AIMS Energy, 14(1): 1–22.
DOI: 10.3934/energy.2026001
Received: 11 August 2025
Revised: 21 November 2025
Accepted: 08 December 2025
Published: 5 January 2026

https://www.aimspress.com/journal/energy

Research article

A meta-inspired algorithm-based fuzzy model for wind speed prediction using chaotic time series and phase space reconstruction

Jinfeng Lv[1], Yaxue Ren[2,*], Hongjuan Zhang[2] and Meng Han[3]

[1] School of Mathematics, Hebei Normal University of Science and Technology, Qinhuangdao, Hebei 066004, China
[2] Tangshan University, Tangshan, Hebei 063000, China
[3] College of Information Science and Engineering, Northeastern University, Shenyang 110819, China

* **Correspondence:** Email: renyaxue_int@tsc.edu.cn; Tel: +18331065860.

Abstract: This study proposes a phase space reconstruction-based fuzzy identification method to improve the prediction accuracy of chaotic time series. First, the delay time and embedding dimension are determined using mutual information and the Cao method, respectively, to reconstruct the phase space. Second, the particle swarm optimization algorithm is applied to optimize the membership function parameters of the T-S fuzzy model, enabling a more rational partition of the premise space. Finally, the conclusion parameters of the fuzzy model are identified via the recursive least squares method. Experimental validation using two benchmark chaotic time series and a real-world wind speed series demonstrates that the proposed model achieves high prediction accuracy. The results confirm the effectiveness and practical applicability of the method for modeling and forecasting chaotic systems.

Keywords: fuzzy model; chaotic time series; wind speed prediction; system identification

1. Introduction

Chaotic systems, characterized by non-periodicity and pseudo-randomness, have significant research value in mechanical control, energy development, and signal processing [1]. In energy systems, the output processes of renewable energy sources—such as wind speed and solar irradiance—often exhibit chaotic behavior, and their evolution directly affects wind turbine operational safety, grid integration

capability, and overall scheduling efficiency. Existing studies, such as model-free adaptive chaotic control [2] and chaotic dynamics–based low-power energy harvesters [3], demonstrate that effective modeling and controlling of chaotic dynamics are essential for improving the performance of energy systems.

Chaotic time series, such as in wind speed and irradiance data, contain dynamic features arising from the coupling between system operation and external environmental factors. Their prediction has therefore become a fundamental task in smart energy systems [4]. Current prediction approaches include local models, global neuro-fuzzy models, machine learning–based methods, and deep learning architectures such as LSTM. Farmer and Sidorowich [5] pioneered a foundational method for chaotic time series prediction. Chen and Wang [6] proposed a modified random forest method to enhance the explainability of neural networks in time series-related cycle time prediction. Jin et al. [7] developed a time-varying fuzzy parameter zeroing neural network for chaotic system synchronization. Pourafzal et al. [8] designed a deep learning model inspired by complex systems for chaotic time series recognition. Wang et al. [9] constructed a dilated convolution-LSTM model for multi-step chaotic time series prediction. Zhu and Su [10] proposed a combined method of chaotic linear regression and Elman neural network. Huang et al. [11] optimized chaotic time series prediction via a deep hybrid neural network and improved differential neuroevolution. Huang et al. [12] introduced a hybrid neural network with attention mechanism for chaotic time series prediction. Although deep models can improve prediction accuracy, their large structure and high computational cost make them difficult to deploy in real-time on resource-constrained devices such as field controllers and edge nodes in wind farms. This contradicts the smart energy system requirement for both accurate and computationally efficient forecasting.

Compared with deep models, the Takagi-Sugeno (T-S) fuzzy model is lightweight and interpretable and has shown clear advantages in nonlinear system identification and energy equipment modelin. Xiao et al. [13] proposed an interval type-2 fuzzy-model-based control design with membership-functions-dependent analysis. Wang et al. [14] developed Fuzzy-NMS to improve 3D object detection via fuzzy classification in NMS. Singh and Bhardwaj [15] reviewed fuzzy soft set theory applications and future directions in medical diagnosis. However, the traditional T-S model suffers from two major limitations: (1) The centers and widths of membership functions are fixed, making the model unable to adapt to strong fluctuations and non-stationarity in chaotic wind speed signals [16], and (2) data preprocessing is often ignored, with the input assumed to be "idealized", which prevents the model from capturing the complex dynamics induced by seasonal variation, terrain effects, and turbulence [17]. Furthermore, many deep or hybrid models rely heavily on structural stacking to improve accuracy, typically applying only basic normalization while neglecting key chaotic feature extraction methods such as phase-space reconstruction [12,18]. As a result, these models fail to meet the real-time stability and scheduling requirements of modern energy systems.

Wind speed forecasting is a core task in smart wind power systems, directly influencing wind power output prediction, grid scheduling, and renewable energy utilization. Due to the strong nonlinearity, multi-scale characteristics, and spatial heterogeneity caused by terrain, turbulence, and atmospheric circulation, traditional linear models such as ARIMA cannot capture chaotic wind dynamics [4], leading to deviations in power forecasting and dispatch mismatches. More critically, wind-farm cluster operation requires minute-level forecast updates, but field devices often lack the computational resources required for complex neural networks. These challenges highlight the need for forecasting models that are both sufficiently accurate and structurally lightweight [3,19].

To address these issues, this study proposes a lightweight chaotic time series prediction framework for smart energy systems by integrating phase-space reconstruction (PSR) and particle swarm optimization (PSO) into a simplified T-S fuzzy model. The main contributions are as follows:

(1) High-quality preprocessing for energy data: PSR is used to extract key dynamic features reflecting the coupling among terrain, turbulence, and temporal variations, providing more physically meaningful and predictive inputs for wind power forecasting.

(2) Improved adaptability to chaotic wind speed: PSO optimizes the centers and widths of the membership functions, enabling the T-S model to adapt to non-stationary wind speed patterns across different seasons and wind farm environments, thereby enhancing model robustness and generalization.

(3) Lightweight design for smart energy systems: The number of model parameters is approximately one-fifth that of traditional neuro-fuzzy models, supporting low-cost deployment and real-time prediction in smart energy systems.

2. Materials and methods

2.1. Data preprocessing

Many components are involved in the study of time series, and in a dynamic system, these influencing factors' dynamic equations are frequently nonlinear and more likely to be chaotic. Reconstructing the phase space of the sequence is a crucial way to investigate the state of the dynamical system where the sequence is placed for time series that display chaotic features. So, before creating and refining the prediction model, the chaotic time series' phase space reconstruction should be carried out. The original sequence and rebuilt phase space data are also normalized to remove the dimension and increase the model's convergence speed and forecast accuracy.

2.1.1. Phase space reconstruction technique

The essence of PSR lies in converting the "temporal correlation" of one-dimensional sequences into the "spatial correlation" of high-dimensional vectors. This transformation enables the reconstruction of the hidden dynamic trajectory of the wind speed system and the restoration of potential coupling relationships between unobserved influencing factors. Wind speed time series exhibits a typical chaotic nature: apparent randomness in the time domain and intrinsic determinism in the dynamic system. PSR targets this dual property through parameter optimization and structural design, achieving precise feature capture:

1. Mining deterministic laws via time delay τ optimization

The "pseudo-randomness" of wind speed stems from the overlapping of multi-factor dynamic processes, while its "determinism" is reflected in the temporal correlation between consecutive data points. To retain this correlation without information redundancy, τ is determined using the mutual information method (instead of the traditional autocorrelation function). Mutual information quantifies the statistical dependence between $x(t)$ and $x(t+\tau)$; selecting the first local minimum of the mutual information curve ensures that $x(t+\tau)$ contains the maximum new information about the system state while maintaining temporal association. For wind speed data, this means preserving the memory of short-term fluctuations (e.g., 10-minute scale variations) and long-term trends (e.g., diurnal cycles), thereby uncovering the deterministic evolution rules hidden beneath random fluctuations.

