Nonlinear Dyn (2023) 111:21837–21859
https://doi.org/10.1007/s11071-023-08977-8

ORIGINAL PAPER

Time series classification of dynamical systems using deterministic learning

Chen Sun · Weiming Wu · Cong Wang

Received: 4 October 2022 / Accepted: 10 September 2023 / Published online: 6 November 2023

Abstract This paper studies the classification of large-scale time series data constructed by nonlinear dynamical systems via deterministic learning. Firstly, a large-scale time series dataset including five classes of dynamical patterns is constructed based on the benchmark Lorenz system. Secondly, the performance of the dynamical pattern recognition method based on deterministic learning is evaluated for the time series classification task in the large-scale time series dataset. Last but not least, based on the concept of representative selection and the recent research on the deterministic learning theory, a novel dynamical pattern recognition-based time series classification framework is modified, including: (1) the selection of a representative training subset with the largest Lyapunov exponent as an example; (2) inherent dynamics modeling via deterministic learning in the representative subset; (3) dynamics comparing between representative training patterns and a given test pattern through generating recognition errors; (4) and label assigning based on the minimal recognition error principle. The significance of this paper is that it improves the application of existing dynamical pattern recognition methods to large-scale datasets by incorporating representative selection strategies. The proposed method can achieve comparable classification accuracy on a small representative subset of chaotic trajectories, as compared to the original large-scale training set. Numerical simulations are provided to demonstrate the effectiveness of the proposed method. The simulation results show that the proposed method outperforms even the baseline deep learning methods on the representative subset in the dynamical system classification tasks.

Keywords Deterministic learning · Time series classification · Dynamical systems · Representative selection

C. Sun · W. Wu (✉) · C. Wang (✉)
Center for Intelligent Medical Engineering, School of Control Science and Engineering, Shandong University, Jinan 250061, China
e-mail: auwuweiming@163.com

C. Wang
e-mail: wangcong@sdu.edu.cn

1 Introduction

Recently, the increasing use of temporal data, in particular time series data, has initiated various research and development attempts in the field of physics, medicine, or meteorology. Unlike other data types, such as images, text, and videos, time series exhibit complex characteristics including temporal ordering, nonstationarity, nonlinearity, chaos, bifurcation, etc. These inherent characteristics make the classification task on time series challenging.

From physics to meteorology, as well as many engineered systems, the observed time series can genuinely be described as (usually nonlinear) dynamical systems, whose temporal evolution is represented by a set of differential or time-recursive equations. The problem of dynamical systems classification, including the classification of different types of systems and the classification of different states of a single system, has always been a focus of researchers. Zhou et al. [1] investigated the classification of chaos in 3-D autonomous quadratic dynamical systems. Based on the Shilnikov criteria, they classified chaos into four types according to whether or not homoclinic and heteroclinic orbits exist. Dong et al. [2] developed a methodology for the topological classification of periodic orbits in the generalized Lorenz-type system. They first located the periodic orbits using the variational method and then classified them by building appropriate symbolic dynamics. These traditional nonlinear dynamics methods usually require available dynamic equations and rigorous theoretical derivations. Nonetheless, in the case of more challenging real-world scenarios, precise dynamic equations are frequently unknown, thereby constraining the applicability of traditional techniques.

In recent years, using machine learning and deep learning methods to study dynamical systems from a data-driven perspective has also attracted the attention of some researchers. Boullé et al. [3] used five different neural networks, including multi-layer perceptron (MLP), fully convolutional network (FCN), large kernel convolutional neural network (LKCNN), residual network (ResNet), etc., to classify time series generated by four different dynamical systems into chaotic or non-chaotic. They assessed the generalization ability of neural networks from low-dimensional phase space to high-dimensional phase space and found that LKCNN exhibited the highest classification accuracy. Hassona et al. [4] investigated the reconstruction of 2D bifurcation diagrams of a resistor–inductor–capacitor (RLC) circuit system based on machine learning. They utilized K-nearest neighbors with dynamic time warping (KNN-DTW), time series forests (TSFs) with entropy, fully convolutional network with long short-term memory (LSTM-FCN), etc., to classify time series as chaotic or periodic, and conducted that the TSFs with entropy show the best classification performance. Uzun [5] studied the classification of time series belonging to three famous dynamical systems: Lorenz, Chen, and Rossler systems. They compared 18 machine learning methods, such as Naive Bayes (NB), support vector machines (SVM), and KNN, and obtained a result that KNN with the highest accuracy. Aricioğlu et al. [6] also studied the classification of two dynamical systems, Chen and Rossler. They converted time series into images and then used SqueezeNet, Visual Geometry Group 19 (VGG-19), and six other depth neural networks based on transfer learning technology. The focus of the above works is to compare and evaluate the classification performance of each single-model, but there is no in-depth discussion on how to deal with the disordered and nonlinear characteristics of chaotic time series. Hybrid models are considered to perform better than singe-models [7]. Karasu and Altan [8] proposed a chaotic Henry gas solubility optimization (CHGSO) based LSTM model to forecast chaotic crude oil time series. They used technical indicators such as trend, volatility, and momentum to capture nonlinear characteristics. Karasu et al. [9] also proposed a hybrid model based on radial basis function-based support vector regression (RBFSVR), which employed indicators such as simple moving average (SMA), exponential moving average (EMA), and a wrapper-based feature selection approach to handle the non-stationary and nonlinear properties of crude oil time series.

The machine learning-based methods usually need to design a large number of features manually and thus have little potential to scale to large datasets by its limitation of time complexity. While deep learning approaches can produce high accuracy without heavy data preprocessing or feature engineering, this end-to-end learning also reduces the interpretability of the neural networks. As a result, it is difficult to apply these deep learning approaches in some fields where high security is required, such as medical diagnosis. Moreover, deep learning approaches rely on large training datasets, which do not perform well for smaller sample sizes. In addition, most of the known literatures on time series classification of dynamical systems seldom consider the inherent dynamics of characteristics of time series, or only quantify it by calculating statistical indicators based on the univariate time series. There is a lack of research on the inherent dynamics relationship among multiple variables of dynamical systems.

For the past few years, a method named deterministic learning has been proposed and gradually developed in the field of machine learning in dynamic environments [10–12]. From the aspect of system identification, this method addresses the dynamics modeling problem from time series data generated by nonlinear dynamical systems. Furthermore, based on the learned knowledge of inherent dynamics within time series via deterministic learning, a new method for rapid dynamical pattern recognition was proposed, in which the similarity between dynamical patterns is measured in a dynamical manner. Thanks to the extraction of inherent dynamics within time series data, the deterministic learning-based rapid dynamical pattern recognition method has shown good interpretability and successfully applied in many fields such as early diagnosis of myocardial ischemia and detection of small oscillation faults [13, 14]. Recently, the rapid dynamical pattern recognition method has been further developed. For example, in Ref. [15], this method was extended to the sampled data to solve the problem of rapid recognition for dynamical patterns composed of sampling sequences. In Ref. [16], a new similarity definition based on structural stability was added to improve the recognition accuracy of dynamical patterns. However, compared with these existing results, there is still less progress in time series classification problems, especially for large-scale time series data.

This study is motivated by solving the problems in the above discussion, i.e., 1) most conventional machine learning and deep learning-based time series classification techniques rarely consider the dynamics information of time series; 2) the demand for extending the existing dynamical pattern recognition method to large-scale time series classification. To this end, this paper further studies the time series classification problem in a large-scale[1] dataset constructed by multivariate dynamical systems via deterministic learning. Firstly, a large-scale time series dataset is constructed based on a benchmark Lorenz system. By varying the bifurcation parameter of this system, a total of 10,000 3-D time series are generated. According to the different periodic windows, where the bifurcation parameter is located, the generated time series (including period, period doubling, and chaos) are labeled into five classes, denoted by the symbols A^2B, A^2B^2, AB, A^3B^3, and A^2BAB^2AB, respectively. The symbols represent the order and the number of repetitions that the trajectory (time series) motion around the fixed points A and B in phase space, which is usually used to describe the different dynamical behaviors. Then, the performance of the dynamical pattern recognition method based on deterministic learning in the time series classification task is evaluated in the large-scale time series dataset. This method achieves competitive accuracy compared to deep learning baseline models such as MLP, FCN, and ResNet. The numerical simulation results show that the dynamics features extracted by deterministic learning have significant discriminability, and the dynamics difference between time series can be used as an effective criterion for classification.

Finally and most importantly, a further investigation to improve the performance of the above method proceeds. The above method needs to model the dynamics of all training instances and calculate the recognition errors from a given test instance to the entire dynamical pattern library when making decisions. Therefore, the storage and computation costs would be very expensive when the dataset becomes very large. Representative selection aims to find a small subset of representatives from a training set that can appropriately represent it [17–19]. For this, we further study a time series classification method based on representative selection and deterministic learning. Specifically, the proposed method adds a representative selection strategy from the perspective of nonlinear dynamics. Before the modeling phase, the largest Lyapunov exponents (LLE) for all dynamical patterns in the training set are calculated, and the chaotic samples are selected to construct a representative subset according to the LLE value. Based on the new representative subset, the modeling and classification are performed using deterministic learning and dynamical pattern recognition. The outline of the proposed method is shown in Fig. 1. The simulation results show that the proposed method can effectively reduce the number of training samples and the runtime required while ensuring the generalization ability of the existing method as mentioned above. When making classification decisions, the representative subset is easier to interpret than the entire dataset. In addition, under the same conditions of using

[1] Large scale in this study refers to a dataset with a size of ten thousand.

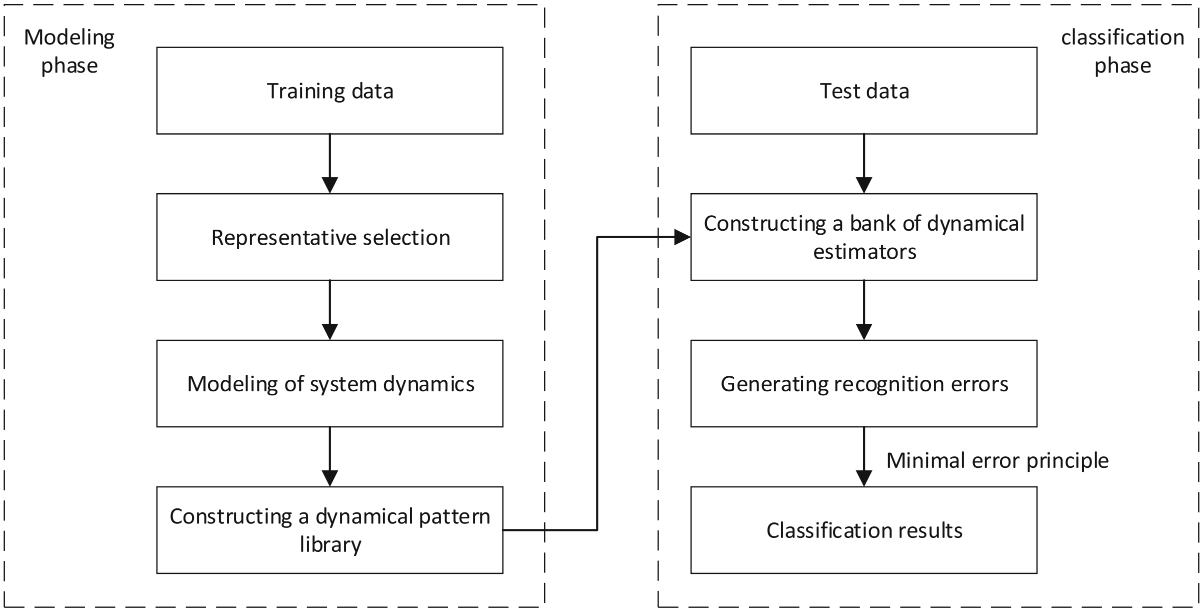

Modeling phase
Training data
Representative selection
Modeling of system dynamics
Constructing a dynamical pattern library
classification phase
Test data
Constructing a bank of dynamical estimators
Generating recognition errors
Minimal error principle
Classification results

Fig. 1 Overall work flow of the proposed method

representative subsets, the proposed method achieves higher classification accuracy than the deep learning baseline models, indicating that the proposed method also has a strong learning ability with small sample sizes.

