, *24*, 408. [CrossRef] [PubMed]
16. Sarveswararao, V.; Ravi, V.; Vivek, Y. ATM cash demand forecasting in an Indian bank with chaos and hybrid deep learning networks. *Expert Syst. Appl.* **2023**, *211*, 118645. [CrossRef]
17. Raubitzek, S.; Neubauer, T. Taming the Chaos in Neural Network Time Series Predictions. *Entropy* **2021**, *23*, 1424. [CrossRef]
18. Yang, G.; Yao, J.; Ullah, N. Neuroadaptive control of saturated nonlinear systems with disturbance compensation. *ISA Trans.* **2022**, *122*, 49–62. [CrossRef]
19. Xu, Y.Q.; Zhen, X.X.; Tang, M. Dynamical System in Chaotic Neurons with Time Delay Self-Feedback and Its Application in Color Image Encryption. *Complexity* **2022**, *2022*, 2832104. [CrossRef]
20. Cheng, W.; Feng, J.B.; Wang, Y. High precision reconstruction of silicon photonics chaos with stacked CNN-LSTM neural networks. *Chaos* **2022**, *32*, 053112. [CrossRef]
21. Lee, J.; Huang, Z.; Lin, L.; Guo, Y.; Lee, R. Chaotic Bi-LSTM and attention HLCO predictor-based quantum price level fuzzy logic trading system. *Soft Comput.* **2022**. [CrossRef]
22. Nasiri, H.; Ebadzadeh, M.M. MFRFNN: Multi-Functional Recurrent Fuzzy Neural Network for Chaotic Time Series Prediction. *Neurocomputing* **2022**, *507*, 292–310. [CrossRef]

23. Lee, R.S. Chaotic Type-2 Transient-Fuzzy Deep Neuro-Oscillatory Network (CT2TFDNN) for Worldwide Financial Prediction. *IEEE Trans. Fuzzy Syst.* **2020**, *28*, 731–745. [CrossRef]
24. Chai, S.H.; Lim, J.S. Forecasting business cycle with chaotic time series based on neural network with weighted fuzzy membership functions. *Chaos Solitons Fractals* **2016**, *90*, 118–126. [CrossRef]
25. González-Zapata, A.M.; Tlelo-Cuautle, E.; Ovilla-Martinez, B.; Cruz-Vega, I.; De la Fraga, L.G. Optimizing Echo State Networks for Enhancing Large Prediction Horizons of Chaotic Time Series. *Mathematics* **2022**, *10*, 3886. [CrossRef]
26. Mohanty, S.; Dash, R. A novel chaotic flower pollination algorithm for modelling an optimized low-complexity neural network-based NAV predictor model. *Prog. Artif. Intell.* **2022**, *11*, 349–366. [CrossRef]
27. Wang, Y.F.; Fu, Y.C.; Xue, H. Improved prediction method of PV output power based on optimised chaotic phase space reconstruction. *IET Renew. Power Gener.* **2020**, *14*, 1831–1840. [CrossRef]
28. Xu, X.H.; Ren, W.J. A Hybrid Model Based on a Two-Layer Decomposition Approach and an Optimized Neural Network for Chaotic Time Series Prediction. *Symmetry* **2019**, *11*, 610. [CrossRef]
29. Ong, P.; Zainuddin, Z. Optimizing wavelet neural networks using modified cuckoo search for multi-step ahead chaotic time series prediction. *Appl. Soft Comput.* **2019**, *80*, 374–386. [CrossRef]
30. Feng, T.; Yang, S.Y.; Han, F. Chaotic time series prediction using wavelet transform and multi-model hybrid method. *J. Vibroeng.* **2019**, *21*, 1983–1999. [CrossRef]
31. Huang, F.; Yin, K.; Zhang, G.; Gui, L.; Yang, B.; Liu, L. Landslide displacement prediction using discrete wavelet transform and extreme learning machine based on chaos theory. *Environ. Earth Sci.* **2016**, *75*, 1376. [CrossRef]
32. Lu, R.; Lei, T. Short-term load forecast using maximum overlap discrete wavelet transform and BP neural network based on chaos theory. In Proceedings of the 2019 31st Chinese Control and Decision Conference (CCDC 2019), Nanchang, China, 3–5 June 2019; pp. 6029–6033.
33. Ramadevi, B.; Bingi, K. Chaotic time series forecasting approaches using machine learning techniques: A review. *Symmetry* **2022**, *14*, 955. [CrossRef]
34. Srinivasan, K.; Coble, N.; Hamlin, J.; Antonsen, T.; Ott, E.; Girvan, M. Parallel Machine Learning for Forecasting the Dynamics of Complex Networks. *Phys. Rev. Lett.* **2022**, *128*, 164101. [CrossRef] [PubMed]
35. Li, K.F.H.; Deng, P.F. Chaotic time series prediction using DTIGNet based on improved temporal-inception and GRU. *Chaos Solitons Fractals* **2022**, *159*, 112183.