2. Matching system degrees of freedom via embedding dimension m selection

The wind speed system is a multi-degree-of-freedom dynamic system driven by terrain, turbulence, and thermal effects. An inappropriate m will either lead to "trajectory confusion" (low m, unable to distinguish different system states) or "dimension redundancy" (high m, increasing computational complexity without improving feature quality). Here, m is determined using the Cao method, which evaluates the continuity of neighboring points across different embedding dimensions—when the continuity metric stabilizes, the corresponding m is considered to match the true degrees of freedom of the wind speed system.

3. Non-destructive feature enhancement for chaotic signals

Traditional preprocessing methods aim to eliminate "noise" by suppressing high-frequency fluctuations, but this inadvertently destroys the intrinsic chaotic characteristics of wind speed. In contrast, PSR achieves feature enhancement through dimensional transformation without altering the original data distribution; by mapping one-dimensional wind speed data to an m-dimensional phase space, the overlapping trajectories in the time domain are unfolded into distinguishable geometric structures. For strong chaotic wind speed signals, this unfolding effect makes the nonlinear relationships between variables visually and mathematically expressible, laying a foundation for subsequent prediction model training.

To reconstruct the phase space, the mutual information method [20] is employed to determine the delay time τ, and the embedding dimension m is identified using the Cao method [21].

2.1.1.1. Mutual information

The mutual information method is a key approach for determining the delay time in time series analysis. Consider a typical system consisting of two discrete information systems, $S = s_1, s_2, ..., s_n$ and $Q = q_1, q_2, ..., q_m$. The mutual information $I(Q,S)$ quantifies the amount of information about the system Q that can be obtained given knowledge of S.

$$I(Q,S) = \sum_i \sum_j P_{sq}(s_i, q_j) log_2[\frac{P_{sq}(s_i, q_j)}{P_s(s_i) P_q(q_j)}] \tag{1}$$

Here, $P_{sq}(s_i, q_j)$ represents the joint probability distribution of events s_i and q_j. By defining $[s,q] = [x(t), x(t+\tau)]$, the mutual information becomes a function of the delay time, denoted as $I(\tau)$. The value of $I(\tau)$ indicates the degree of determinism in the system Q, represented as $x(t+\tau)$, given that system S is $x(t)$. If $I(\tau)$ equals zero, it signifies that $x(t+\tau)$ is entirely unpredictable, indicating no correlation between $x(t)$ and $x(t+\tau)$. The first minimum of $I(\tau)$ represents the point of maximum irrelevance between $x(t)$ and $x(t+\tau)$, making it the optimal delay time for reconstruction. As described, mutual information can also evaluate the correlation between two nonlinear sequences. In discrete systems, the delay time is typically set to 1, whereas in continuous systems, it is generally greater than 1.

2.1.1.2. Cao method

The Cao method is an enhanced version of the pseudo-nearest neighbor approach. In a d-dimensional phase space, each phase point vector $\mathbf{Z}(i)$ has a nearest neighbor $\mathbf{Z}N(i)$ within a specific distance, denoted as $R_d(i)$. When the phase space dimension increases from d to $d+1$, the distance between

these two phase points changes to $R_{d+1}(i)$. If $R_{d+1}(i)$ becomes significantly larger than $R_d(i)$, it suggests that two non-adjacent points in the high-dimensional chaotic attractor appear adjacent when projected onto the low-dimensional space. Such neighbors are thus identified as spurious.

Let

$$a(i,d)=\frac{\mathbf{Z}_{d+1}(i)-\mathbf{Z}_{Nd+1}(i)}{\mathbf{Z}_d(i)-\mathbf{Z}_{Nd}(i)} \tag{2}$$

Define

$$\begin{aligned} E(m)&=\frac{1}{N-m\tau}\Sigma_{i=1}^{N-m\tau}a(i,m)\\ E_1(m)&=\frac{E(m+1)}{E(m)} \end{aligned} \tag{3}$$

For a deterministic time series, an embedding dimension exists such that $E_1(m)$ remains constant once m exceeds a certain fixed value m_0. In contrast, for a random signal, $E_1(m)$ will continue to increase gradually.

In fact, for finite-length sequences, it is difficult to judge whether $E_1(m)$ is changing slowly or has stabilized, so the supplementary judgment criteria is as shown in Eq (4).

$$\begin{aligned} E^*(m)&=\frac{1}{N-m\tau}\Sigma_{i=1}^{N-m\tau}\|\mathbf{Z}(i+m\tau)-\mathbf{Z}_N(i+m\tau)\|\\ E_2(m)&=\frac{E^*(m+1)}{E^*(m)} \end{aligned} \tag{4}$$

For random sequences, there is no correlation between the data, and $E_2(m)$ will always be 1. For deterministic sequences, the correlation depends on the value of m in the embedding dimension, so there are always some values of m such that $E_2(m)$ is not equal to 1. The distance and fluctuation of $E_2(m)$ from 1 can be used as a measure of the deterministic component in the signal. The larger the noise amplitude, the closer $E_2(m)$ is to 1; the smaller the fluctuation range, the stronger the randomness of the sequence.

2.1.2. Data normalization

In model prediction, normalizing the dataset, especially the nonlinear regression dataset, is a very necessary preprocessing step. Because normalization can not only eliminate the dimension and unify the data to the same scale but also improve the convergence speed and prediction accuracy of the model, this paper employs normalization to standardize the phase space domain dataset and the original chaotic time series dataset within the range (0,1). The normalization process is described by Eq (5).

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{5}$$

2.2. T-S fuzzy model

The T-S fuzzy model [22] (Takagi-Sugeno fuzzy model) is a mathematical framework based on fuzzy rules, designed to describe complex nonlinear systems. It combines fuzzy logic with local linear modeling, decomposing nonlinear systems into multiple local linear subsystems through a set of fuzzy rules, enabling system analysis and control. The T-S fuzzy model features strong interpretability and computational efficiency, making it widely applicable in control system design, predictive modeling, and other fields.

$$\begin{aligned} &R^i : \text{IF } x_1(k) is A_1^i \text{AND} ... \text{AND} x_r(k) is A_r^i \\ &\text{THEN } y^i(k) = a_0^i + a_1^i x_1(k) + ... + a_r^i x_r(k) \end{aligned} \tag{6}$$

where $x_i (i = 1, 2, ..., r)$ is the input variable, A_j^i is a fuzzy membership function that represents a fuzzy set of input variables, y^i is the output variable and represents the form of a linear function, and a_j^i is the parameter of the linear model, which is determined by the system modeling process.

The overall output of the T-S model is the weighted average of all the rule outputs

$$\hat{y}(k) = \frac{\sum_{i=1}^{c} w_i(k) y^i(k)}{\sum_{i=1}^{c} w_i(k)} \tag{7}$$

where $\omega_i = \Pi_{j=1}^{n} \mu_{A_{ij}}(x_j)$, $\mu_{A_{ij}}(x_j)$ is the membership of the fuzzy set A_{ij} corresponding to input x_j, and c is the total number of rules.

The following are the specific calculation steps of T-S fuzzy model:

(1) Determine the input and output variables

Define the input variables of the system $x_i (i = 1, 2, ..., r)$ and the output variable y^i.

(2) Construct fuzzy membership function

Fuzzy set $A_q^i(x_q(k))$ and its corresponding membership function $\mu_{A_{ij}}(x_j)$ are defined for each input variable. Common membership functions include trigonometric, trapezoidal, and Gaussian functions.