Compared with the existing dynamical pattern recognition method, the proposed method incorporates a representative selection strategy to achieve more efficient learning of dynamics knowledge. The main contributions of this paper are summarized as follows:

(1) A large-scale time series dataset composed of various dynamical patterns is constructed, providing a foundation for studying the classification problem of dynamical systems.

(2) Extending the dynamical pattern recognition method to large-scale time series classification tasks and verifying the dynamics features extracted by deterministic learning have significant discriminability in classification.

(3) A representative selection strategy is proposed from the perspective of nonlinear dynamics. Only a few chaotic trajectories are selected for dynamics modeling, which can ensure the classification performance of the existing method and enhance its applicability on large-scale datasets.

The rest of this paper is organized as follows. Section 2 proposes the method of time series classification of dynamical systems based on representative selection and deterministic learning. Section 3 introduces the Lorenz system and the dataset generated from it. Section 4 analyzes and discusses the results of the numerical simulation. The conclusions are drawn in Sect. 5.

2 Method

The symbols used in this section and their definitions are shown in Table 1.

2.1 Problem formulation

Consider a class of nonlinear dynamical systems:

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x};\boldsymbol{p}_s), \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{1}$$

where $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T \in \mathbb{R}^{\boldsymbol{n}}$ denotes the state vector of this system, $\boldsymbol{F}(\boldsymbol{x};\boldsymbol{p}_s) = [f_1(\boldsymbol{x};\boldsymbol{p}_s), f_2(\boldsymbol{x};\boldsymbol{p}_s).,\cdots,f_n(\boldsymbol{x};\boldsymbol{p}_s)]^T$ denotes the system dynamics, where each component $f_i(\boldsymbol{x};\boldsymbol{p}_s)$ is a continuous nonlinear function but unknown, $\boldsymbol{p}_s$ is the parameter vector of this system, which can be linear or nonlinear, and $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$ denotes the initial sate.

Table 1 List of basic symbols

Symbols	Definition
$\mathbb{R}^n$, $\mathbb{N}$	n-dimensional real vector space, natural vector space
$\mathcal{P}$, $\mathcal{P}_{rep}$	time series training set, representative subset
ϕ^s, ϕ^r	training time series, test time series
y^s, K, M	class labels, number of categories, number of training time series
$\boldsymbol{W}^*$	ideal weights of the RBF neural network
$\boldsymbol{S}$	regressor vector of the RBF neural network
$\widehat{\boldsymbol{W}}$	estimation of the RBF neural network weights vector
$\overline{\boldsymbol{W}}$	constant RBF neural network weights vector
m	number of neurons of RBF network
T	sampling period
$\boldsymbol{x}$	state vector of the system
$\widehat{\boldsymbol{x}}$	state vector of the dynamical RBF network identifiers
$\overline{\boldsymbol{x}}$	state vector of the dynamical estimators
$\boldsymbol{x}^T$, $\Vert\boldsymbol{x}\Vert$, $\Vert\boldsymbol{x}\Vert_{A1}$	the transpose, Euclidean norm, and average L_1 norm of $\boldsymbol{x}$

Consider the time series data sampled from the above system, denoted as $\phi^s_{T,N}(\boldsymbol{x}_0) = [\boldsymbol{x}(t_0), \boldsymbol{x}(t_0+T), \cdots, \boldsymbol{x}(t_0+(N-1)T)]$, where T is the sampling period and N is the total steps. This time series is abbreviated as ϕ^s, which can be represented by the following Euler model:

$$\boldsymbol{x}[k+1] = \boldsymbol{x}[k] + T\boldsymbol{F}(\boldsymbol{x}[k];\boldsymbol{p_s}) \tag{2}$$

where $\boldsymbol{x}[k] = [x_1[k], x_2[k], \cdots, x_n[k]]^T \in \mathbb{R}^n$ denotes the state vector of the Euler model.

For the time series classification problem, a method based on representative selection and deterministic learning will be presented, which consists of a modeling phase and a classification phase.

(1) Modeling phase: consider a training set $\mathcal{P} = \{(\phi^1, y^1), (\phi^2, y^2), \cdots, (\phi^M, y^M)\}$, where M is the number of time series, and y^s is the class label corresponding to the time series ϕ^s with $y^s \in [1, K]$, where K is the number of classes.[2] The goal of the modeling phase is to learn the inherent dynamics $\boldsymbol{F}(\boldsymbol{x};\boldsymbol{p^s})$ of the time series ϕ^s in the training set $\mathcal{P}$ (or the representative subset $\mathcal{P}_{rep}$).

(2) Classification phase: consider a test time series ϕ^r, which is generated by sampling the following dynamical system:

$$\dot{\boldsymbol{x}} = \boldsymbol{F}(\boldsymbol{x};\boldsymbol{p_r}), \boldsymbol{x}(t_0) = \boldsymbol{x}_0 \tag{3}$$

The goal of the classification phase is to find the most similar training dynamical pattern to the test time series ϕ^r from the training set $\mathcal{P}$ (or the representative set $\mathcal{P}_{rep}$) according to the dynamic differences, and then to assign the class label of the most similar training pattern to the test time series.

2.2 Modeling phase

Consider each training time series $\phi^s, s \in \{1, \cdots, M\}$ in the training set $\mathcal{P}$ or the representative subset $\mathcal{P}_{rep}$ (for details on the representative selection method, see Sect. 2.4). Recall that the representation by Euler model:

$$x_i[k+1] = x_i[k] + Tf_i(\boldsymbol{x}[k];\boldsymbol{p_s}), i \in \{1, \cdots, n\} \tag{4}$$

where $\boldsymbol{x}[k] = [x_1[k], x_2[k], \cdots, x_n[k]]^T \in \mathbb{R}^n$ denotes the state vector of the Euler model, T is the sampling period, which is known.

The objective of the modeling phase is to accurately approximate the system dynamics $f_i(\boldsymbol{x}[k];\boldsymbol{p_s})$ via deterministic learning method. In the first step, a kind of Radial Basis Function (RBF) neural network is constructed in the following form:

$$f_{nn}(\boldsymbol{Z}) = \sum_{j=1}^{m} w_j s_j(\boldsymbol{Z}) = \boldsymbol{W}^T\boldsymbol{S}(\boldsymbol{Z}) \tag{5}$$

[2] K was pre-labeled in the training set $\mathcal{P}$, and in the simulation of this study $K = 5$.

where $\boldsymbol{Z}$ is the input vector, $\boldsymbol{W}$ is the RBF neural network weight vector, $\boldsymbol{S}(\mathbf{Z}) = [s_1(\|\mathbf{Z} - \boldsymbol{\mu}_1\|), s_2(\|\mathbf{Z} - \boldsymbol{\mu}_2\|), \cdots, s_m(\|\mathbf{Z} - \boldsymbol{\mu}_m\|)]^T$ is the regressor vector of the RBF neural network, in which $s_j\left(\|\mathbf{Z} - \boldsymbol{\mu}_j\|\right) = \exp\left[\frac{-(\mathbf{Z}-\boldsymbol{\mu}_j)^T(\mathbf{Z}-\boldsymbol{\mu}_j)}{\eta^2}\right]$ denotes a Gaussian function, $\boldsymbol{\mu}_j(j = 1, 2, \cdots, m)$ is neuron centers, m is the number of neurons, and η is the width of the receptive field. Both m and η are constants.

The RBF neural network has been shown to approximate continuous (or discrete) functions with a desired accuracy [20]. In addition, it has been proved that when the neuron centers are placed on a regular lattice, the regressor vector $\boldsymbol{S}(\boldsymbol{Z})$ of RBF neural network can meet the persistency of excitation (PE) condition along recurrent trajectory $\boldsymbol{Z}$ [10, 12, 21]. The satisfaction of this condition can ensure the accurate identification of nonlinear system dynamics. So, the RBF neural network is employed in the deterministic learning, and the following dynamical RBF network identifiers [22] are used to model the dynamical function $f_i(\boldsymbol{x}; \boldsymbol{p_s})$:

$$\begin{aligned} \hat{x}_i[k+1] &= x_i[k] + \alpha_i(\hat{x}_i[k] - x_i[k]) \\ &\quad + T\hat{\boldsymbol{W}}_i^T[k]\boldsymbol{S}(\boldsymbol{x}[k]) \ , \ i \in \{1, \cdots, n\} \end{aligned} \tag{6}$$

where $\hat{\boldsymbol{x}}[k] = [\hat{x}_1[k], \hat{x}_2[k], \ldots, \hat{x}_n[k]]^T$ denotes the state vector of the dynamical RBF network identifier, $\boldsymbol{x}[k]$ is the input vector of the identifier, $0 < \alpha_i < 1$ is a constant, representing the gain of the identifier, and $\hat{\boldsymbol{W}}_i^T[k]\boldsymbol{S}(\boldsymbol{x}[k])$ denotes the embedded RBF neural network for approximating the unknown dynamics $f_i(\boldsymbol{x}; p_s)$.

The weights of the RBF neural network are updated by the Lyapunov-based learning law:

$$\begin{aligned} \widehat{\boldsymbol{W}}_i[k+1] = \widehat{\boldsymbol{W}}_i[k] - T\gamma\boldsymbol{S}(\boldsymbol{x}[k])e_i[k+1], i \\ \in \{1, \cdots, n\} \end{aligned} \tag{7}$$

where $\gamma > 0$ is a constant, denoting the learning rate, and $e_i[k] = \widehat{x}_i[k] - x_i[k]$ is state tracking error.

By using the dynamical RBF neural network identifiers and weight update law described above, the learning problem of system dynamics can be converted into the following stability problem for LTV systems:

$$\begin{aligned} \begin{bmatrix} e_i[k+1] \\ \widetilde{\boldsymbol{W}}_i[k+1] \end{bmatrix} &= \begin{bmatrix} \alpha_i & T\boldsymbol{S}^T[k] \\ -\alpha_i T\gamma\boldsymbol{S}[k] & I - \gamma T^2\boldsymbol{S}[k]\boldsymbol{S}[k]^T \end{bmatrix} \\ &\quad \begin{bmatrix} e_i[k] \\ \widetilde{\boldsymbol{W}}_i[k] \end{bmatrix} + \begin{bmatrix} -\epsilon_i[k] \\ T^2\gamma\boldsymbol{S}[k]\epsilon_i[k] \end{bmatrix} \end{aligned} \tag{8}$$

where $\widetilde{\boldsymbol{W}}_i[k] = \widehat{\boldsymbol{W}}_i[k] - \boldsymbol{W}_i^*$ denotes the weight estimation error of the RBF neural network and $\boldsymbol{W}_i^*$ is the ideal weight.

According to Theorem 1 in Ref. [22], the exponential stability of the above learning error system (8) can be guaranteed with the satisfaction of the PE condition of the regression vector $\boldsymbol{S}(x)$. It means the parameter estimation error $\widetilde{\boldsymbol{W}}$ will exponentially converge to 0, i.e., the weights $\widehat{\boldsymbol{W}}$ converge to a neighborhood around their ideal values $\widehat{\boldsymbol{W}}^*$. Now, a locally accurate modeling of the unknown inherent dynamics $f_i(\boldsymbol{x}; \boldsymbol{p_s})$ can be achieved with this RBF neural network:

$$f_i(\boldsymbol{x}; \boldsymbol{p_s}) = \overline{\boldsymbol{W}}_i^T\boldsymbol{S}(\boldsymbol{x}) + \epsilon_i(\boldsymbol{x}), i \in \{1, \cdots, n\} \tag{9}$$

where $\epsilon_i(\boldsymbol{x})$ denotes the actual modeling error and $\overline{\boldsymbol{W}}_i$ denotes the constant RBF neural network weights, obtained by the following equation:

$$\overline{\boldsymbol{W}}_i = \frac{1}{k_b - k_a + 1}\sum_{k=k_a}^{k_b}\widehat{\boldsymbol{W}}_i[k], i \in \{1, \cdots, n\} \tag{10}$$

where $[k_a, k_b]$ denotes a time interval after the transient process, in which k_a and k_b are constants.

Based on the above deterministic learning method, the locally accurate identification/modeling for inherent dynamics within the training sample ϕ^s is achieved and the learned knowledge $\overline{\boldsymbol{W}}_i^T$ is stored to form a dynamical pattern library. This will be further applied in the subsequent classification phase. Figure 2 summarizes the main procedure of the modeling phase.