36. Sangiorgio, M.; Dercole, F.; Guariso, G. Forecasting of noisy chaotic systems with deep neural networks. *Chaos Solitons Fractals* **2021**, *153*, 11570. [CrossRef]
37. Xie, H.L.; Zhang, L.; Lim, C.P. Evolving CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer. *IEEE Access* **2020**, *8*, 161519–161541. [CrossRef]
38. Huang, W.T.; Li, Y.T.; Huang, Y. Deep Hybrid Neural Network and Improved Differential Neuroevolution for Chaotic Time Series Prediction. *IEEE Access* **2020**, *8*, 159552–159565. [CrossRef]
39. Qian, Y.S.; Zeng, J.W.; Zhang, S.F. Short-Term Traffic Prediction Based on Genetic Algorithm Improved Neural Network. *Tech. Gaz.* **2020**, *27*, 1270–1276.
40. Chen, L.G.; Chiang, H.D.; Dong, N. Group-based chaos genetic algorithm and non-linear ensemble of neural networks for short-term load forecasting. *IET Gener. Transm. Distrib.* **2016**, *10*, 1440–1447. [CrossRef]
41. Kao, C.H.; Hsu, C.F.; Don, H.S. Design of an adaptive self-organizing fuzzy neural network controller for uncertain nonlinear chaotic systems. *Neural Comput. Appl.* **2012**, *21*, 1243–1253. [CrossRef]
42. Akaike, H. Information theory and the maximum likelihood principle. In *International Symposium on Information Theory*; Petrov, V., Csaki, F., Eds.; Akademiai Kiado: Budapest, Hungary, 1973.
43. Jayawardena, A.W.; Xu, P.C.; Tsang, F.L.; Li, W.K. Determining the structure of a radial basis function network for prediction of nonlinear hydrological time series. *Hydrol. Sci. J.* **2006**, *51*, 21–44. [CrossRef]
44. Xu, P.; Jayawardena, A.; Li, W.K. Model selection for RBF network via generalized degree of freedom. *Neurocomputing* **2013**, *99*, 163–171. [CrossRef]
45. Ye, J. On measuring and correcting the effects of data mining and model selection. *J. Am. Stat. Assoc.* **1998**, *93*, 120–131. [CrossRef]
46. Takens, F. Detecting Strange Attractors in Turbulence. In *Dynamical Systems and Turbulence, Warwick 1980*; Springer: Berlin/Heidelberg, Germany, 1981; pp. 366–381.
47. Chen, K.; Han, B.T. A survey of state space recognition of chaotic time series analysis. *Comput. Sci.* **2005**, *32*, 67–70. (In Chinese)
48. Graven, P.; Wahba, G. Smoothing noisy data with Spine function. *Numer. Math.* **1979**, *31*, 377–403.
49. Carruba, V.; Aljbaae, S.; Domingos, R.C.; Huaman, M.; Barletta, W. Chaos identification through the autocorrelation function indicator ACFI. *Celest. Mech. Dyn. Astron.* **2021**, *133*, 38. [CrossRef]
50. Kennel, M.B.; Brown, R.; Abarbanel, H.D.I. Determining embedding dimension for phase-space reconstruction using a geometrical construction. *Phys. Rev. A* **1992**, *45*, 3403. [CrossRef] [PubMed]
51. Cao, L.Y. Practical method for determining the minimum embedding dimension of a scalar time series. *Phys D* **1997**, *110*, 43–50. [CrossRef]
52. Uzal, L.C.; Grinblat, G.L.; Verdes, P.F. Optimal reconstruction of dynamical systems: A noise amplification approach. *Phys. Rev. E* **2011**, *84*, 016223. [CrossRef] [PubMed]

53. Keogh, E.J.; Chakrabarti, K.; Pazzani, M.J. Dimensionality reduction for fast similarity search in large time series databases. *Knowl. Inf. Syst.* **2001**, *3*, 263–286. [CrossRef]
54. Chen, Z.; Zuo, W.; Hu, Q.; Lin, L. Kernel sparse representation for time series classification. *Inf. Sci.* **2015**, *292*, 15–26. [CrossRef]
55. Rosas-Romero, R.; Diaz-Torres, A.; Etcheverry, G. Forecasting of stock return prices with sparse representation of financial time series over redundant dictionaries. *Expert Syst. Appl.* **2016**, *57*, 37–48. [CrossRef]
56. Kotteti, C.M.M.; Dong, X.S.; Qian, L.J. Ensemble deep learning on time-series representation of tweets for rumor detection in social media. *Appl. Sci.* **2020**, *10*, 7541. [CrossRef]
57. Li, Q.; Xu, P. Estimation of Lyapunov spectrum and model selection of a chaotic time series. *Appl. Math. Model.* **2012**, *36*, 6090–6099. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.