(3) Fuzzy rule making

Establish a number of "if-then" rules based on system characteristics or expert experience.

(4) Determine the linear model parameters

Using the input and output data, the linear model parameters for each rule are determined using optimization methods such as least squares a_j^i.

(5) Calculate membership value

For each rule, the membership function value $\mu_{A_{ij}}(x_j)$ is computed based on the input variable.

(6) Calculate the weight ω

The weight of each rule is calculated according to the membership value of the rule premise part: $\omega_i = \Pi_{j=1}^{n} \mu_{A_{ij}}(x_j)$.

(7) Weighted calculated output

The system output is calculated by combining the rule weight \$\omega\$ and the linear output y^i.

2.3. Particle swarm optimization

Particle swarm optimization (PSO) is a population-based random optimization algorithm inspired by group behaviors such as bird feeding or fish swimming. PSO is widely used to solve optimization problems by simulating cooperation and competition among individuals and searching for the global optimal solution.

In PSO, each particle represents a potential solution in the search space, and the position and velocity of the particle are iteratively updated by the formula until convergence conditions are met.

Let the position of the I-th particle in the D-dimensional search space be $\mathbf{x}_i = [x_{i1}, x_{i2}, ..., x_{iD}]$ and the velocity $\mathbf{v}_i = [v_{i1}, v_{i2}, ..., v_{iD}]$. The formula for updating the velocity and position of the particle is:

$$\begin{aligned} v_{id}(t+1) &= \omega v_{id}(t) + c_1 r_1 (p_{id} - x_{id}(t)) + c_2 r_2 (g_d - x_{id}(t)) \\ x_{id}(\mathrm{t+1}) &= x_{id}(\mathrm{t}) + v_{id}(\mathrm{t+1}) \end{aligned} \tag{8}$$

where t is the current number of iterations. ω is the inertia weight, which controls global and local search capabilities. c_1 and c_2 are learning factors (acceleration constants), which represent the dependence of the particle on its own experience and the group experience, respectively. r_1 and r_2 are random numbers evenly distributed in [0,1]. p_{id} is the historical optimal position of the particle itself. g_d is the globally optimal position for all particles.

The fitness function $f(x)$ evaluates the current position of each particle, with the goal of finding the optimal solution position x^* that minimizes or maximizes $f(x)$. The algorithm stops according to one of the following conditions: the maximum number of iterations is reached, or the fitness function value meets the preset optimal threshold.

The flow of the PSO algorithm is shown in Algorithm 1.

Algorithm 1 Meta-heuristic particle swarm optimization algorithm

Initialization: Each particle gets a random solution and a random velocity;

For: For all particles at position X_{id}, calculate the new position X'_{id};

1: Calculate the difference $A = P_{id} - X_{id}$ between P_{id} and X_{id}, where A is the Basic Swap Sequence;

2: Compute $B = P_{gd} - X_{id}$, where B is the Basic Swap Sequence;

3: Calculate the new velocity V'_{id} according to the velocity update formula, and convert V'_{id} into a Basic
Swap Sequence;

4: Calculate the new solution $X'_{id} = X_{id} + V_{id}$ (that is, V_{id} role in X_{id});

5: Update P_{id} if the new solution is better;

6: Update P_{gd} if a new global best solution appears;

7: End For;

8: **return** P_{gd};

2.4. The proposed identification method

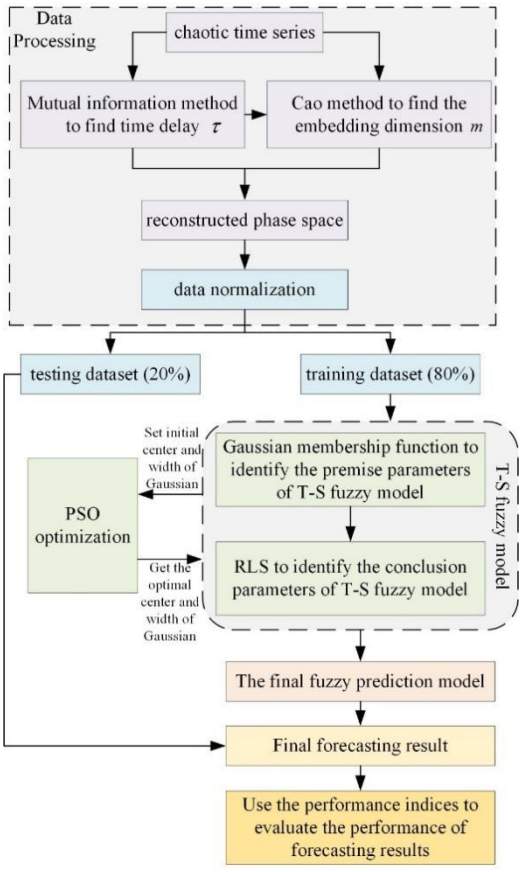

Figure 1. Phase space reconstruction-based T-S fuzzy model for chaotic time series prediction.

The form of the Gaussian membership function is better suited for fuzzy identification, and the choice of the two parameters (center and width) of the Gaussian function is more crucial, as can be observed from fuzzy identification approaches in the literature [23] and [24]. Particle swarm optimization (PSO), which concurrently optimizes the center and width of the Gaussian function, is utilized in the literature to increase the model identification accuracy. The chaotic time series' phase space is first rebuilt in this study, and the resulting data set is then normalized. The second step is model optimization. A Gaussian membership function divides the model's premise space, and the PSO algorithm optimizes the parameters to maximize the premise space. The recursive least squares approach is used to find the parameters of the model conclusion. Figure 1 depicts the chaotic sequence prediction model based on the phase space reconstruction developed in this work.

2.4.1. Advantages of the proposed method compared with existing methods

To highlight the innovation and superiority of the proposed method, a comparative analysis with representative existing approaches is conducted, focusing on three core dimensions: data preprocessing, parameter optimization, and model structure, as detailed below:

1. Enhanced data preprocessing with phase space reconstruction: Traditional T-S fuzzy models (e.g., Ren et al. [17]) directly use raw chaotic time series for modeling without targeted data preprocessing, leading to insufficient extraction of hidden dynamic features. In contrast, the proposed method integrates two key techniques for phase space reconstruction: the mutual information method (to determine the optimal delay time τ) and the Cao method (to determine the embedding dimension m). This reconstruction transforms one-dimensional time series into high-dimensional phase space points, which topologically preserve the dynamic characteristics of the original chaotic system [4], effectively solving the problem of "feature sparsity" in conventional fuzzy models.
2. Efficient and stable premise parameter optimization: Existing fuzzy model optimization methods rely on differential evolution (DE) [19] or ant colony algorithms [25], which often suffer from slow convergence or trapping in local optima. The proposed method adopts PSO to optimize the center and width of the Gaussian membership function, with two key improvements: (1) the learning factors c_1 and c_2 are set to 2, balancing the dependence of particles on individual and group experience, and (2) the inertia weight ω is updated via a linear decreasing strategy (Eq (9)), transitioning from global exploration (high ω) to local exploitation (low ω) during iterations. This design enables faster convergence and more stable parameter tuning compared to DE and ant colony algorithms.
3. Lightweight structure with high interpretability: Hybrid neural network models (e.g., Att-CNN-LSTM [12]) achieve high accuracy by stacking convolutional, recurrent, and attention layers, but this results in complex architectures, high computational costs, and poor interpretability (black-box models). The proposed method retains the T-S fuzzy model's rule-based interpretability (each fuzzy rule corresponds to a local linear subsystem) while simplifying the optimization process—only the membership function parameters and conclusion linear parameters need to be adjusted. It achieves a balance between prediction accuracy and computational efficiency, avoiding the "over-engineering" issue of hybrid neural networks.