2.3 Classification phase

Considering all RBF neural network weights $\overline{\boldsymbol{W}}_i^s, s \in \{1, \cdots, M\}$ obtained in the previous phase, a bank of M dynamical estimators is constructed:

$$\begin{aligned} \overline{x}_i^s[k+1] = u_i[k] + b_i\left(\overline{x}_i^s[k] - u_i[k]\right) + T\overline{\boldsymbol{W}}_i^{sT}\boldsymbol{S}(\boldsymbol{u}[k]), i \\ \in \{1, \cdots, n\}, s \in \{1, \cdots, M\} \end{aligned} \tag{11}$$

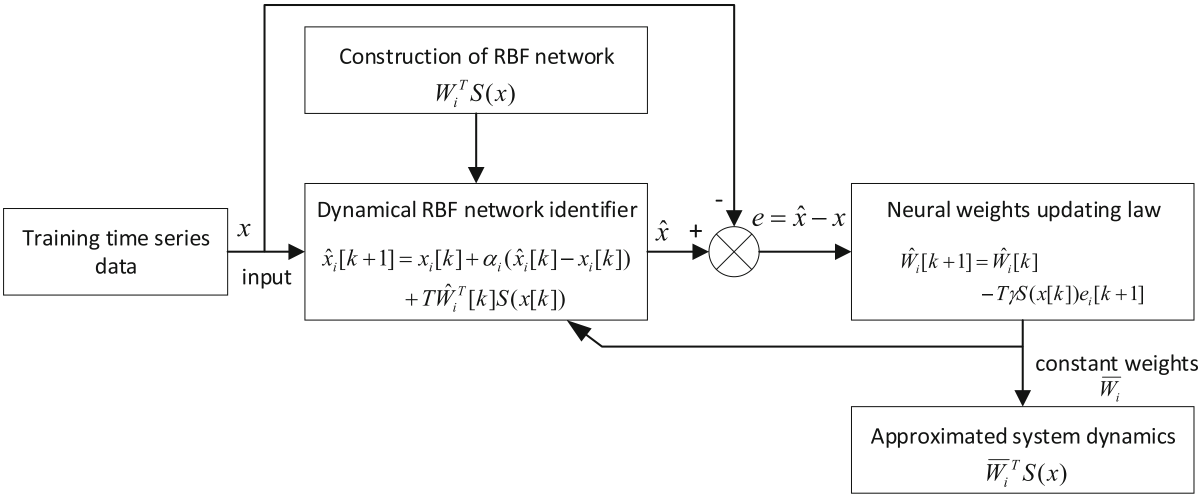

Construction of RBF network
$W_i^T S(x)$
Training time series data
x
input
Dynamical RBF network identifier
$\hat{x}_i[k+1]=x_i[k]+\alpha_i(\hat{x}_i[k]-x_i[k])$
$+T\hat{W}_i^T[k]S(x[k])$
$\hat{x}$
+
-
$e=\hat{x}-x$
Neural weights updating law
$\hat{W}_i[k+1]=\hat{W}_i[k]$
$-T\gamma S(x[k])e_i[k+1]$
constant weights
$\overline{W}_i$
Approximated system dynamics
$\overline{W}_i^T S(x)$

Fig. 2 Flowchart of the modeling phase

where $s=1,\cdots,M$ represents the sth dynamical estimator, $\bar{\boldsymbol{x}}^s[k]=\left[\bar{x}_1^s[k],\bar{x}_2^s[k],\cdots,\bar{x}_n^s[k]\right]^T$ is the state of the dynamical estimators, $\boldsymbol{u}[k]=[u_1[k],u_2[k],\cdots,u_n[k]]^T$ denotes the input of the dynamical estimators, and $0<b_i<1$ is a constant, denoting the gain of the dynamical estimators.

Consider the test sample ϕ^r generated by the sampling of the dynamic system (3), which is expressed by the Euler model as follows:

$$x_i[k+1]=x_i[k]+Tf_i(\boldsymbol{x}[k];\boldsymbol{p_r}), i\in\{1,\cdots,n\} \tag{12}$$

where $\boldsymbol{x}[k]=[x_1[k],x_2[k],\cdots,x_n[k]]^T$ is the state of the Euler model.

By inputting the test sample into the bank of dynamical estimators (11) in parallel, i.e., $u_i[k]=x_i[k], i\in\{1,\cdots,n\}$, the recognition errors corresponding to the sth training sample can be generated:

$$\tilde{x}_i^s[k]=\bar{x}_i^s[k]-x_i[k], i\in\{1,\cdots,n\}, s\in\{1,\cdots,M\} \tag{13}$$

These recognition errors can be formulated by the following discrete-time dynamical system:

$$\tilde{x}_i^s[k+1]=b_i\tilde{x}_i^s[k]+T\left(\overline{\boldsymbol{W}}_i^{sT}\boldsymbol{S}(\boldsymbol{x}[k])-f_i(\boldsymbol{x}[k];\boldsymbol{p_r})\right), i \in\{1,\cdots,n\}, s\in\{1,\cdots,M\} \tag{14}$$

According to the similarity definition of dynamical pattern in reference [15], the similarity between the training sample and the test sample can be determined by the dynamics difference ($\left|\overline{\boldsymbol{W}}_i^{sT}\boldsymbol{S}(\boldsymbol{x}[k])-f_i(\boldsymbol{x}[k];\boldsymbol{p_r})\right|, \forall s\in\{1,\cdots,M\}$), and a small dynamic difference will lead to a small recognition error. In practice, the similarity is quantified by computing the average L_1 norm of the recognition error:

$$\|\tilde{x}_i^s[k]\|_{A1}=\frac{1}{K_p}\sum_{j=k-K_p+1}^{k}\left|\tilde{x}_i^s[j]\right|, i\in\{1,\cdots,n\}, s \in\{1,\cdots,M\} \tag{15}$$

where $K_p\in\mathbb{N}$ is a constant, denoting the range of the average L_1 norm.

Under the above classification mechanism, the test sample can be classified by searching the class label of the training sample corresponding to the minimum average L_1 norm. Figure 3 summarizes the main procedure of the classification phase.

Remark 1 In the dynamical pattern recognition mechanism, the dynamics of the training set need to be modeled first, and the modeling results are stored to form a dynamical pattern library. Then the time series to be classified is compared with the whole library to calculate the recognition errors by constructing a bank of dynamical estimators. So, it will face storage and computing challenges on large-scale datasets. In the next subsection, a representative selection strategy will be proposed to solve this problem.

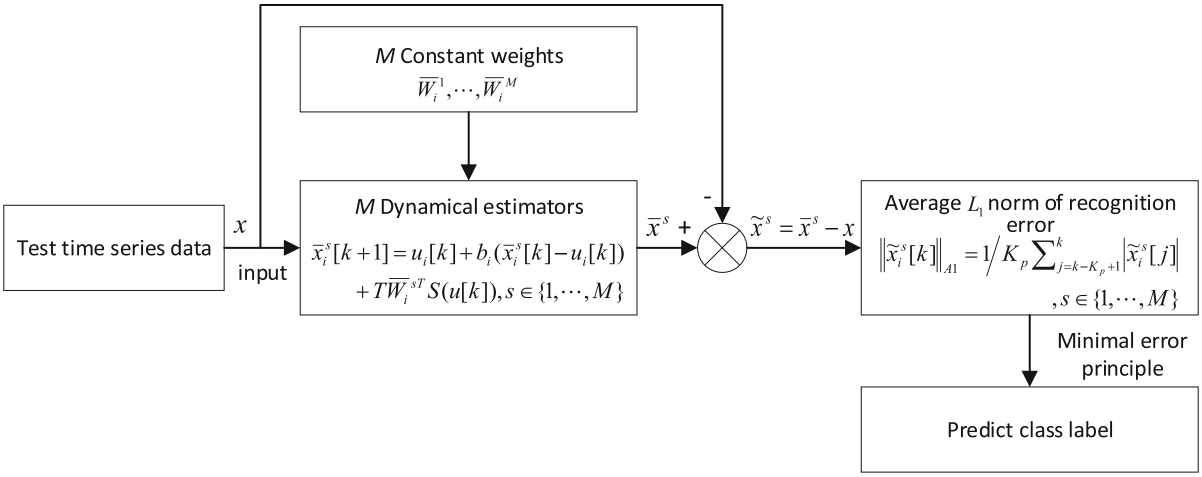

M Constant weights
$\overline{W}_i^1,\cdots,\overline{W}_i^M$
Test time series data
x
input
M Dynamical estimators
$\bar{x}_i^s[k+1]=u_i[k]+b_i(\bar{x}_i^s[k]-u_i[k])$
$+T\overline{W}_i^{sT}S(u[k]),s\in\{1,\cdots,M\}$
$\bar{x}^s$ +
-
$\tilde{x}^s=\bar{x}^s-x$
Average L_1 norm of recognition error
$\|\tilde{x}_i^s[k]\|_{A1}=1/K_p\sum_{j=k-K_p+1}^{k}|\tilde{x}_i^s[j]|$
$,s\in\{1,\cdots,M\}$
Minimal error principle
Predict class label

Fig. 3 Flowchart of the classification phase

2.4 Representative selection

Given a training set $\mathcal{P}$ consisting of time series ϕ^s, the purpose is to obtain a representative subset $\mathcal{P}_{rep}$ from $\mathcal{P}$ ($\mathcal{P}_{rep}\subseteq\mathcal{P}$) by designing a selection rule. Herein, we propose a representative selection strategy based on the Largest Lyapunov Exponent (LLE) as an example.

The LLE is an important quantitative metric to measure the dynamical properties of a dynamical system, which represents the average exponential rate of convergence or divergence between adjacent trajectories in the phase space. The LLE can be calculated as follows [23]:

$$\lambda=\frac{1}{t_{N\prime}-t_0}\sum_{k=1}^{N\prime}\ln\frac{L(t_k)}{L(t_{k-1})} \tag{16}$$

where $L(t_k)$ denotes the distance between two adjacent trajectories at time t_k, and $N\prime$ is the total number of steps. Negative LLE means that the trajectory of the system will converge to a fixed point; zero LLE means that the trajectory will converge to a limit cycle (periodic trajectory), and positive LLE means that the trajectory will behave as chaos (chaotic trajectory).

Chaos is a random-like motion exhibited by deterministic systems, which is neither periodic nor convergent. Chaos has ergodicity. That is, it can visit all its possible states [24]. Therefore, it can be inferred that chaotic trajectories contain more dynamics information than periodic trajectories. Selecting chaotic trajectories to construct a representative subset will more comprehensively characterize the dynamics of the original dataset.

The specific selection process of the representative subset $\mathcal{P}_{rep}$ is twofold: 1) calculation of the LLE λ^s of the time series φ^s in the training set $\mathcal{P}$ in turn; 2) and the selection of the chaotic time series whose LLE is larger than 0, i.e.,

$$\mathcal{P}_{rep}=\{\varphi^s\epsilon\mathcal{P}|\lambda^s>0\},s\in\{1,\cdots,M\} \tag{17}$$

To this end, $\mathcal{P}_{rep}$ can be employed to represent the original training set $\mathcal{P}$. Further numerical validation is shown in Sect. 4.3.

Remark 2 Chaos is a common phenomenon that exists in most nonlinear systems. Note that the proposed representative selection strategy, which selects chaotic trajectories as representative pattern via the LLE, is mainly for nonlinear dynamic systems that can generate chaotic trajectories.

Remark 3 Compared to the existing dynamical pattern recognition methods, the new development of the proposed method is incorporating the representative selection strategy and the deterministic learning theory. This involves selecting only chaotic trajectories as representative patterns for modeling from the perspective of nonlinear dynamics. This approach effectively reduces the number of training samples and algorithm runtime and improves the application of the existing method on large-scale datasets.

