Information Sciences 373 (2016) 1–20

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Functional echo state network for time series classification

Qianli Ma[a,c,*], Lifeng Shen[a], Weibiao Chen[b], Jiabin Wang[a], Jia Wei[a], Zhiwen Yu[a]

[a] School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China
[b] Zhanjiang Air Traffic Management Station, Civil Aviation Administration of China, Zhanjiang 524017, China
[c] Department of Computer Science and Engineering, University of California, San Diego, CA 92093, USA

ARTICLE INFO

Article history:
Received 26 January 2016
Revised 25 August 2016
Accepted 26 August 2016
Available online 31 August 2016

Keywords:
Echo state network
Time series classification
Temporal aggregation
Functional space

ABSTRACT

Echo state networks (ESNs) are a new approach to recurrent neural networks (RNNs) that have been successfully applied in many domains. Nevertheless, an ESN is a predictive model rather than a classifier, and methods to employ ESNs in time series classification (TSC) tasks have not yet been fully explored. In this paper, we propose a novel ESN approach named functional echo state network (FESN) for time series classification. The basic idea behind FESN is to replace the numeric variable output weights of an ESN with time-varying output-weight functions and introduce a temporal aggregation operator to the output layer that can project temporal signals into discrete class labels, thereby transforming the ESN from a predictive model into a true classifier. Subsequently, to learn the output-weight functions, a spatio-temporal aggregation learning algorithm is proposed based on orthogonal function basis expansion. By leveraging the nonlinear mapping capacity of a reservoir and the accumulation of temporal information in the time domain, FESN can not only enhance the separability of different classes in a high-dimensional functional space but can also consider the relative importance of temporal data at different time steps according to dynamic output-weight functions. Theoretical analyses and experiments on an extensive set of UCR data were conducted on FESN. The results show that FESN yields better performance than single-algorithm methods, has comparable accuracy with ensemble-based methods and exhibits acceptable computational complexity. Interestingly, for some time series datasets, we visualized some interpretable features extracted by FESN via specific patterns within the output-weight functions.

1. Introduction

A time series is defined as a sequence of numerical data points in successive order, usually occurring at uniform time intervals. Time series classification (TSC) is the task of extracting the best discriminating features from the time series data and detecting their patterns. TSC has attracted great interest from the machine learning and data mining communities and has been studied in many applications such as brain diagnosis [9,17,50], voltage stability assessment [63], stock market analysis [28], and so on. Unlike traditional classification problems featuring variables that are independent of their relative positions, TSC has the specific challenge of the ordering variables being highly correlated, which leads to the loss of some important information if the traditional machine learning algorithms treat each variable as an independent attribute.

* Corresponding author.
E-mail address: qianlima@scut.edu.cn (Q. Ma).

http://dx.doi.org/10.1016/j.ins.2016.08.081
0020-0255/

In recent decades, a large amount of research has focused on temporal data classification, and many efficient techniques have been proposed. Those techniques can be roughly divided into three categories: 1) Distance-based methods (also called instance-based methods [5]) estimate a time series instance based on its distance to the tagged samples. Examples include 1-Nearest Neighbor (1NN) with Euclidean distance (ED) [18,24], Dynamic Time Warping (DTW) [6,34,38,40,60], Edit Distance on Real sequence (EDR) [13], Edit Distance with Real Penalty (ERP) [12], Fixed Cardinality Warping Distances [51], Classification trees [23], Longest Common Subsequence (LCSS) [33,56], the Dissimilarity (DISSIM) model [27], Sequence Weighted Alignment (Swale) [49], Spatial Assembling Distance (SpADe) [15], and Geometric Template Matching (GeTeM) [26]. 2) Feature-based methods extract various representative features (such as scaling properties, entropy and distribution [21,29]) from a time series for classification. Examples include Discrete Fourier Transformation (DFT) [24], Discrete Wavelet Transformation (DWT) [8], Pattern Extraction [31], Shapelet [32,44,59], Bag-of-features (BoF) [5], Kernel sparse representation [16], Structural Generative Descriptions (SGDs) [30], and DTW-distance features [39]. 3) In contrast to the single-algorithm methods listed in (1) and (2), ensemble-based methods that create collections of different classifiers to achieve higher accuracy have recently been developed. The Elastic Ensemble (EE) [43] is a combination of 1NN classifiers based on 11 elastic distance measures. The final classification is obtained by a voting scheme that assigns weights according to cross-validation training set accuracy. The Shapelet ensemble (SE) [3] is a shapelet transformation based on a heterogeneous ensemble. The collective of transformation-based ensembles (COTE) method [3] involves pooling 35 different classifiers into a single ensemble based on time and frequency domains features and has achieved state-of-the-art accuracy performance [1].