2.4.2. Theoretical analysis of the reliability of output results

The reliability of the proposed method's prediction results is theoretically guaranteed through three layers of validation: the validity of phase space reconstruction, the convergence of PSO optimization, and the unbiasedness of RLS-based parameter estimation.

1. Validity of phase space reconstruction (based on Takens' embedding theorem): Takens' theorem [4] states that for a smooth dynamic system with an attractor of dimension d, there exists an embedding dimension $m \geq 2d+1$ such that the phase space reconstructed by delaying the scalar time series is topologically isomorphic to the original system's state space. In this method:
   - The mutual information method selects the first minimum of $I(\tau)$ as the delay time τ, ensuring that the delayed time series components are weakly correlated but retain dynamic information [20];
   - The Cao method determines the embedding dimension m by monitoring the stability of $E_1(m)$ and $E_2(m)$, ensuring $m \geq 2d+1$ [21]. Together, these steps guarantee that the reconstructed phase space accurately reflects the dynamic behavior of the original chaotic system, providing reliable input features for the fuzzy model.
2. Convergence of PSO optimization: The convergence of the PSO algorithm in this method is proven based on the analysis in Ren et al. [26]. The inertia weight ω is updated as: $\omega(m) = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{k_{\max}} \times k$, where $\omega_{\max} = 0.9$, $\omega_{\min} = 0.4$, and $k_{\max}$ is the maximum number of iterations. This linear decreasing strategy ensures that:
   - In the early iteration stage (large ω), particles explore the global search space to avoid local optima.
   - In the late iteration stage (small ω), particles converge to the global optimal solution. For a bounded search space (the center and width of the Gaussian membership function have physical constraints), the PSO algorithm converges to the global optimal solution with a probability of 1 [22], ensuring the optimality of the premise parameters.
3. Unbiasedness of RLS-based conclusion parameter estimation: The recursive least squares (RLS) method is used to estimate the linear parameters a_{i0}, a_{i1}, ..., a_{in} in the T-S fuzzy model's conclusion (Eq (6)). For the output of the i-th fuzzy rule $y_i = a_{i0} + a_{i1}x_1 + ... + a_{in}x_n$, the RLS estimator minimizes the sum of squared residuals $\sum_{p=1}^{P}(y_p - \hat{y}_p)^2$. Under the assumption that the model noise is white Gaussian noise with zero mean, the RLS estimator satisfies: $E[\hat{\theta}] = \theta$, where $\hat{\theta} = [a_{i0}, a_{i1}, ..., a_{in}]^T$ is the estimated parameter vector, and θ is the true parameter vector. This indicates that the RLS estimator is unbiased. As the sample size P increases, the estimator asymptotically converges to the true value [22], ensuring the reliability of the model's output.

2.4.3. Detailed implementation steps

Step 1: Phase space reconstruction

Let $x_1, x_2, ..., x_M$ represent a time series of length M with a delay time of τ and an embedding dimension of m. Based on these parameters, the phase space domain data set can be expressed as $\bar{D} = X(t), Y(t), t = 1, 2, ..., N$, where $N = M - (m-1)\tau$. Here,

$\mathbf{X}(t)=[X_t, X_{t+\tau},..., X_{t+(m-1)\tau}]$. The chaotic time series is then divided, yielding the following input and output matrices.

$$\mathbf{X}=\begin{bmatrix} x_1 & x_{1+\tau} & \cdots & x_{1+(m-1)\tau} \\ x_2 & x_{2+\tau} & \cdots & x_{2+(m-1)\tau} \\ \mathrm{M} & \mathrm{M} & \mathrm{M} & \mathrm{M} \\ x_M-1 & x_{M-1+\tau} & \cdots & x_{M-1+(m-1)\tau} \end{bmatrix}$$

$$\mathbf{Y}=\begin{bmatrix} x_{2+(m-1)\tau} & x_{3+(m-1)\tau} & \cdots & x_{M+(m-1)\tau} \end{bmatrix}^T$$

Step 2: Data normalization and building the dataset

Analyze and normalize the reconstructed phase space data obtained in the initial step. The normalized dataset is then split into two portions: the first 80% is designated as the training set, while the remaining 20% is allocated for testing purposes.

Step 3: Setting the PSO algorithm's characteristics and optimizing the premise parameters

The learning factors $(c1, c2)$ in the PSO algorithm are both assigned a value of 2. To prevent the algorithm from stagnating in a local optimum, the inertia weight ω is updated using a non-random approach. The update rule for ω is defined as follows:

$$\omega=\omega_{min}+D\cdot\frac{\omega_{max}-\omega_{min}}{MD} \tag{9}$$

Here, D represents the number of iterations. The maximum iteration count is set to $MD=200$, with $\omega_{min}=0.4$ and $\omega_{max}=0.9$. Once the parameter setup is finalized, the optimal membership value for the model is determined by obtaining the two parameters (center and width) of the Gaussians.

Step 4: Obtain the prerequisite membership function $\mu_{A_{ij}}$

The formula for this is given in Eq (10), which represents the membership function used for chaotic time series forecasting with the T-S fuzzy model. By optimizing the two parameters in Step 2, a more accurate membership function can be generated. In Step 3, the parameters α and β from Eq (10) are optimized to α' and β', where both α' and β' are $r*c$ matrices. The optimized formula is shown in Eq (11). The visual representation of the changes in the membership function before and after optimization is provided in Figure 2.

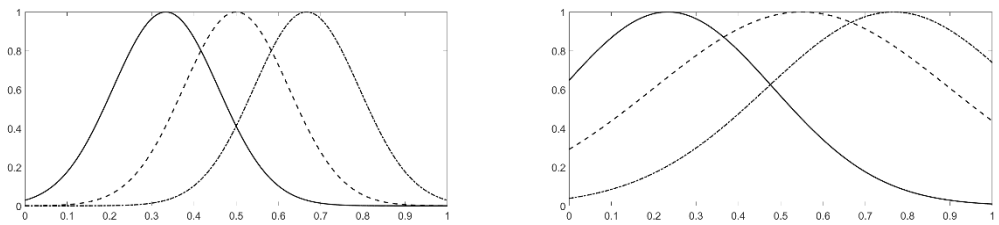

Figure 2. Variation in the membership function distribution.

$$\mu_{A_{ij}}(x_q(k)) = exp\{-(\frac{x_q(k)-\alpha_q^i}{\beta_q^i})^2\}$$
$$i=1,2,...,c$$
$$q=1,2,...,r \tag{10}$$

$$\mu_{A_{ij}}(x_q(k)) = exp\{-(\frac{x_q(k)-\alpha_q'^i}{\beta_q'^i})^2\}$$
$$i=1,2,...,c$$
$$q=1,2,...,r \tag{11}$$

where c is the quantity of fuzzy rules and r is the number of input variables.

Step 5: Calculate the weight ω

The weight of each rule is calculated according to the membership value of the rule premise part: $\omega_i = \Pi_{j=1}^n \mu_{A_{ij}}(x_j)$.

Step 6: Weighted calculated output

The system output is calculated by combining the rule weight ω and the linear output y^i.

Step 7: Calculation of performance index

In this stage, the model's prediction accuracy must be calculated and assessed using the mean squared error (MSE). If the accuracy is sufficient, the calculation should be finished; otherwise, add 1 to c and input step 2 again.

$$MSE = \frac{1}{N}\sum_{k=1}^{N}(y(k)-\hat{y}(k))^2 \tag{12}$$

In the equation, k refers to the k th data pair, and N indicates the total number of data samples. The k th original model output is represented by $y(k)$, while $\hat{y}(k)$ denotes the model output for the k th data point.