3 Dynamical systems and dataset

3.1 Lorenz system

The benchmark Lorenz system is a classical three-dimensional chaotic system derived from a mathematical model for atmospheric convection [25]. The dynamical equations of this system are given in (18), where σ, ρ and β are the system parameters and x_1, x_2 and x_3 are the system state variables.

$$\begin{cases} \dot{x}_1 = \sigma(x_2 - x_1) \\ \dot{x}_2 = x_1(\rho - x_3) - x_2 \\ \dot{x}_3 = x_1x_2 - \beta x_3 \end{cases} \tag{18}$$

The Lorenz system has rich dynamical behaviors and can produce periodic, quasi-periodic, and chaotic solutions under different parameter conditions. Many scholars have fixed the parameters $\sigma = 10, \beta = 8/3$ and investigated the effect of the parameter $\rho \in [0, \infty]$. Sparrow [26] collated studies from previous literature and gave three famous periodic windows, i.e., $C_1 = [\,99.524 \quad 100.795\,]$, $C_2 = [\,145 \quad 166\,]$, and $C_3 = [\,214.364 \quad \infty\,]$. Take periodic window C_1 as an example. In this window, the trajectories approach a stable periodic orbit that circles the first fixed point twice and then the second fixed point once, which we denote A^2B, as seen in Fig. 4a. As ρ decreases, there is a cascade of period-doubling bifurcations. The period doubles to $\left(A^2B\right)^2$, $\left(A^2B\right)^4$, and so on (Fig. 4b and c, respectively). As ρ continues to approach the lower boundary of this window, the trajectories eventually reach full chaos, as seen in Fig. 4d. In the other two periodic windows C_2 and C_3, these trajectories undergo period-doubling bifurcations as ρ decreases like that of window C_1. The trajectories in C_2 and C_3 are shown in Fig. 4e-h and i-l, which are described by A^2B^2 and AB, respectively.

From the above, it is known that for different values of ρ, the dynamical behaviors of the system are very different but for some ranges of ρ, the dynamical behaviors are similar. In addition to the three periodic windows mentioned above, we found two much more complicated period orbits A^3B^3 and A^2BAB^2AB by numerical computation, as seen in Fig. 4(m)-(p) and Fig. 4(q)-(t), respectively. The corresponding periodic window is $C_4 = [\,91.9 \quad 93.24\,]$ and $C_5 = [\,131 \quad 133\,]$ (approximately). The window size we have determined may not be one hundred percent accurate but will exhibit the dynamical behavior very well.

3.2 Creating the dataset

This section explains how to generate the time series datasets based on the Lorenz system. When creating the dataset, the ODE45 function in Matlab was used to numerically solve the Lorenz system. Taking the parameters $\sigma = 10$, $\beta = 8/3$ and varying ρ in five periodic windows yielded diversified time series data. The time series generated by the same periodic window were regarded as the same class. 2000 different values of parameter ρ were selected for each periodic window. Thus, the dataset contains a total of 10,000 time series with 3-dimensional in 5 classes. In the actual calculation process, the length of the generated time series was variable, and the range of the periodic windows would be enlarged appropriately to involve more dynamical patterns. In this way, it is ensured that the classification task is a difficult enough problem. The used calculation and system parameters' values are shown in Table 2.

Notice that each class in the dataset contains both periodic and chaotic time series. The total dataset was split into the training set (60%), validation set (5%) and the test set (35%). In order to ensure data distribution was balanced in each class, the hierarchical random sampling method was adopted. Details of the distribution of the dataset are given in Table 3.

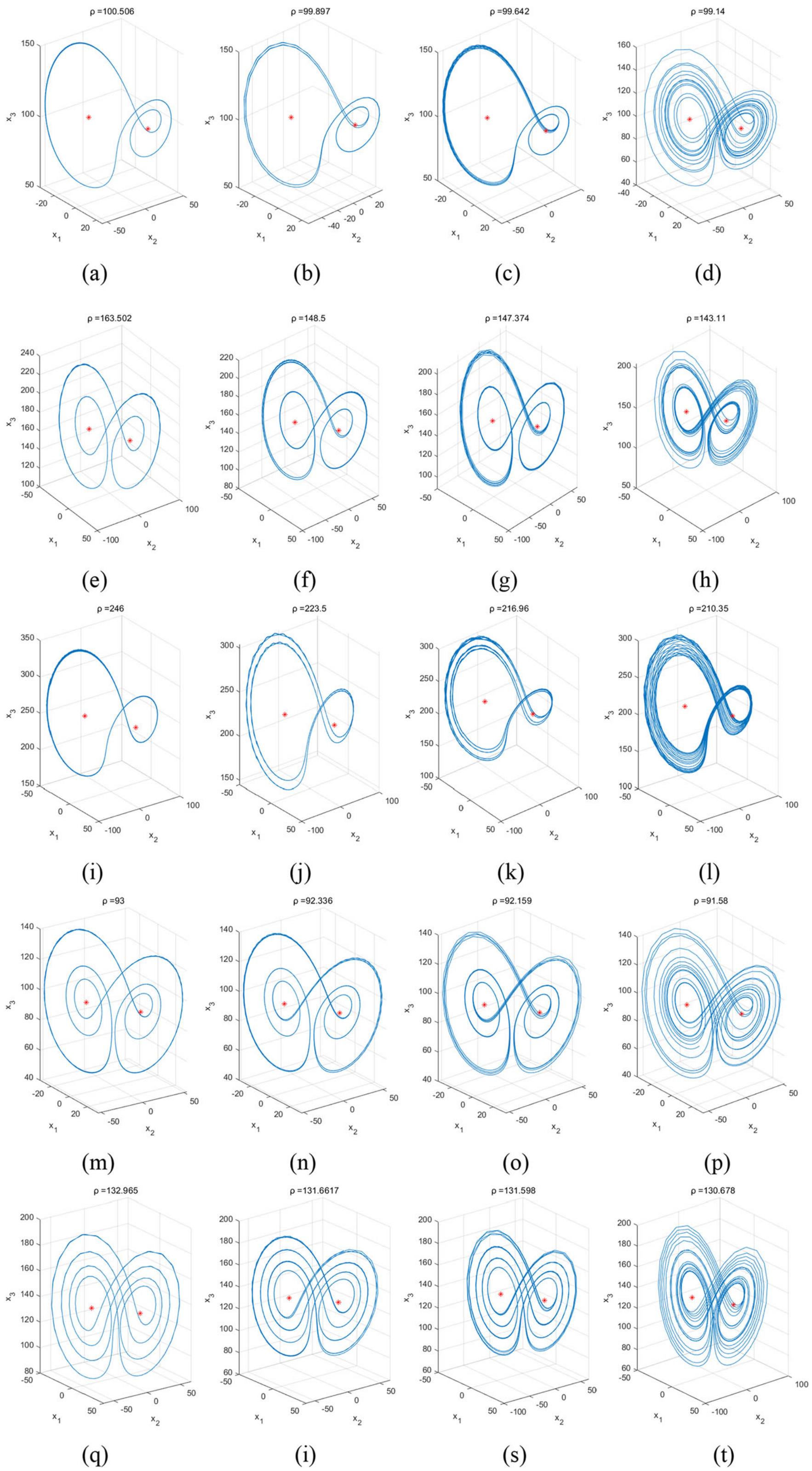
4 Numerical results and Discussion

4.1 Classification performance

In this section, we evaluated the classification performance of the dynamical pattern recognition method based on deterministic learning on the Lorenz dataset. Performance was evaluated in terms of accuracy, precision, recall, and F1-score. The formulas for calculating these metrics are as follows:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FN} + \text{FP}} \tag{19}$$

Fig. 4 The phase portrait of the Lorenz system. (a) A^2B period orbit for $\rho = 100.506$; (b) $(A^2B)^2$ period orbit for $\rho = 99.897$; (c) $(A^2B)^4$ period orbit for $\rho = 99.642$; (d) C_1 chaotic orbit for ρ=99.14; (e) A^2B^2 period orbit for $\rho = 163.502$; (f) $(A^2B^2)^2$ period orbit for $\rho = 148.5$; (g) $(A^2B^2)^4$ period orbit for $\rho = 147.374$; (h) C_2 chaotic orbit for ρ=143.11; (i) AB period orbit for $\rho = 246$; (j) $(AB)^2$ period orbit for $\rho = 223.5$; (k) $(AB)^4$ period orbit for $\rho = 216.96$; (l) C_3 chaotic orbit for ρ=210.35; (m) A^3B^3 period orbit for $\rho = 93$; (n) $(A^3B^3)^2$ period orbit for $\rho = 92.336$; (o) $(A^3B^3)^4$ period orbit for $\rho = 92.159$; (p) C_4 chaotic orbit for ρ=91.58; (q) A^2BAB^2AB period orbit for $\rho = 132.965$; (r) $(A^2BAB^2AB)^2$ period orbit for $\rho = 131.6617$; (s) $(A^2BAB^2AB)^4$ period orbit for $\rho = 131.598$; (t) C_5 chaotic orbit for ρ=130.678



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (i) (s) (t)

Table 2 Calculation and the system parameters used

The range of windows	System parameters	Initial conditions	The length of time series	The step size of RK4	The classes of time series
$C_1 = [98.885\ \ 100.885]$	$\sigma = 10, \beta = \frac{8}{3}$	$(x_0, y_0, z_0) = (1, 1, 1)$	750–3750	0.01	A^2B
$C_2 = [140\ \ 166]$	$\rho\epsilon(C_1 \cup C_2 \cup C_3$ $\cup C_4 \cup C_5)$				A^2B^2
$C_3 = [206\ \ 246]$					AB
$C_4 = [91.24\ \ 93.24]$					A^3B^3
$C_5 = [130.4\ \ 133]$					A^2BAB^2AB

Table 3 Distribution of the dataset

	C_1	C_2	C_3	C_4	C_5	Total
Training	1200	1200	1200	1200	1200	6000
Validation	100	100	100	100	100	500
Test	700	700	700	700	700	3500
Total	2000	2000	2000	2000	2000	**10,000**

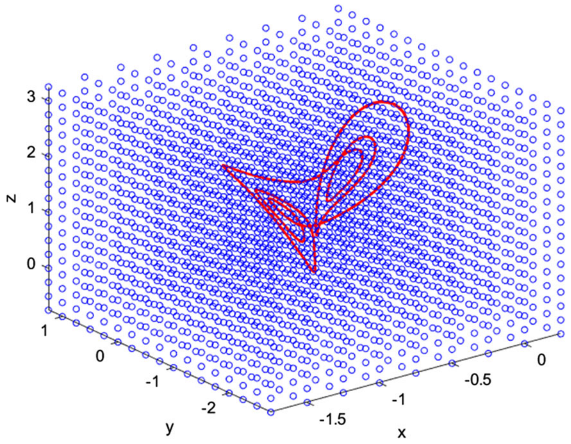


Fig. 5 The layout of neurons and the input time series in the phase space

$$\text{Precision} = \frac{TN}{TP + \text{FP}} \tag{20}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{21}$$

$$\text{F1} - \text{score} = \frac{1}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \tag{22}$$

where TP, TN, FP, and FN are the numbers of true positives, true negatives, false positives, and false negatives, respectively.

In the modeling phase, all time series in the training set were trained by deterministic learning. According to the above analysis in Sect. 3.1, we know that the second dynamical equation of the Lorenz system, i.e., $f_2(x_1, x_2, x_3) = \dot{x}_2 = x_1(\rho - x_3) - x_2$, contains a nonlinear term x_1x_3 and bifurcation parameter ρ, so we consider the $f_2(x_1, x_2, x_3)$ could well reflect the inherent dynamics of the Lorenz system. For this reason, we chose to model the $f_2(x_1, x_2, x_3)$ by employing the RBF network, which took $[x_1, x_2, x_3]^T$ as the input vector.

As a preprocessing step, each time series was z-normalized to have a mean equal to zero and a standard deviation equal to one. This is considered a common practice before classifying time series data [27]. In this way, all time series were normalized to the region $[-1.6, 0.2] \times [-2.5, 1] \times [-0.5, 3]$ in phase space. Thus, a unified RBF network structure can be constructed in a regular lattice with the neuron number $m = 2601$. The layout of the neurons in phase space is demonstrated in Fig. 5. The weights of the RBF network were updated based on Eq. (7) from the initial values $\boldsymbol{W}(0) = 0$ and the learning rate γ was set to be 200. After many iterations, the weights of some neurons close to the input trajectory were updated and converged to their optimal values $\boldsymbol{W}^*$. The process of weights convergence is shown in Fig. 6. Based on the deterministic learning, the dynamics of the Lorenz system can be approximated along the input trajectories, then these constant weights were stored for each training instance. Figure 7 shows the modeling results for some examples in the dataset.