All three categories of classification methods have achieved good performance in many domains. In particular, 1NN with DTW is difficult to beat [4], and shapelet classification can have high accuracy and acceptable interpretation for time series [63]. However, distance-based methods and feature-based methods need to explicitly define distances or features. There is evidence showing that no distance is suitable for all time series data [22,62]. Although ensemble-based methods tend to have higher accuracy than many single-algorithm methods, they suffer from high complexity and become computationally inefficient when faced with the big data volumes common in real-world applications. Furthermore, the development of specific features for each task depends heavily on domain or expert data knowledge and is time-consuming, labor-intensive, and empirical. Therefore, it is a challenge to automatically find the best discriminating features and a suitable distance for a time series classification task.

In recent years, echo state networks (ESNs) [37] have emerged as novel recurrent neural networks (RNNs) that can efficiently process the temporal dependency of time series [46] with high nonlinear mapping capacity and dynamic memory. An ESN has sparse random connections in its hidden layer (reservoir), and the only parameters are the output weights, which can be adapted using a simple linear regression. Therefore, ESNs and its variants have been successfully applied to time series prediction [10,19,37,42,45,47,48], resulting in accuracies several orders of magnitude higher than previous techniques.

However, how to employ an ESN in TSC tasks has not yet been fully explored. Little research effort has been devoted to study ESNs applied to TSC. M. D. Skowronski et al. [53,54] proposed a predictive ESN classifier to classify time series in automatic speech recognition. L. Wang et al. [57] proposed a multivariate TSC model, Conceptor-ADE (CADE), which is a combination of an ESN [36] and the adaptive differential evolution (ADE) algorithm. Recently, H. Chen et al. [11] proposed a model metric co-learning (MMCL) approach for TSC, in which the core model is an ESN using a cycle reservoir with jumps (CRJ) [52]. However, these ESN-based methods are still essentially predictive models rather than classifiers. Their methodologies involve either setting a real-valued output (not a class label) for each class and voting for the corresponding class by analyzing the scales of each output predicted value, or training different predictors and assigning a class label corresponding to the predictor that could predict it best [25,46,61]. In actuality, these are not time series classifiers, which should project temporal signals into discrete class labels.

In this paper, we propose a novel ESN approach named Functional Echo State Network (FESN) for time series classification. We regard the neural states $X(t)$ in the reservoir as functions of time, and then, from a functional analysis point of view, these functions become the points in a high-dimension functional space. Therefore, the problem of TSC is equivalent to seeking the optimal discriminating hyperplanes in the functional space (this is where the term‘functional’in FESN is derived from). FESN can be trained in an end-to-end manner without requiring any predefined distances or handcrafted features. Our contributions can be summarized as follows:

(1) We replace the numeric variable readout weights of ESNs with time-varying functions, which are dynamic weighting functions. These output-weight functions take the relative importance of temporal data at different time steps into consideration.
(2) We introduce a special operator named‘temporal aggregation’, which is the integral of the product of the reservoir's neural states $X(t)$ and the output-weight functions, mapping the neural state functions $X(t)$ from functional space into real number space. This transforms the ESN from a predictive model into a true classifier.
(3) To learn the output-weight functions, a spatio-temporal aggregation learning algorithm is proposed based on orthogonal function basis expansion.
(4) FESN leverages the nonlinear mapping capacity of the reservoir and the accumulation of temporal information in the time domain, which can not only enhance the separability of different classes in a high-dimensional functional space but also consider the relative importance of temporal data at different time steps according to the dynamic output-weight functions.

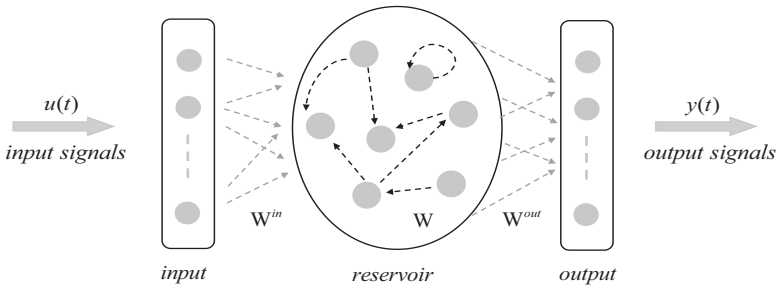

Fig. 1. The architecture of an Echo State Network.