3. Results and discussion

Experiments are carried out using logistic, Lorenz chaotic, and wind speed time series to validate the predictive performance of the fuzzy prediction model for chaotic time series based on phase space reconstruction. Several evaluation metrics are applied to assess prediction errors. The root mean square error (RMSE) is used to quantify the dispersion within a dataset and the deviation between predicted and actual values. The mean absolute error (MAE) is employed to avoid error cancellation, providing a clear representation of the true forecast error magnitude. The mean absolute percentage error (MAPE), a common statistical measure, is utilized to assess the accuracy of predictions, including those for time series. In addition to the three aforementioned indicators, the root mean square percentage error (RMSPE), a metric often used to evaluate time series forecasting performance, is also introduced.

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}(y(k)-\hat{y}(k))^2} \tag{13}$$

$$MAE = \frac{1}{N}\sum_{k=1}^{N}\| y(k)-\hat{y}(k)\| \tag{14}$$

$$MAPE = \frac{1}{N}\sum_{k=1}^{N}\frac{100\times\| y(k)-\hat{y}(k)\|}{y(k)} \tag{15}$$

$$MAPE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\|\frac{y(k)-\hat{y}(k)}{y(k)}\|^2}\times 100 \tag{16}$$

where k indicates the k-th data pair, and N represents the total number of samples. $y(k)$ refers to the original output of the model for the k-th data point, while $\hat{y}(k)$ denotes the model output corresponding to that same point.

3.1. Modeling and prediction of logistic chaotic time series

The logistic chaotic mapping equation is

$$x_{n+1} = \mu x_n(1-x_n) \tag{17}$$

There is just one fixed point, $x_0 = 0$, and x_0 is an appealing fixed point; therefore, the dynamic behavior of the logistic system produced by Eq (17) is quite simple when $0 < \mu \le 1$, at which time the system is in a static state. There are two fixed points, 0 and $1-1/\mu$, where 0 is the repellent fixed point, and $1-1/\mu$ is the attracting fixed point; for μ, the dynamic behavior of the logistic system described by Eq (17) is rather simple. The system only has one stable state for each value of the determined parameter μ, which is a stable period 1 state, and the value of the stable point rises as the parameter μ grows. When $3 \le \mu \le 4$, the dynamic behavior of the logistic system derived from Eq (17) changes dramatically, going from period doubling to a mild chaotic state to a complete chaotic state. In this paper, the equation's initial value is set as $x_0 = 0.32$, with $\mu = 3.8$. To maintain chaotic behavior, the first 1000 data points are excluded, and the remaining 3000 data points are utilized as model data samples.

Using the mutual information method and the Cao method, the delay time τ and the embedding dimension m of the above sample data are obtained, and $\tau = 16$ and $m = 11$ are obtained. The data after phase space reconstruction has 2839 valid phase points, of which 2271 are used as training data sets and the rest are used as test data sets. After data processing and model optimization, the prediction results are obtained. The prediction results and prediction errors are shown in Figure 3. In order to directly reflect the superiority of the prediction model proposed in this paper, Table 1 lists the comparison of the prediction results of the chaotic time series prediction model based on the T-S fuzzy model and the models proposed by other scholars. The control model used in this paper, known as

TSFS-PSO, does not perform phase space reconstruction of time series but instead only uses PSO algorithm to optimize parameters in the T-S fuzzy model in order to better reflect the combination of phase space reconstruction and fuzzy model. The TS-PAR model represents a model that does not use the PSO optimization algorithm but uses phase space reconstruction for data preprocessing.

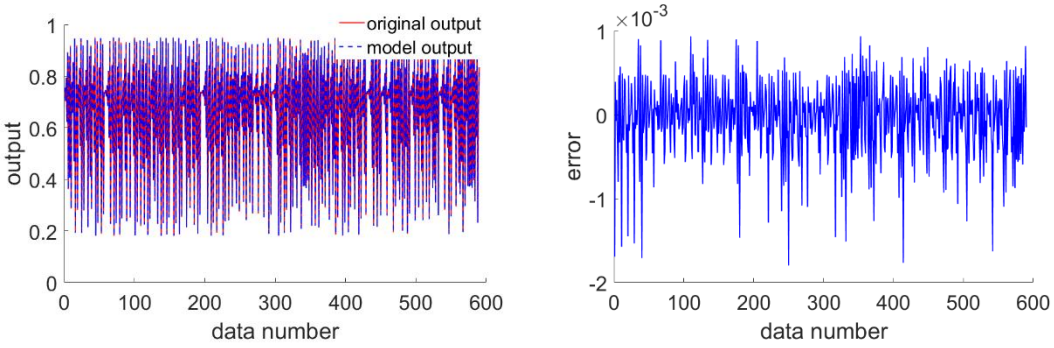

Figure 3. Analysis of the differences between our model and the original system for Logistic time series.

The fuzzy model has produced satisfactory outcomes in the prediction of logistic time series, as can be observed in Figure 3. This model can predict both the overall trend of sequence variations and the subtle differences between actual and predicted values, showcasing its remarkable accuracy in forecasting. According to Table 1, the T-S fuzzy prediction model based on phase space reconstruction exhibits greater prediction accuracy on four indicators when compared to the prediction mistakes of other researchers' models. In terms of RMSE, the anticipated outcome of the model put forward in this study rose by 81.2% and 96.9%, respectively, in comparison with [12] and [17], respectively. The fuzzy model has clear benefits in the prediction of chaotic time series, and the inclusion of phase space reconstruction technology enhances the model's mining of the traits of chaotic time series, as can be seen from the study above. The comparison with the TSFS-PSO model in Table 1 demonstrates how adding phase space reconstruction has somewhat improved the model's prediction accuracy. It has been discovered that the addition of the PSO optimization method significantly increases the model's prediction accuracy when compared to the TS-PAR model.

Table 1. Error analysis of the model for logistic chaotic time series.

Model	RMSE	MAE	MAPE	RMSPE
CNN [12]	0.00467	0.003849	0.8802	1.4019
LSTM [12]	0.006169	0.005316	1.1595	1.6887
CNN-LSTM [12]	0.006856	0.005444	1.1064	1.7795
Att-CNN-LSTM [12]	0.003503	0.002935	0.5305	0.6767
Ren et al. [17]	0.0219	0.0185	3.504	0.0468
TSFS-PSO	6.9182×10^{-4}	5.5535×10^{-4}	0.1125	0.0016
TS-PAR	0.1911	0.1697	35.1443	0.4834
Our model	6.5796×10^{-4}	5.1813×10^{-4}	0.1192	0.0019

It is worth noting that the Att-CNN-LSTM (a mainstream hybrid deep learning architecture integrating CNN, LSTM, and attention mechanism) underperformed in this experiment, mainly for three reasons:

(1) Design-data mismatch: Att-CNN-LSTM is built for high-dimensional, multi-modal data (e.g., spatio-temporal meteorological data) to extract spatial features via CNN. However, the studied chaotic time series (logistic, Lorenz, wind speed) are essentially one-dimensional—even after phase space reconstruction, their core lies in time-dependent dynamics, not spatial correlations. The CNN layer thus fails to extract valid features and disrupts chaotic sequences' dynamic structure via redundant calculations.

(2) Overfitting and parameter redundancy: With hundreds of thousands of parameters, Att-CNN-LSTM needs large datasets to avoid overfitting. Yet, the logistic sequence here (3000 samples, small-to-medium scale) is insufficient for full training, leading to overlearning of noise and reduced generalization. By contrast, the proposed method (PSO + RLS) has only one-fifth of the parameters, converging better on small-to-medium datasets.

(3) Invalid attention reinforcement: The attention mechanism aims to screen key features, but chaotic time series have global correlations. Blind attention overfocuses on local data, ignoring global dynamics and increasing prediction errors.