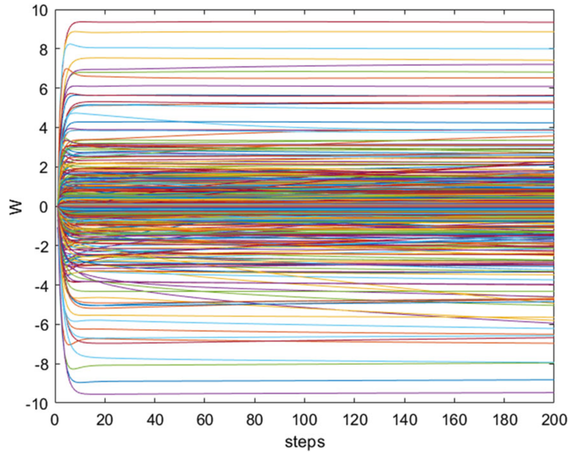


Fig. 6 The convergence of neural weights

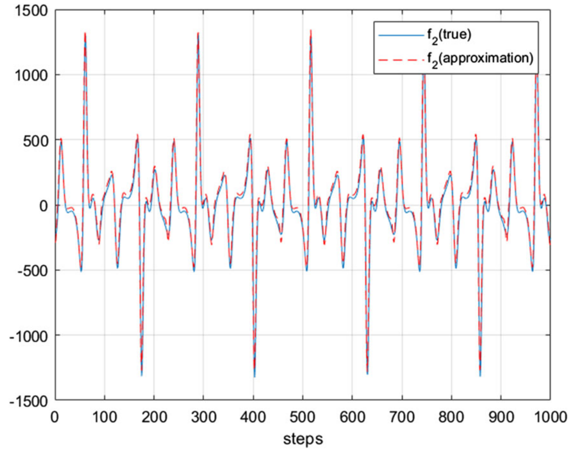


Fig. 7 Dynamical function modeling of Lorenz system by deterministic learning

In the classification phase, the constant weights were used to build dynamical estimators based on Eq. (11). The gain b of the dynamical estimators was set to be 0.5. Since 6000 instances were trained in the modeling phase, so a total of 6000 corresponding dynamical estimators were built in this phase. Then, each test instance was fed into the dynamical estimators to generate recognition errors. Thus, there were 6000 recognition errors for each test instance, which reflected the dynamic differences from a given test to all training instances. According to the decision principle based on the minimal error, the class label corresponding to the minimal average L_1 norm of the recognition error was assigned to the test instance. Figure 8 shows an example that is classified based on the minimal error principle.

Table 4 shows the confusion matrix of the dynamical pattern recognition method based on deterministic learning on the Lorenz dataset. From this matrix, it is easy to calculate the classification performance metrics, presented in Table 5. It can be seen that the classification accuracy of this method achieves 99.03%, which demonstrates the effectiveness of the deterministic learning applied in the time series classification of dynamical systems.

The deterministic learning method can model the inherent dynamics of time series, and the extracted dynamics features reflect the essential differences between time series, which has more significant discriminability. Therefore, using the recognition errors estimated based on dynamic differences as classification criteria is suitable for this time series classification task. On the other hand, it can be seen from Table 4 that the misclassified cases mainly focus on identifying C_2 as C_1, with a total of 19 instances. This may be because the C_2 and C_1 trajectories have similar dynamics in the left-wing (i.e., both of them circle the left fixed point twice in each period), and when the dynamics of the right-wing are not sufficiently distinct, a misclassification is likely to occur.

We also recorded the variation of runtime and classification accuracy with the size of the training set. We resampled the original training set (6000 instances) randomly to generate many training subsets with different sizes, and the test set (3500 instances) remained unchanged. Table 6 gives the runtime of the classification phase on different sizes of training sets, and Table 7 shows the corresponding test set classification accuracy. The computational environment is Windows 10 with Intel Xeon W-10885 M 2.40 GHz CPU and 64G RAM.

It can be seen from Table 6 that the runtime increases with the growth of the training set size. This is because the dynamical pattern recognition method based on deterministic learning first calculates the dynamic differences between the test sample and each training sample in the classification phase, and then realizes the classification of the test sample by searching the minimum error, so when the size of the training set increases significantly, the runtime required for classification increases as well.

From Table 7, we can see that the classification accuracy also increases with the growth of the training

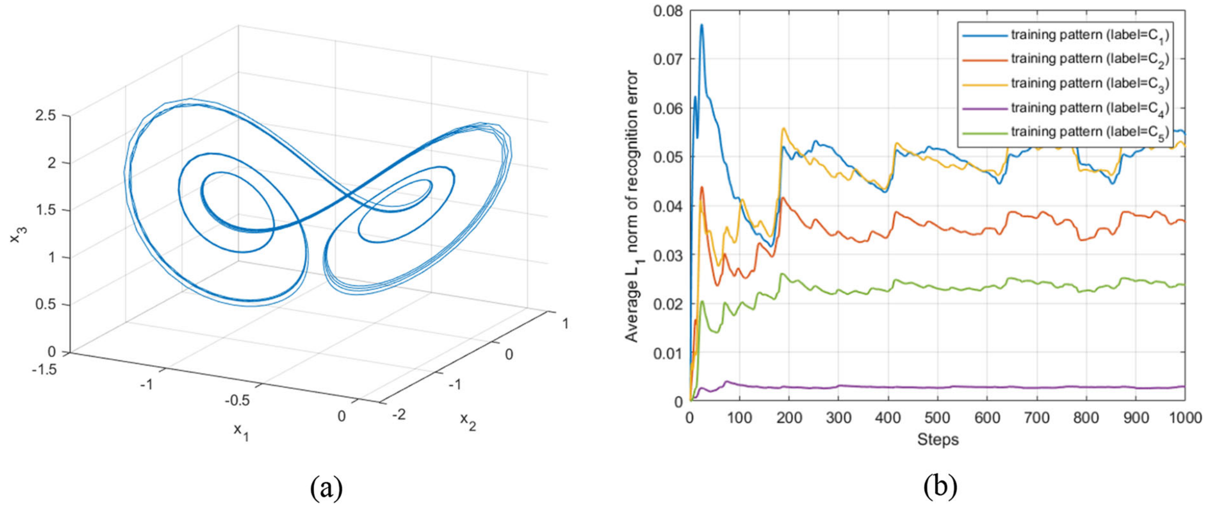


Fig. 8 Classification decision based on minimal average L_1 norm. (a) An example of test instance belongs to C_4. (b) Average L_1 norm of recognition error of this test instance against five training patterns, in which the error between test instance and training pattern (label = C_4) is minimal

Table 4 Confusion matrix of classification result of the dynamical pattern recognition method based on deterministic learning

		Predicted Class				
		C_1	C_2	C_3	C_4	C_5
True Class	C_1	**695**	0	0	5	0
	C_2	19	**681**	0	0	0
	C_3	2	0	**698**	0	0
	C_4	5	0	0	**695**	0
	C_5	0	3	0	0	**697**

Table 5 Performance metrics of the dynamical pattern recognition method based on deterministic learning

	Precision	Recall	F1-score
C_1	0.9639	0.9929	0.9782
C_2	0.9956	0.9729	0.9841
C_3	1.0	0.9971	0.9986
C_4	0.9929	0.9929	0.9929
C_5	1.0	0.9957	0.9979
Macro	0.9905	0.9903	0.9903
Overall Accuracy: 0.9903			

set size, but it no longer changes significantly when the training set size is up to a certain threshold. For example, when the training set size is $2^6(64)$,$2^{12}(4096)$, and 6000, the corresponding test set classification accuracy is 88.09%, 98.97%, and 99.03%, respectively. More training instances can build a richer dynamical pattern library, which helps to match more similar patterns in the classification phase and obtain higher classification accuracy. However, when the training instances have covered all the dynamical patterns, the accuracy will level off and no longer improve, and continuing to increase the training set size will only increase the runtime.

4.2 Comparison with baseline approaches

This section compared the performance of the dynamical pattern recognition method based on deterministic learning with other baseline classifiers on the Lorenz dataset. We consider three neural networks: multilayer perceptron (MLP), fully convolutional network (FCN), and residual network (ResNet). All three architectures were firstly proposed as baseline approaches for time series classification in [28]. These architectures are shown in Fig. 9.

(1) MLP: This is the most traditional form of deep neural network in the field of machine learning and essentially consists of several cascaded fully connected layers. The architecture adopted by Wang [28] contains three hidden layers and a softmax layer. Each hidden layer is composed of 500 neurons with ReLU activation function.

Table 6 Runtime of classification phase of the dynamical pattern recognition method based on deterministic learning on different sizes of training sets

Training number	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	6000
Runtime(h)	0.6035	0.6137	0.6445	0.7080	0.7599	0.9564	1.3630	1.7845

Table 7 Test set classification accuracy of the dynamical pattern recognition method based on deterministic learning on different sizes of training sets

Training number	2^6	2^7	2^8	2^9	2^{10}	2^{11}	2^{12}	6000
Accuracy	0.8809	0.8943	0.9140	0.9509	0.9600	0.9714	0.9897	0.9903

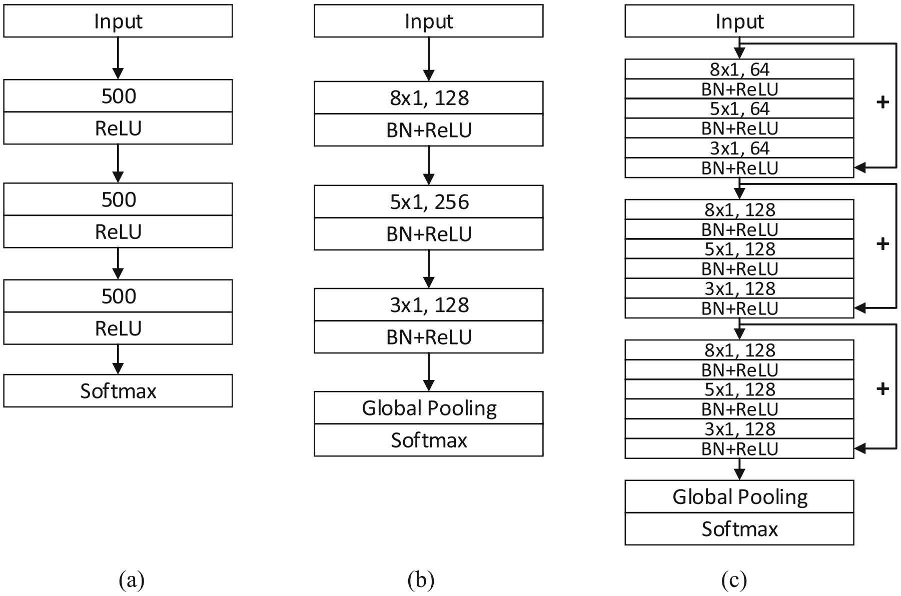


Fig. 9 The architectures of three baseline approaches. (a) MLP (b) FCN (c) ResNet

Moreover, a dropout operation is added before each layer to reduce overfitting. The number of neurons in the input layer depends directly on the length of the input time series.

(2) FCN: This is a kind of CNN without fully connected (FC) layers. As a replacement for FC, the global average pooling (GAP) layer is used to reduce drastically the number of parameters in the network. The architecture considered in [28] comprises three convolutional blocks, a GAP layer, and a softmax layer, each block consisting of a convolutional layer, a batch normalization layer, and a ReLU activation layer. The kernel size of the convolutional layer in each block is 8 × 1, 5 × 1, and 3 × 1, respectively, and the corresponding number is 128, 256, and 128. FCN is widely adopted for time series classification and is considered a competitive architecture.

(3) ResNet: This is an example of very deep convolutional neural networks. The main characteristic of ResNet is the addition of shortcut connections which enables a direct flow of the gradient. The architecture designed in [28] has 11 layers which consist of three residual blocks, a GAP layer, and a softmax layer. Each residual

Table 8 Optimization hyperparameters for the three baseline models

	Optimization algorithm	Loss	Max epochs	Batch	Initial learning rate
MLP	Adadelta	Entropy	1000	16	0.01
FCN	Adam	Entropy	1000	16	0.01
ResNet	Adam	Entropy	1000	32	0.01

block is composed of three convolutional blocks, which simply reuse the structures of the FCN, but the number of kernels is modified to 64, 128, and 128. ResNet is considered the most accurate deep neural network for time series classification tasks.