(5) Theoretical analysis and extensive experiments conducted on UCR data sets show that FESN has competitive or superior performance compared with previous methods and acceptable time complexity. For some time series datasets, we extract various interpretable features with FESN and visualize the specific patterns in the output-weight functions.

The remainder of this paper is organized as follows. First, we briefly introduce the standard ESN in Section 2. Then, we give the definitions of the temporal and spatial aggregation operators and related concepts in Section 3. In Section 4, we describe the proposed FESN in detail, including its structure, learning algorithm and time complexity. Section 5 demonstrates the efficiency of FESN using the results of comparison experiments on UCR datasets and the effects of the concerned parameters. Various experiments are conducted to further explore the interpretable features extracted by FESN and visualize the specific patterns in the output-weight functions. Finally, conclusions are presented in Section 6.

2. Echo state networks

An ESN consists of three basic components: an input layer, a large fixed reservoir and an output layer. The general architecture of an ESN is illustrated in Fig. 1.

Here, we consider a simple ESN without the input-to-output and the output-to-reservoir connections. If K, N and L are the numbers of input, internal and output units, respectively, the input-to-reservoir, reservoir and reservoir-to-output weights are collected by an N-by-K matrix $\mathbf{W}^{in}$, an N-by-N matrix $\mathbf{W}$ and an L-by-N matrix $\mathbf{W}^{out}$. The matrices $\mathbf{W}^{in}$, $\mathbf{W}$ and $\mathbf{W}^{out}$ are initialized randomly, and only the matrix $\mathbf{W}^{out}$ needs to be trained.

The ESN is trained by supervised learning process. Two main steps are involved. The first step is to update the reservoir states, and the other step is to learn the weight matrix $\mathbf{W}^{out}$ of the reservoir-to-output layer. The equations for the entire system are as follows:

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{W}\mathbf{x}(t) + \mathbf{W}^{in}\mathbf{u}(t+1)) \tag{1}$$

$$\mathbf{y}(t+1) = \mathbf{f}^{out}(\mathbf{W}^{out}\mathbf{x}(t+1)) \tag{2}$$

where $\mathbf{u}$, $\mathbf{x}$ and $\mathbf{y}$ are the inputs, the internal states and the outputs, respectively; $\mathbf{f}$ contains the activation functions of the internal units (usually the *tanh* function in ESN); and $\mathbf{f}^{out}$ contains the linear activation functions of the output units.

In the updating step, the ESN projects the input signals into the high-dimensional state spaces in the reservoir, and in the training step, the weight matrix $\mathbf{W}^{out}$ can be learned by linear regression. Some research studies have indicated that the reservoir has the same function as the kernel in kernel-based learning methods [7,41] and is able to incorporate temporal information present in the inputs [55]. From this perspective, there are three main characteristics that distinguish an ESN from a traditional RNN:

(1) Input signals drive the reservoir and produce an echo response in a high-dimensional space, which enables the reservoir to have the same function as that of the kernel in kernel-based learning methods [7,41].
(2) The large reservoir with fixed weights avoids the vanishing gradient and exploding gradient problems that exist in conventional RNNs.
(3) The output signal is a linear combination from the reservoir, and simple linear regression algorithms can compute the linear readout layer weights.

Therefore, training an ESN is both simple and fast, and it will not get stuck in local minima, which endows it with high computational capabilities for modeling temporal information.

3. Temporal and spatial aggregation operators

The concept of aggregation originates from the description of the biological brain. The brain is a large-scale interconnected network that consists of billions of neurons. Among these neurons, biochemical signals can be transmitted from

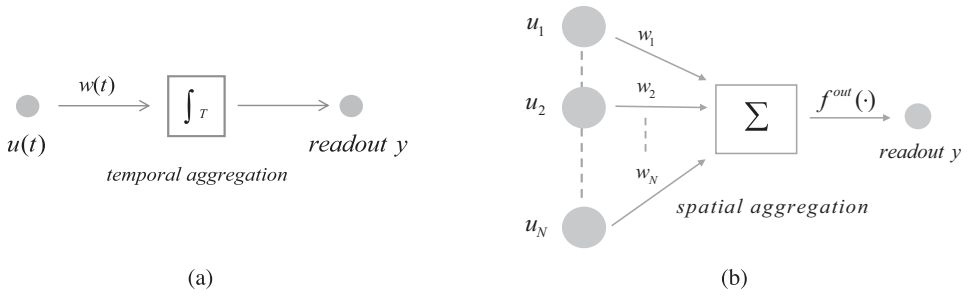

Fig. 2. (a) The temporal aggregation operator. (b) The spatial aggregation operator.