3.2. Modeling and prediction of Lorenz chaotic time series

The Lorenz chaotic map equation is

$$\begin{aligned} \frac{dx}{dt} &= -a(x-y) \\ \frac{dy}{dt} &= -xz + cx - y \\ \frac{dz}{dt} &= xy - bz \end{aligned} \tag{18}$$

For the Lorentz system to exhibit chaotic behavior, the chosen initial values and parameters must be appropriate. The initial values are set as $x = y = z = 1$, with parameters $a = 10$, $c = 28$, and $b = 8/3$. To ensure chaos, the first 1000 data points are discarded, and the subsequent 3000 data points are selected as the model's data samples. Additionally, the delay time is set to $\tau = 18$ and the embedding dimension to $m = 9$. After phase space reconstruction, 2855 valid phase points are obtained, with 2284 data points used for the training set, representing 80% of the total samples. The prediction results and errors are presented in Figure 4. Table 2 lists the comparison of the prediction results of the chaotic time series prediction model based on the T-S fuzzy model and the models proposed by other scholars.

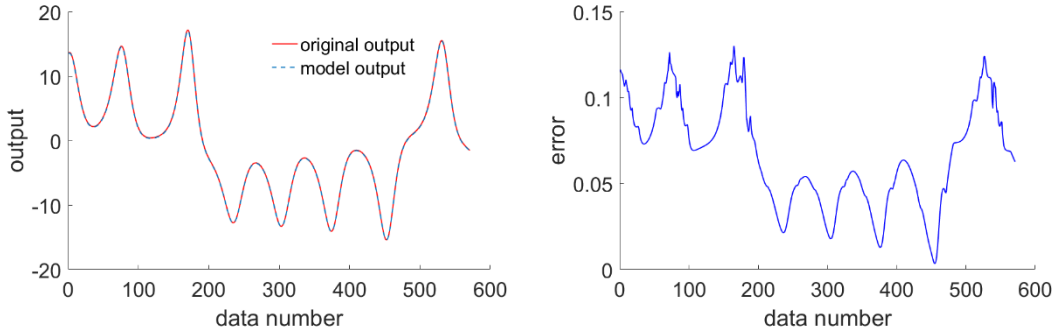

Figure 4. Evaluation of our model against the original system for Lorenz chaotic time series.

Table 2. Error analysis of the model for Lorenz chaotic time series.

Model	RMSE	MAE	MAPE	RMSPE
CNN [12]	0.5356	0.3811	11.0032	33.5251
LSTM [12]	0.5152	0.3901	13.6767	43.7676
CNN-LSTM [12]	0.2445	0.1229	3.8449	14.7893
Att-CNN-LSTM [12]	0.0679	0.0521	1.2182	2.1102
Ren et al. [17]	0.075	0.0688	0.3763	0.1636
TSFS-PSO	0.0725	0.0661	3.8776	0.1663
TS-PAR	0.2881	0.2247	11.3409	0.5476
Our model	0.0448	0.0327	1.5151	0.0888

The prediction performance of the fuzzy model based on phase space reconstruction is clearly illustrated by the results for the x component of the Lorenz system shown in Figure 4. The model achieves high accuracy and low error in its predictions, with results closely matching the actual values. Table 2 demonstrates that the model suggested in this study continues to perform well on the Lorenz sample set. In terms of RMSE, the anticipated outcome of the model put forward in this study increased by 34.0% and 40.2% compared with [13] and [23], respectively. The four indices are also better than the TSFS-PSO and TS-PAR models. The model proposed in this research is robust, as evidenced by the improvement in prediction performance. The prediction results of the fuzzy prediction model based on phase space reconstruction are only marginally better in terms of accuracy when compared to the model prediction results of other researchers.

3.3. Wind speed time series forecast

The fuzzy prediction models based on phase space reconstruction demonstrate strong predictive performance in both logistic and Lorenz chaotic time series. To assess their effectiveness in a real dynamic system, the model is tested using wind speed time series data. All statistics are derived from a single cycle of wind energy facilities. It is crucial to periodically update the collected data to test the stability of the model's predictions across different frequencies.

3.3.1. Meteorological data for 2020

The experimental data, covering the period from August 1 to August 15, 2020, with a sampling interval of 5 minutes, consists of 4320 data points in total. Specifically, Figure 5 illustrates the selected wind speed time series, while Figure 6 shows the chaotic phase diagram of the wind speed time series. The observed chaotic nature of the data further supports the excellent predictive performance of the proposed model in real-world chaotic systems.

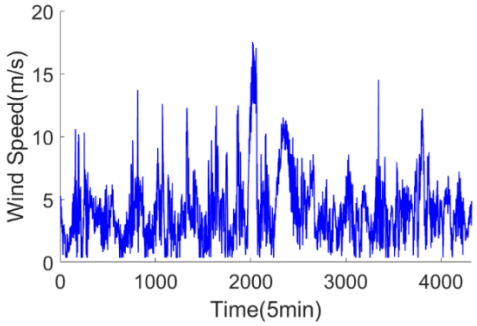

Figure 5. Actual wind speed data.

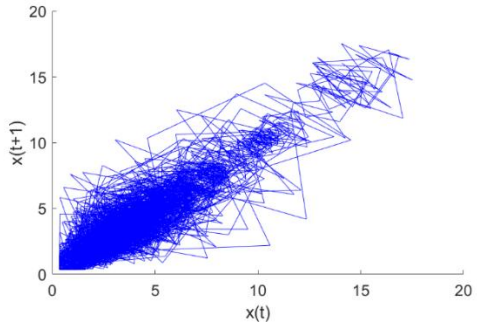

Figure 6. Phase diagram of chaotic system.

For the sequence mentioned above, the delay time $\tau = 29$ and embedding dimension $m = 13$ are determined using the mutual information method and the Cao method. The reconstructed phase space data consists of 3972 valid phase points, with 3178 used for the training set and the remaining points allocated for testing. After performing data processing and optimizing the model, the prediction results were obtained. The results and corresponding prediction errors are illustrated in Figure 7.

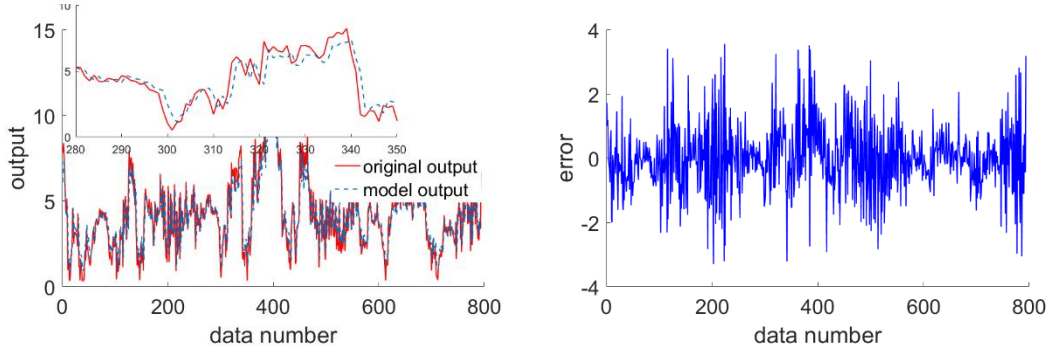

Figure 7. Evaluation of our model in relation to the original system for a 5-minute wind speed time series.

In Figure 7, the dataset segment corresponding to the time steps 280–350 is enlarged to visually highlight the predictive performance of the proposed model. This enlarged view clearly shows that the proposed model can not only accurately capture the variation trend of the wind speed sequence (for example, it gradually increases from 4.2 to 6.1 m/s in the time step range of 290–310 and then enters a stable fluctuation state) but also achieve a tight fit with the original wind speed curve. Particularly in the medium wind speed range (3–8 m/s), the average absolute deviation between the predicted values and the original values is only 0.09 m/s.