The Python implementations of the above three deep networks are available in [29]. Since the structure of some neural networks depends on the length of the input time series, the unequal length data in the Lorenz dataset needs to be processed in advance. We used the same approach as Fawaz et al. [30] and Keogh et al. [31] to linearly interpolate the time series of each dimension; thus, each time series will have a length equal to the length of the longest time series. This step is also helpful for parallel computation over the graphical processing units (GPUs). We also did z-normalize each time series to maintain the same preprocessing way as the deterministic learning. Table 8 shows the setting of optimization hyperparameters for the described models. Note that the learning rate was reduced by 0.5 times whenever the training loss did not improve for 30 epochs, and the training process would be early stopped whenever the validation loss did not improve for 60 epochs to avoid overfitting. After training, the model with the lowest loss value on the validation set was saved as the best model. In order to reduce the bias caused by the random weight initialization, we trained the three models with 10 different runs each. The mean accuracy over the 10 runs was calculated, as shown in Table 9.

According to Table 9, ResNet achieved the highest classification accuracy of 99.30%, providing further evidence to support the claim made by Fawaz et al. [30] that ResNet is currently the state-of-the-art method for time series classification. Fawaz et al. attribute the success of ResNet to its deep flexible architecture, suggesting that deeper neural networks are more powerful than shallower ones. FCN achieved 98.03% accuracy, but with a higher standard deviation of 0.022, suggesting that the performance of FCN is less stable, possibly due to the influence of initial weights. MLP performed the worst results with only 56.32% accuracy, which may be attributed to the use of the FC layer that processes time series elements independently, leading to the loss of temporal information. Comparing the dynamical pattern recognition method based on deterministic learning with these three deep models, we can see that the accuracy of the dynamical pattern recognition method based on deterministic learning (99.03%) outperforms MLP (56.32%) and FCN (98.03%), and is fairly close to ResNet (99.30%). The results show that the dynamical pattern recognition method is competitive with current state-of-the-art approaches in time series classification of dynamical systems.

Table 9 Accuracy comparison of different methods on Lorenz dataset

	Accuracy
MLP	0.5632(± 0.014)
FCN	0.9803(± 0.022)
ResNet	0.9930(± 0.006)
This study	0.9903(0)

4.3 Classification based on representative selection

We further explore the effective and efficient learning of the dynamical pattern recognition method based on deterministic learning on large-scale datasets. Considering that the essence of this method is to learn and utilize the inherent dynamics information of the data, we study the representative selection of the data from the perspective of nonlinear dynamics, expecting to

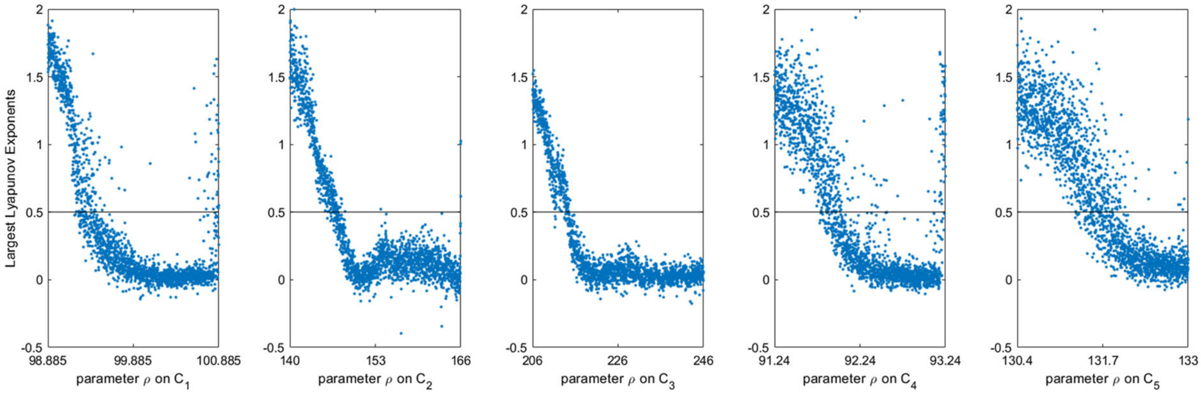


Fig. 10 The LLE of time series in five periodic windows

Table 10 Performance metrics of the dynamical pattern recognition method based on deterministic learning and representative selection

	Precision	Recall	F1-score
Class1	0.9626	0.9929	0.9775
Class2	0.9626	0.9729	0.9841
Class3	1.0	0.9971	0.9986
Class4	0.9929	0.9914	0.9921
Class5	1.0	0.9957	0.9979
Macro	0.9902	0.9900	0.9900
Overall Accuracy: 0.9900			

improve its practical application on large-scale datasets.

According to Sect. 2.4, firstly, the LLE of the time series in the original training set was calculated. Figure 10 shows the variation of the LLE value of time series in five periodic windows. We can see that time series in each window, the majority are non-chaotic, and a few are chaotic. Considering that the calculation value of the LLE has a certain error (the actual calculated LLE values of periodic trajectories were not strictly 0), it was decided to use a threshold value of 0.5 instead of 0 for representative selection. Therefore, we selected 1331 representative instances (chaotic instances) from 6000 training instances to form a representative subset.

Then, based on the new representative subset, the modeling and classification were performed using deterministic learning and dynamical pattern recognition. Relevant parameter settings were the same as those in Sect. 4.1. Table 10 shows the classification performance under the representative selection strategy.

From Table 10, the proposed dynamical pattern recognition method based on representative selection achieved 99% accuracy, which was almost equal to that of the existing dynamical pattern recognition method (99.03%). The results show that the representative selection strategy proposed in this paper is effective, that is, the chaotic subset selected based on the LLE can well represent the dynamics information of the original training set. By using the representative selection strategy, the accuracy achieved with only 1331 chaotic training instances is almost the same as that obtained with 6000 training instances, thereby greatly reducing storage and computational costs, and improving the applicability of the existing method to large-scale datasets.

Remark 4 From the perspective of nonlinear dynamics, chaotic trajectories are considered to contain richer dynamics information because of their ergodicity. In other words, the chaotic trajectories contain the dynamic characteristics of the periodic trajectories. And the essence of deterministic learning is to model the inherent dynamics of time series. Therefore, training on chaotic trajectories will be more efficient to learn the main dynamics of this class of trajectories, which is why choosing a small number of chaotic trajectories as representative patterns for modeling can still ensure high accuracy in the classification phase.

In addition, the classification performance of MLP, FCN and ResNet on the representative subset was evaluated. Figure 11 shows the accuracy of the

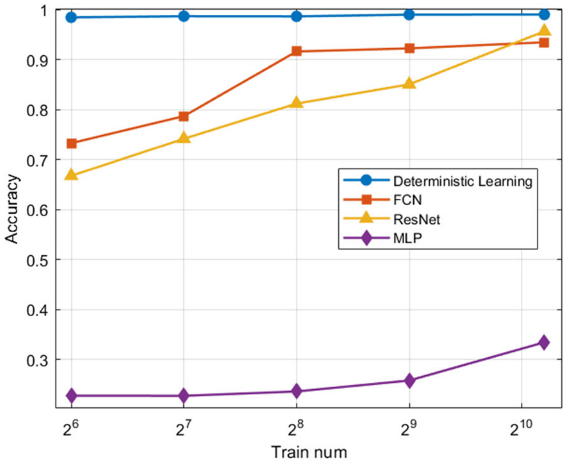


Fig. 11 Comparing the 4 methods' accuracy variation with respect to the number of representative instances

proposed method, MLP, FCN, and ResNet for varying numbers of representative instances. The accuracy of the three deep networks was approximately proportional to the size of the training sets. With a small sample size (2^6), all three networks failed to achieve effective learning. When the training subset took all chaotic samples (1331), the accuracy of MLP, FCN, and ResNet reached 33.40%, 93.44%, and 95.63%, respectively. Compared with the results obtained based on the original training set, MLP reached 59% of the original level, FCN reached 95% of the original level, and ResNet reached 96% of the original level. Although FCN and ResNet show good performance when the training subset takes all chaotic samples, this is not enough to show that the conclusion that chaotic samples are more representative of dynamics is still valid for deep networks.

The improved accuracy of these three networks is largely attributable to the increased amount of data, which allows the networks to automatically learn more discriminative features. Of course, due to the black-box nature of neural networks, we do not know what these features are (in this respect, the deterministic learning method has better interpretability, because the deterministic learning method extracts dynamics features, and realizes classification by comparing dynamics differences). Our findings agree with the deep learning literature that a large training set is needed to achieve high accuracy when training deep networks.

It is worth noting that the accuracy of the proposed method was 98.43% at the data size of 2^6, and stabilized at a very high accuracy level at all subsequent data sizes. This shows that this method has more advantages than the deep neural network in small sample learning. Besides, compared with Table 7, it can be seen that on the same number of training instances, the accuracy obtained by representative selection was higher than that by random selection. The results also verify the effectiveness of the proposed representative selection strategy.

Remark 5 The possible reasons why the proposed method works well even with a small number of samples are as follows: (1) the dynamics features extracted by deterministic learning with the representative selection have significant discriminability; (2) the classification principle based on minimal recognition error is like the exhaustive nearest neighbor search, which is competitive for small data samples.

Furthermore, the features extracted by deterministic learning method and the features extracted by deep learning methods (FCN and ResNet) were compared.

Deep learning-based features were obtained by extracting the weights of the intermediate layer of the trained neural network. We chose the latent representation after the Global Average Pooling (GAP) layer (the one before the Softmax classifier), as the feature vector extracted by deep learning.

Deterministic learning-based features were obtained by calculating the average L_1 norm of the recognition errors generated from dynamical estimators. To be consistent with the feature extraction of deep learning, a vector of the average L_1 norm of recognition errors in training sets was taken as the feature vector extracted by deterministic learning. To this end, both of them consider the pre-decisional information as the features for a fair comparison.

The multi-dimensional scaling (MDS) technique [30] is employed to map the feature vectors into a two-dimensional space for visualization. The visualization results on the test set of FCN features, ResNet features, and deterministic learning features trained on different representative subsets are shown in Fig. 12–14. Different colors represent different classes.

In Fig. 12(a), we can easily observe that at the representative subset size of 2^6, the features learned by FCN are linearly inseparable when projected into two-

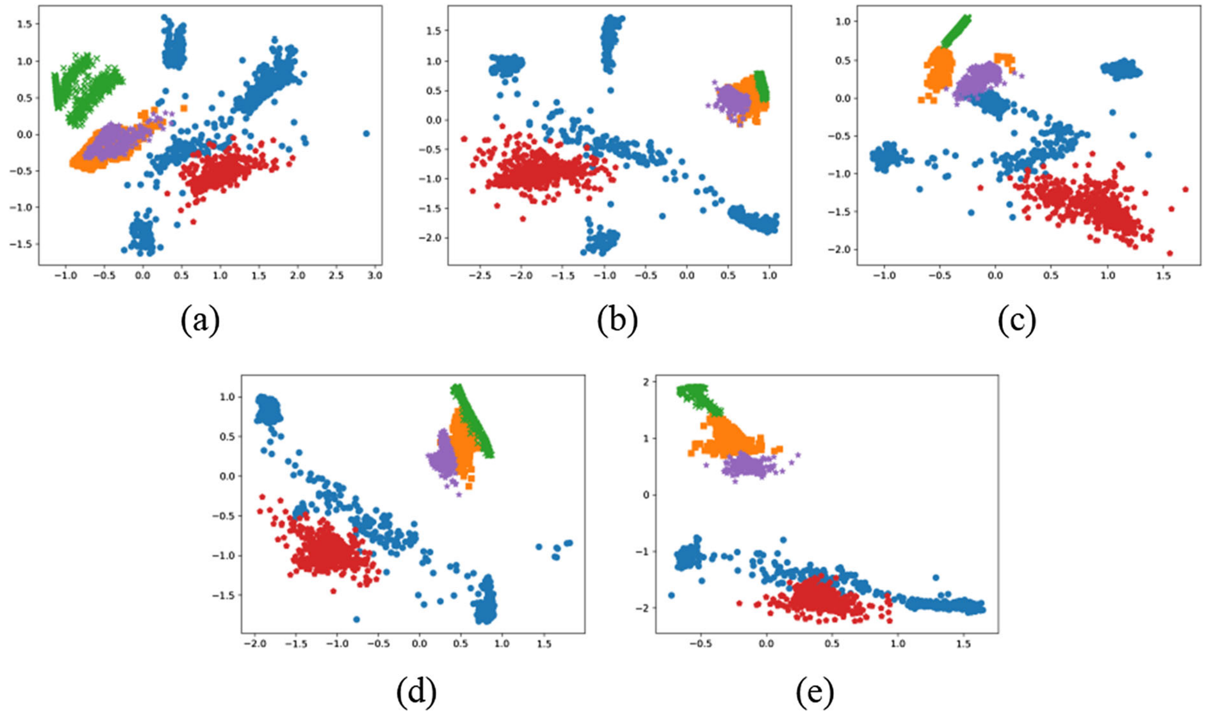


Fig. 12 Visualizations of the FCN-based feature vectors with MDS. (a) The size of 2^6. (b) The size of 2^7. (c) The size of 2^8. (d) The size of 2^9. (e) The size of 1331

dimensional space, and the features show separability when the training set size is gradually increased. In Fig. 13, the features learned by ResNet have a similar change as FCN. It is noteworthy that the distributions of features learned by deep learning methods exhibit significant variations across datasets of different sizes. In contrast, as shown in Fig. 14, the features obtained by deterministic learning show good separability at just the representative subset size of 2^6, and maintain a similar distribution structure at each subsequent size of the dataset. These observations could also explain the accuracy variation of the above methods on different representative subsets.

Moreover, to prove the benefits of the dynamics features learned by deterministic learning, we evaluated the classification performance of different features by using the same classification mechanism, i.e., the nearest neighbor classifier (1-NN).

We calculated the Euclidean distance for the feature vector and then used the 1-NN classifier for classification. The accuracy of different features with 1-NN classifier trained by representative subsets is shown in Fig. 15.

As can be seen from Fig. 15, in the case of using the 1-NN classifier, the feature extracted from deterministic learning method still maintains the highest classification accuracy on the representative subsets. Compared to Fig. 11, an interesting phenomenon can be found that the accuracy of FCN and ResNet has improved to some extent after replacing the 1-NN classifier. In addition, it is noteworthy that the dynamical pattern recognition method based on deterministic learning modified here (which utilizes feature vectors with the 1-NN classifier) achieves a higher classification accuracy of 99.57% compared to the previous accuracy of 99% on the representative subset of 1331 instances.

4.4 Noise analysis

This section further evaluates the robustness of the dynamical pattern recognition method and the deep

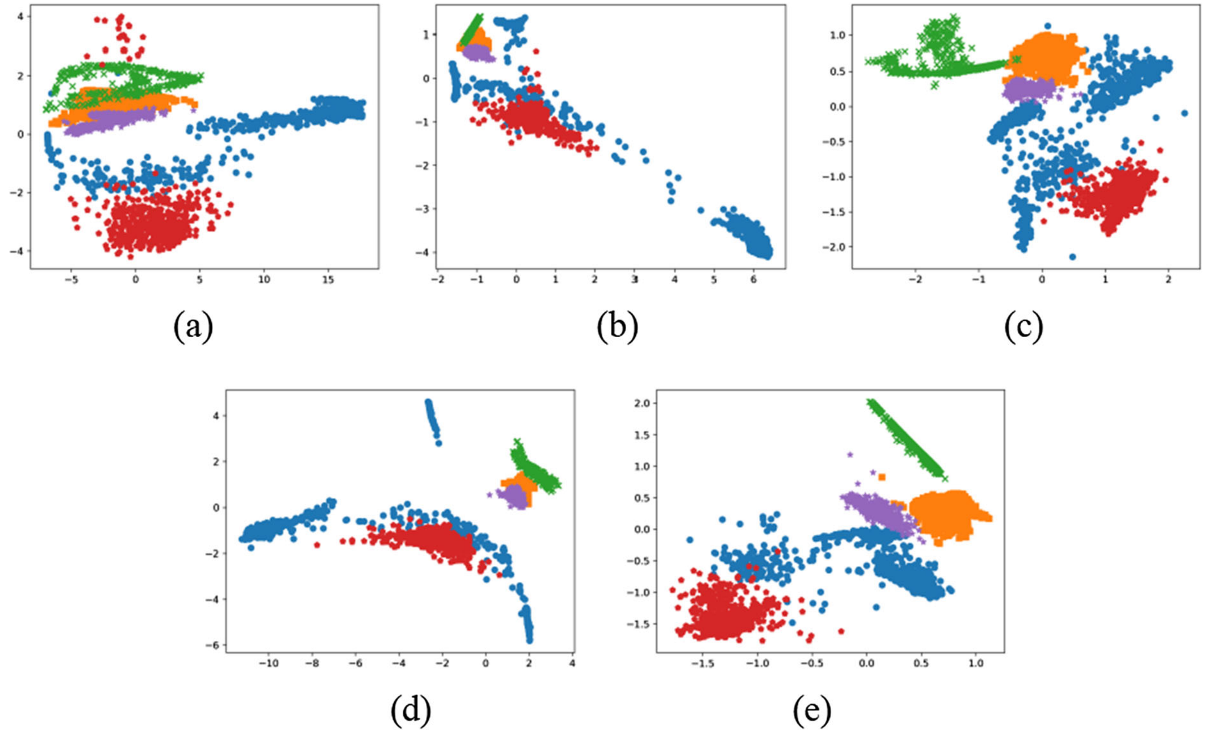


Fig. 13 Visualizations of the ResNet-based feature vectors with MDS. (a) The size of 2^6. (b) The size of 2^7. (c) The size of 2^8. (d) The size of 2^9. (e) The size of 1331

learning methods on different noise levels. We corrupted the test set data with Gaussian noise of different signal-to-noise ratio (SNR) levels. Figure 16 shows an example trajectory after adding different noise levels.

In order to obtain reliable simulation results and avoid being affected by random errors, we randomly generate Gaussian noise for each time series, and repeat the above process 40 times for different SNR levels, in which the Gaussian noise generated each time is different. Table 11 and Table 12 show the test set accuracies of those methods under different noise levels on the original training set (6000 instances) and the representative subset (1331 instances), respectively. The accuracy of each noise level in both tables is the average of all runs. In particular, we also give the confusion matrix of the dynamical pattern recognition method, as shown in Table 13 (on the original training set) and Table 14 (on the representative subset). The confusion matrix shows the classification results of one run close to the average.

As can be seen in Table 11 and Table 12, for the dynamical pattern recognition method based on deterministic learning, when the SNR is 45db, there is almost no drop in the test set classification accuracy on both the original training set and the representative subset, and when the SNR is 30db, the test set classification accuracy only drops by about 1% on average. The results show that this method has good robustness under noisy conditions. For the compared deep learning methods, it also shows strong anti-noise ability. In addition, from both confusion matrices of the dynamical pattern recognition method, it can be seen that the stronger the noise added, the more C_4 and C_5 data are misclassified as C_1 and C_2, respectively.

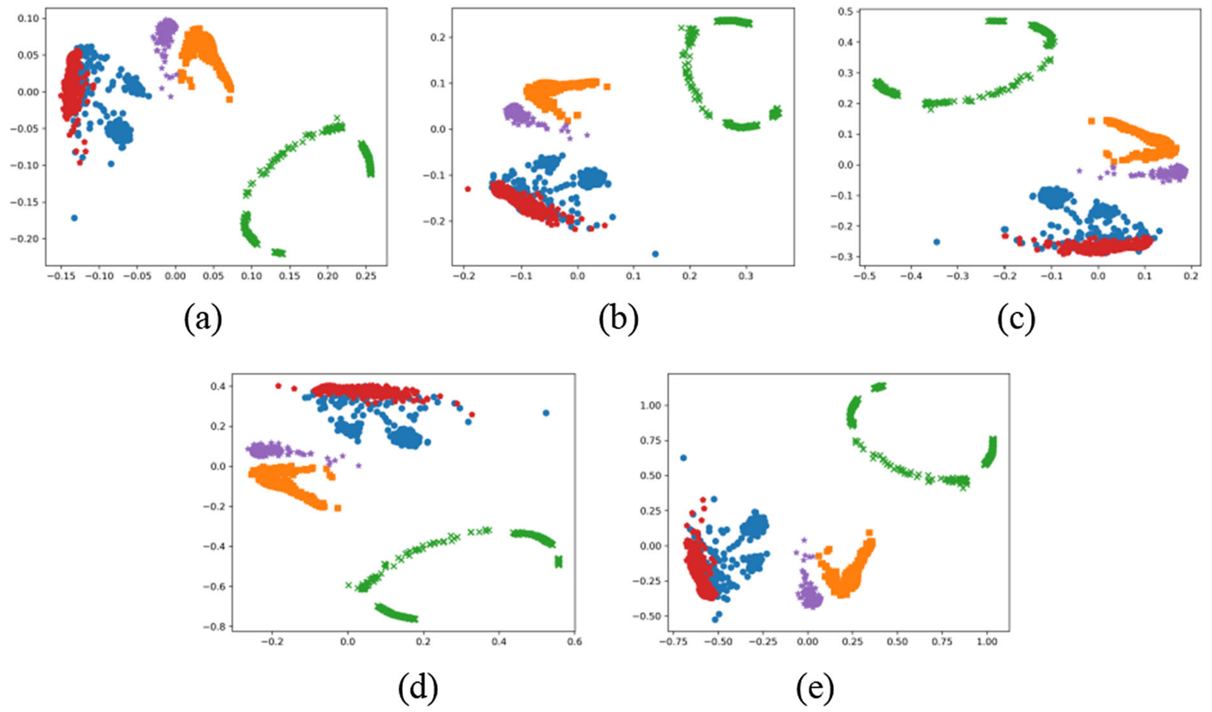


Fig. 14 Visualizations of the deterministic learning-based feature vectors with MDS. (a) The size of 2^6. (b) The size of 2^7. (c) The size of 2^8. (d) The size of 2^9. (e) The size of 1331

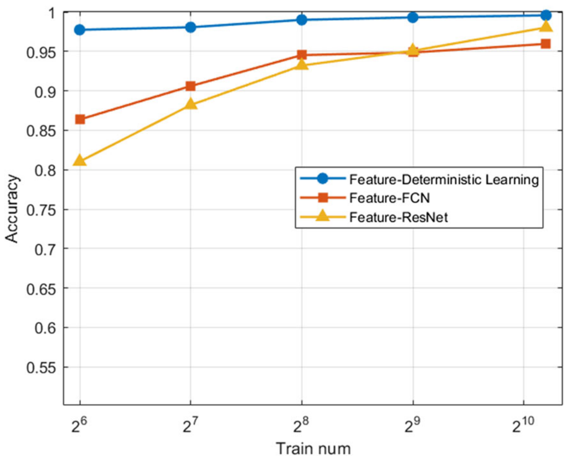


Fig. 15 Comparing the accuracy of different features with 1NN classifier on the representative subsets

5 Conclusion

In this paper, a method based on representative selection and deterministic learning has been proposed for the time series classification of dynamical systems. The proposed method includes a modeling phase and a classification phase. The purpose of the modeling phase is to use the deterministic learning method to model the inherent dynamics of the representative samples. The purpose of the classification phase is to classify the test instances by comparing the dynamic differences between the test instances and training instances. To demonstrate the effectiveness of our method, a large-scale time series dataset has been constructed based on the benchmark Lorenz system, in which data were labeled into 5 classes according to different dynamical behaviors. By using the representative subset, the selected representative samples have been seen to be easier to interpret than the whole dataset, and the cost of storing and computation is reduced, which is beneficial to the application in large-scale datasets. In addition, in the comparison numerical experiments with deep learning, our method has achieved comparable accuracy and has more advantages in terms of interpretability and small sample