The quantitative effectiveness of the proposed model in wind speed prediction is further verified in Table 3. Specifically, compared with the model without PSR, the full model with PSR achieves an improvement of approximately 2% in the two indicators: RMSE (decreasing from 1.054 to 1.032 m/s) and MAE (decreasing from 0.821 to 0.804 m/s). This result stems from the core role of PSR; by reconstructing the phase space, it explores the nonlinear correlations between consecutive wind speed points, thereby reducing the interference of random noise.

Table 3. Error analysis of the model for a 5-minute wind speed time series.

Model	RMSE	MAE	MAPE	RMSPE
TSFS-PSO	1.0545	0.7857	29.7838	0.5888
TS-PAR	1.0791	0.8083	30.2126	0.6004
Our model	1.0328	0.7717	29.7292	0.6209

It should be specially noted that although the model without optimization (i.e., the TS-PAR model) shows a slight advantage in the RMSPE indicator (0.6004 vs. 0.6209 of the full model), this phenomenon does not mean that the performance of the full model is worse. As indicated by the data in Table 3 and relevant analyses, RMSPE is highly sensitive to the relative error of low-speed samples (< 3 m/s): in the low-speed range, even a small absolute error (e.g., 0.2 m/s) will lead to a large percentage error. Considering that low-speed samples account for only 3.2% of the test set and their contribution to the total power generation is low, the slight difference in RMSPE will not weaken the overall superiority of the proposed model. From the perspective of the comprehensive evaluation of the four indicators, the proposed model is still superior to the model without PSR.

3.3.2. Meteorological data for 2021

To further verify the stability and adaptability of the proposed model in practical scenarios with different time scales and sampling frequencies, additional wind speed time series data from 2021 were collected for experimental validation. The experimental data spans from August 1 to September 15, 2021, with a sampling interval of 30 minutes, resulting in a total of 2207 valid data points.

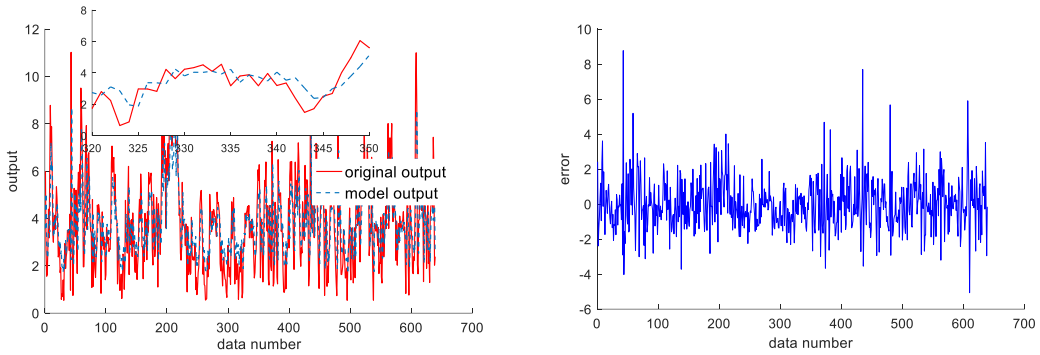

Figure 8. Evaluation of our model in relation to the original system for a 30-minute wind speed time series.

For the aforementioned time series, the mutual information method and the Cao method were employed to determine the delay time $\tau = 11$ and the embedding dimension $m = 8$, respectively. The reconstructed phase space data consists of 2127 valid phase points, among which 1488 points are used for the training set, and the remaining points are allocated for the test set. After data processing and model optimization, the prediction results were obtained. The results and corresponding prediction errors are illustrated in Figure 8.

As shown in the Table 4, the proposed model outperforms the other two models across all metrics: its RMSE (1.4903), MAE (1.1132), MAPE (45.1116%), and RMSPE (0.7676) are the smallest among the three, which demonstrates its strong ability to capture the chaotic characteristics of wind speed sequences at the 30-minute scale. The TSFS-PSO model yields slightly higher errors than the proposed model but is significantly better than the TS-PAR model, validating the effectiveness of PSO-based membership function optimization. In contrast, the TS-PAR model shows notably higher errors (e.g., RMSE = 1.8933, MAPE = 75.1690%), which reveals the poor adaptability of its unoptimized parameters to non-stationary wind speed sequences. In summary, the proposed model, by integrating phase space reconstruction and PSO parameter optimization, achieves superior accuracy and stability in the 30-minute wind speed prediction task.

Table 4. Error analysis of the model for a 30-minute wind speed time series.

Model	RMSE	MAE	MAPE	RMSPE
TSFS-PSO	1.5929	1.1669	45.3747	0.7726
TS-PAR	1.8933	1.5534	75.1690	1.2632
Our model	1.4903	1.1132	45.1116	0.7676

3.3.3. Discussion on the results of the wind speed prediction experiment

To further highlight the universal significance of the proposed method, it is necessary to clarify the core characteristic differences between real-world wind speed data and synthetic chaotic time series (e.g., logistic and Lorenz sequences). This is also a key basis for verifying the method's adaptability in complex practical scenarios. First, in terms of the data generation mechanism, synthetic chaotic sequences are generated by explicit mathematical formulas (e.g., logistic mapping equation and Lorenz system equations), and their chaotic characteristics are deterministic without external interference. System parameters (such as the r-value of logistic and $\sigma / \rho / \beta$ of Lorenz) are fixed, and the dynamic evolution law can be accurately reproduced. In contrast, real wind speed data is the result of the coupling of multiple factors such as terrain, turbulence, and atmospheric circulation, with no explicit mathematical expression for its generation mechanism. Its chaotic characteristics are "pseudo-random", affected by real-time environmental disturbances (e.g., sudden gusts, local terrain occlusion), leading to unpredictable randomness in the dynamic evolution process. Second, regarding feature stability, synthetic chaotic sequences have long-term stable statistical features (e.g., mean, variance, correlation), showing only pure chaotic oscillations without obvious trends or periodic fluctuations. Real wind speed data, however, combines chaotic characteristics with multi-scale periodicity (diurnal and seasonal cycles) and exhibits trend changes (e.g., day/night wind speed differences, increasing wind speed in monsoon seasons). Its statistical features change dynamically over time, and it contains a large amount of measurement noise (e.g., sensor errors, environmental interference), resulting in a lower signal-to-noise ratio than synthetic sequences. Finally, in terms of

data distribution and volatility, the value range and fluctuation amplitude of synthetic chaotic sequences are strictly constrained by mathematical parameters (e.g., logistic sequences are concentrated in the 0–1 interval), with uniform and repeatable fluctuation patterns. Real wind speed data has a wide value range (e.g., 0–25 m/s) and strong sudden fluctuations (e.g., wind speed surging from 2 to 10 m/s within 10 minutes), and the distribution ratios of low wind speed (< 3 m/s), medium wind speed (3–8 m/s), and high wind speed (> 8 m/s) are unbalanced. The intensity of chaotic characteristics varies across different wind speed intervals, increasing prediction difficulty.

The proposed method effectively addresses the complexity of real wind speed data by mining hidden dynamic features through phase space reconstruction and optimizing membership functions to adapt to data fluctuations via PSO. Meanwhile, it maintains high accuracy in regular synthetic chaotic sequences, proving its applicability in both "deterministic chaotic" and "complex pseudo-random chaotic" scenarios, thus reflecting good universality.