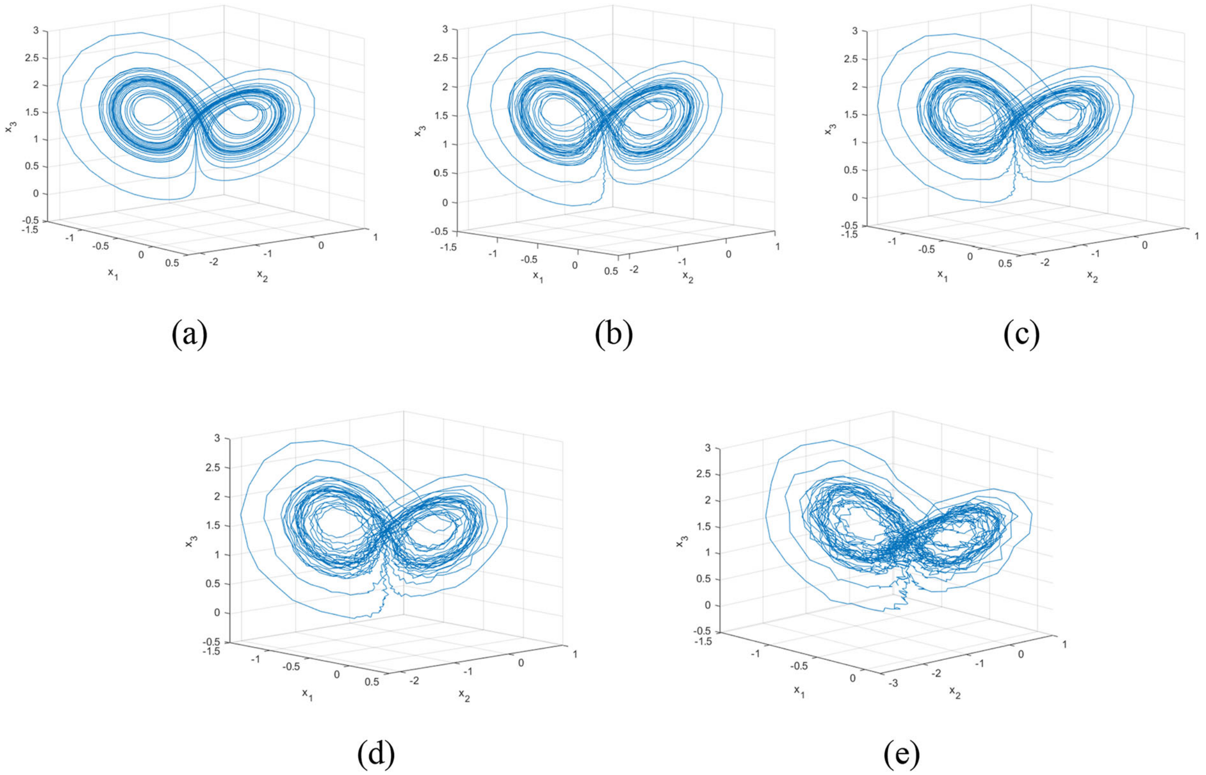


Fig. 16 The trajectory with different noise levels. (a) Data without noise, (b) SNR = 45db, (c) SNR = 40db, (d) SNR = 35db, (e)SNR = 30db

Table 11 Test set accuracy comparison of different noise levels of different methods on the original training set (6000 instances)

	SNR = 45db	SNR = 40db	SNR = 35db	SNR = 30db
MLP	0.5606(± 0.002)	0.5604(± 0.001)	0.5602(± 0.001)	0.5601(± 0.002)
FCN	0.9689(0)	0.9679(± 0.001)	0.9653(± 0.001)	0.9593(± 0.001)
ResNet	0.9897(± 0.001)	0.9884(± 0.001)	0.9860(± 0.001)	0.9784(± 0.001)
This study	0.9902(0)	0.9896(0)	0.9886(± 0.001)	0.9827(± 0.001)

Table 12 Test set accuracy comparison of different noise levels of different methods on the representative subset (1331 instances)

	SNR = 45db	SNR = 40db	SNR = 35db	SNR = 30db
MLP	0.3291(± 0.001)	0.3289(± 0.001)	0.3275(± 0.001)	0.3272(± 0.001)
FCN	0.9291(± 0.001)	0.9258(± 0.001)	0.9177(± 0.001)	0.9078(± 0.002)
ResNet	0.9526(± 0.001)	0.9489(± 0.002)	0.9459(± 0.001)	0.9439(± 0.001)
This study	0.9900(0)	0.9895(0)	0.9878(± 0.001)	0.9799(± 0.001)

Table 13 Confusion matrix of the dynamical pattern recognition method for different levels of noise on the original training set (6000 instances)

	SNR = 45db					SNR = 40db				
	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5
C_1	**695**	0	0	5	0	**693**	0	0	7	0
C_2	19	**681**	0	0	0	19	**681**	0	0	0
C_3	2	0	**698**	0	0	2	0	**698**	0	0
C_4	5	0	0	**695**	0	6	0	0	**694**	0
C_5	0	3	0	0	**697**	0	3	0	0	**697**
	SNR = 35db					SNR = 30db				
	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5
C_1	**693**	0	0	7	0	**694**	0	0	6	0
C_2	19	**680**	0	0	1	19	**679**	0	0	2
C_3	2	0	**698**	0	0	2	0	**698**	0	0
C_4	9	0	0	**691**	0	22	0	0	**678**	0
C_5	0	4	0	0	**696**	0	10	0	0	**690**

Table 14 Confusion matrix of the dynamical pattern recognition method for different levels of noise on the representative set (1331 instances)

	SNR = 45db					SNR = 40db				
	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5
C_1	**695**	0	0	5	0	**695**	0	0	5	0
C_2	19	**681**	0	0	0	19	**681**	0	0	0
C_3	2	0	**698**	0	0	2	0	**698**	0	0
C_4	5	0	0	**695**	0	10	0	0	**690**	0
C_5	0	3	0	0	**697**	0	3	0	0	**697**
	SNR = 35db					SNR = 30db				
	C_1	C_2	C_3	C_4	C_5	C_1	C_2	C_3	C_4	C_5
C_1	**698**	0	0	2	0	**696**	0	0	4	0
C_2	19	**681**	0	0	0	19	**679**	0	0	2
C_3	2	0	**698**	0	0	2	0	**698**	0	0
C_4	13	0	0	**687**	0	31	0	0	**669**	0
C_5	0	4	0	0	**696**	0	11	0	0	**689**

learning. This method is expected to play an important role in biomedical disease diagnosis. The modern view is that human organs such as the heart and brain can be regarded as complex nonlinear dynamical systems, which have chaotic physiological rhythms and are the source of big data generation. The proposed method can be applied to the classification of time series data such as ECG and EEG. As a follow-up, we will further investigate the classification performance of this method on a wider range of datasets.

Funding This study was funded by National Natural Science Foundation of China, 61890922, Cong Wang, 62203263, Weiming Wu, Natural Science Foundation of Shandong Province,ZR2020ZD4, Cong Wang, ZR2022QF062, Weiming Wu.

Data availability The datasets generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare no potential conflicts of interest for the research, authorship, and/or publication of this article.

References

1. Zhou, T., Chen, G.: Classification of chaos in 3-D autonomous quadratic systems-I: basic framework and methods. Int. J. Bifurc. Chaos **16**, 2459–2479 (2006)
2. Dong, C., Liu, H., Jie, Q., Li, H.: Topological classification of periodic orbits in the generalized Lorenz-type system with diverse symbolic dynamics. Chaos Solitons Fractals **154**, 111686 (2022)
3. Boullé, N., Dallas, V., Nakatsukasa, Y., Samaddar, D.: Classification of chaotic time series with deep learning. Physica D D **403**, 132261 (2020)
4. Hassona, S., Marszalek, W., Sadecki, J.: Time series classification and creation of 2D bifurcation diagrams in nonlinear dynamical systems using supervised machine learning methods. Appl. Soft Comput.Comput. **113**, 107874 (2021)
5. Uzun, S.: Machine learning-based classification of time series of chaotic systems. Eur. Phys. J. Spec. Top. **231**, 493–503 (2022)
6. Aricioğlu, B., Uzun, S., Kaçar, S.: Deep learning based classification of time series of Chen and Rössler chaotic systems over their graphic images. Physica D D **435**, 133306 (2022)
7. Yağ, İ., Altan, A.: Artificial Intelligence-Based Robust Hybrid Algorithm Design and Implementation for Real-Time Detection of Plant Diseases in Agricultural Environments. Biology. 11, (2022)
8. Karasu, S., Altan, A.: Crude oil time series prediction model based on LSTM network with chaotic Henry gas solubility optimization. Energy **242**, 122964 (2022)
9. Karasu, S., Altan, A., Bekiros, S., Ahmad, W.: A new forecasting model with wrapper-based feature selection approach using multi-objective optimization technique for chaotic crude oil time series. Energy **212**, 118750 (2020)
10. Wang, C., Hill, D.J.: Learning from neural control. IEEE Trans. Neural Netw.Netw. **17**, 130–146 (2006)
11. Wang, C., Hill, D.J.: Deterministic learning and rapid dynamical pattern recognition. IEEE Trans. Neural Netw.-Netw. **18**, 617–630 (2007)
12. Wang, C., Hill, D.J.: Deterministic learning theory for identification, recognition, and control. CRC Press, Boca Raton, FL (2009)
13. Wang, C., Dong, X., Ou, S., Wang, W., Hu, J., Yang, F.: A new method for early detection of myocardial ischemia: cardiodynamicsgram (CDG). Sci. China Inf. Sci. **59**, 1–11 (2016)
14. Chen, T., Wang, C., Chen, G., Dong, Z., Hill, D.J.: Small Fault Detection for a Class of Closed-Loop Systems via Deterministic Learning. IEEE Trans Cybern. **49**, 897–906 (2019)
15. Wu, W., Wang, Q., Yuan, C., Wang, C.: Rapid dynamical pattern recognition for sampling sequences. Sci. China Inf. Sci. **64**, 132201 (2021)
16. Wu, W., Zhang, F., Wang, C., Yuan, C.: Dynamical pattern recognition for sampling sequences based on deterministic learning and structural stability. Neurocomputing **458**, 376–389 (2021)
17. Elhamifar, E., Sapiro, G., Vidal, R.: See all by looking at a few: Sparse modeling for finding representative objects. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1600–1607 (2012)
18. Balcázar, J., Dai, Y., Watanabe, O.: A random sampling technique for training support vector machines. In: Lecture Notes in Computer Science. pp. 119–134. Springer Berlin Heidelberg, Berlin, Heidelberg (2001)
19. Zhu, F., Ye, N., Yu, W., Xu, S., Li, G.: Boundary detection and sample reduction for one-class Support Vector Machines. Neurocomputing **123**, 166–173 (2014)
20. Park, J., Sandberg, I.W.: Universal Approximation Using Radial-Basis-Function Networks. Neural Comput.Comput. **3**, 246–257 (1991)
21. Gorinevsky, D.: On the persistency of excitation in radial basis function network identification of nonlinear systems. IEEE Trans. Neural Netw.Netw. **6**, 1237–1244 (1995)
22. Wu, W., Wang, C., Yuan, C.: Deterministic learning from sampling data. Neurocomputing **358**, 456–466 (2019)
23. Wolf, A., Swift, J.B., Swinney, H.L., Vastano, J.A.: Determining Lyapunov exponents from a time series. Physica D D **16**, 285–317 (1985)
24. Mitkowski, P., Mitkowski, W.: Ergodic theory approach to chaos: Remarks and computational aspects. Int. J. Appl. Math. Comput. Sci.Comput. Sci. **22**, 259–267 (2012)
25. Lorenz, E.N.: Deterministic nonperiodic flow. Journal of atmospheric sciences. **20**, 130–141 (1963)
26. Sparrow, C.: The Lorenz equations: Bifurcations, chaos, and strange attractors. Springer, New York, NY (1982)
27. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Discov. **31**, 606–660 (2017)
28. Wang, Z., Yan, W., Oates, T.: Time series classification from scratch with deep neural networks: A strong baseline. In: 2017 International Joint Conference on Neural Networks (IJCNN). pp. 1578–1585 (2017)
29. www.github.com/hfawaz/dl-4-tsc.
30. Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A.: Deep learning for time series classification: a review. Data Min. Knowl. Discov. **33**, 917–963 (2019)
31. Ratanamahatana, C.A., Keogh, E.: Three myths about dynamic time warping data mining. In: Proceedings of the 2005 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, Philadelphia, PA (2005)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.