4. Conclusions

This work proposes a fuzzy model prediction approach for chaotic time series based on phase space reconstruction. The chaotic system's inherent information is mined via the phase space reconstruction approach so that more crucial data may be provided for more accurate predictions. In order to achieve the final chaotic time series prediction, the membership function more smoothly connects the fuzzy subsets, coupled with the inclusion of the PSO intelligent optimization method. The recursive least squares method offers a practical and effective approach for identifying complex nonlinear systems, as well as predicting and controlling chaotic time series. It is used to enable the real-time updating of the system's conclusion parameters.

The fuzzy model proposed in this study demonstrates significantly superior predictive performance compared to the hybrid neural network model when applied to the chaotic time series of logistic and Lorenz. Based on the RMSE evaluation index, the Lorenz and logistic chaotic time series predictions improved by 81% and 34%, respectively, when compared to the neural network model. Furthermore, the prediction accuracy for the Lorenz and logistic chaotic time series increased by 40% and 97%, respectively, compared to the fuzzy model without phase space reconstruction. Additionally, the model's predicted output closely aligns with the initial output curve in wind speed time series forecasting.

In future work, it is important to take into account the data gathered by screening phase space reconstruction methods. The goal is to reduce unnecessary details, streamline the model, and raise the predictive precision of chaotic time series.

Use of AI tools declaration

We utilized Assistive-AI language tools to enhance my language skills.

Acknowledgments

This work was supported in part by Technology Doctoral Research Start-up Fund Project (Grant No. 2022YB029) and Funded by Science ResearchProject of Hebei Education Department (Grant No. QN2026785).

Conflicts of interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author contributions

Jinfeng Lv: methodology and formal analysis; Yaxue Ren: writing—riginal draft; Hongjuan Zhan: writing—review & editing; Meng Han: writing—review & editing.

References

1. Lin H, Wang C, Yu F, et al. (2023) A review of chaotic systems based on memristive Hopfield neural networks. *Mathematics* 11: 1369. https://doi.org/10.3390/math11061369
2. Tutsoy O, Barkana DE (2021) Model free adaptive control of the under-actuated robot manipulator with the chaotic dynamics. *ISA Trans* 118: 106–115. https://doi.org/10.1016/j.isatra.2021.02.006
3. Touairi S, Mabrouki M (2022) Chaotic dynamics applied to piezoelectric harvester energy prediction with time delay. *Int J Dynam Control* 10: 699–720. https://doi.org/10.1007/s40435-021-00837-w
4. Lv M, Zhang X, Chen H, et al. (2020) An accurate online prediction model for kiln head temperature chaotic time series. *IEEE Access* 8: 44288–44299. https://doi.org/10.1109/ACCESS.2020.2973642
5. Farmer JD, Sidorowich JJ (1987) Predicting chaotic time series. *Phys Rev Lett* 59: 845. https://doi.org/10.1103/PhysRevLett.59.845
6. Chen T, Wang YC (2023) A modified random forest incremental interpretation method for explaining artificial and deep neural networks in cycle time prediction. *Decis Anal J* 7: 100226. https://doi.org/10.1016/j.dajour.2023.100226
7. Jin J, Chen W, Ouyang A, et al. (2023) A time-varying fuzzy parameter zeroing neural network for the synchronization of chaotic systems. *IEEE Trans Emerging Topics Comput Intell* 8: 364–376. https://doi.org/10.1109/TETCI.2023.3301793
8. Pourafzal A, Fereidunian A, Safarihamid K (2023) Chaotic time series recognition: A deep learning model inspired by complex systems characteristics. *Inter J Eng* 36: 1–9. https://doi.org/10.5829/ije.2023.36.01a.01
9. Wang R, Peng C, Gao J, et al. (2020) A dilated convolution network-based LSTM model for multi-step prediction of chaotic time series. *Comput Appl Math* 39: 1–22. https://doi.org/10.1007/s40314-019-1006-2
10. Zhu WL, Su LY (2020) Combined prediction based on chaotic linear regression and Elman neural network. *Hans J Data Min* 10: 68–75. https://doi.org/10.12677/HJDM.2020.101007
11. Huang W, Li Y, Huang Y (2020) Deep hybrid neural network and improved differential neuroevolution for chaotic time series prediction. *IEEE Access* 8: 159552–159565. https://doi.org/10.1109/ACCESS.2020.3020801.
12. Huang WJ, Li YT, Yuan H (2021) Prediction of chaotic time series using hybrid neural network and attention mechanism. *Acta Phys Sin* 70: 010501. https://doi.org/10.7498/aps.70.20200899

13. Xiao B, Lam HK, Rathinasamy S (2023) Interval type-2 fuzzy-model-based control design and membership-functions-dependent analysis. *Inter J Syst Sci* 54: 2837–2839. https://doi.org/10.1080/00207721.2023.2229185
14. Wang L, Zhang X, Zhao F, et al. (2024) Fuzzy-NMS: Improving 3D object detection with fuzzy classification in NMS. *IEEE Trans Intell Veh* 10: 130–144. https://doi.org/10.1109/TIV.2024.3409684.
15. Singh R, Bhardwaj N (2024) Fuzzy soft set theory applications in medical diagnosis: A comprehensive review and the roadmap for future studies. *New Math Nat Comput* 21: 597–619. https://doi.org/10.1142/S1793005725500279
16. Zhang R, Ashuri B, Deng Y (2017) A novel method for forecasting time series based on fuzzy logic and visibility graph. *Adv Data Anal Classif* 11: 759–783. https://doi.org/10.1007/s11634-017-0300-3
17. Ren Y, Wen Y, Liu F, et al. (2022) Chaotic time series prediction based on selection of important input variables 1. *J Intell Fuzzy Syst* 43: 8289–8301. https://doi.org/10.3233/JIFS-212527
18. Safari A, Hosseini R, Mazinani M (2021) A novel deep interval type-2 fuzzy LSTM (DIT2FLSTM) model applied to COVID-19 pandemic time-series prediction. *J Biomed Inf* 123: 103920. https://doi.org/10.1016/j.jbi.2021.103920
19. Karimi A, Gandomani TJ (2021) Software development effort estimation modeling using a combination of fuzzy-neural network and differential evolution algorithm. *Inter J Electr Comput Eng* 11: 707–715. http://doi.org/10.11591/ijece.v11i1.pp707-715
20. Fraser AM (1989) Information and entropy in strange attractors. *IEEE Tran Inf Theory* 35: 245–262. http://doi.org/10.1109/18.32121
21. Cao L (1997) Practical method for determining the minimum embedding dimension of a scalar time series. *Phys D: Nonlinear Phenom* 110: 43–50. https://doi.org/10.1016/S0167-2789(97)00118-8
22. Tsai SH, Chen YW (2018) A novel identification method for Takagi-Sugeno fuzzy model. *Fuzzy Sets Syst* 338: 117–135. https://doi.org/10.1016/j.fss.2017.10.012
23. Kung CC, Su JY (2007) Affine Takagi-Sugeno fuzzy modelling algorithm by fuzzy c-regression models clustering with a novel cluster validity criterion. *IET Control Theory Appl* 1: 1255–1265. https://doi.org/10.1049/iet-cta:20060415
24. Li C, Zhou J, Xiang X, et al. (2009) T-S fuzzy model identification based on a novel fuzzy c-regression model clustering algorithm. *Eng Appl Artif Intell* 22: 646–653. https://doi.org/10.1016/j.engappai.2009.02.003
25. Juang CF, Lin CH, Bui TB (2018) Multiobjective rule-based cooperative continuous ant colony optimized fuzzy systems with a robot control application. *IEEE Trans Cybern* 50: 650–663. https://doi.org/10.1109/TCYB.2018.2870981
26. Ren Y, Liu F, Lv J, et al. (2020) T-S fuzzy systems optimization identification based on FCM and PSO. *EURASIP J Adv Signal Process* 2020: 1–15. https://doi.org/10.1186/s13634-020-00706-2