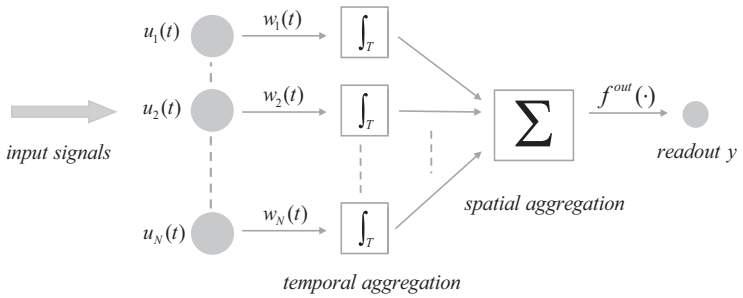

Fig. 3. The process of spatio-temporal aggregation.

multiple neurons to others. Each one receives signals from multiple dendrites and processes the weighted aggregation result nonlinearly. Furthermore, output signals on some special neurons can be activated according to the neurons'threshold levels. This is called information aggregation [35] and includes two aspects: temporal aggregation and spatial aggregation.

Temporal aggregation was first proposed by X. G. He to efficiently process temporal data in the process neuron network (PNN) [35]. The process of temporal aggregation with an input temporal signal $u(t)$ driven in a single unit is shown in Fig. 2(a):

In Fig. 2, $w(t)$ is the weight function, and $\int_T$ is the temporal aggregation operator. The mathematical model of temporal aggregation is given by the following:

$$y = \int_{t=0}^{T} w(t)u(t)dt \tag{3}$$

where T denotes the length of the input signal, $y \in \mathbb{R}$.

As shown in Eq. 3, similar to a biological neuron processing signals by weighted aggregation of signals from multiple dendrites, the operator accumulates the dynamic weighting of the input signal $u(t)$ and maps it into a real-value readout, y. Therefore, it can be regarded as a functional operator that accepts a function and outputs a real number.

Spatial aggregation is an operator widely used in artificial neural networks (ANNs). For example, let Σ denote the spatial aggregation operator. The spatial aggregation is shown in Fig. 2(b), where f^{out} is the activation function of the corresponding output unit. Similarly, the mathematical model of spatial aggregation is given in Eq. 4, where θ is a threshold in f^{out}, (omitted in Fig. 2(b)):

$$y = f^{out}\left(\sum_{i=1}^{N} w_i u_i - \theta\right) \tag{4}$$

where N is the number of inputs.

There are two distinct differences between temporal aggregation and spatial aggregation: (1) the former mainly accumulates the information from time-varying input signals $u(t)$ in the time domain, while the latter aggregates the information from the inputs, u_i, that are time independent; and (2) the temporal aggregation accumulates temporal information in a single unit, while spatial aggregation is the linear combination of multiple input neurons. Briefly, temporal aggregation characterizes the temporal information of input signals with dynamic weighting functions, $W(t)$, while spatial aggregation obtains joint information from multiple independent inputs with static weights, w_i.

In a biological neuron, the output is related to the relative timing of the input pulse. To simulate a biological brain, the temporal aggregation operator needs to be added to an ANN. The combination of temporal aggregation and spatial aggregation, called spatio-temporal aggregation, is shown in Fig. 3.

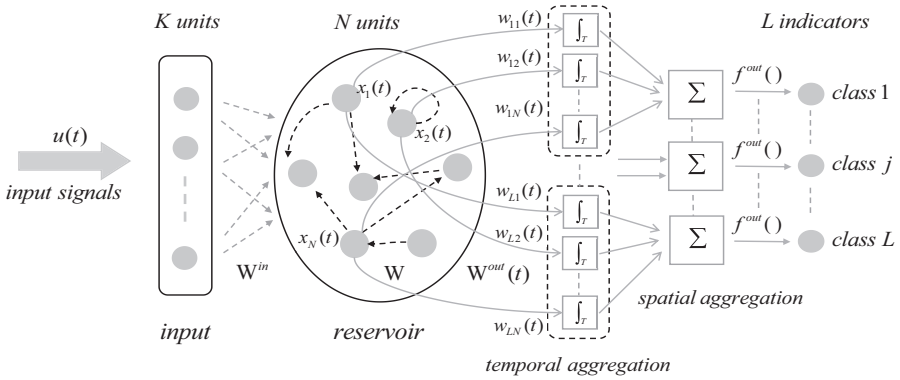

Fig. 4. The general structure of FESN.

In this manner, the output of spatio-temporal aggregation can be given as follows:

$$y = f^{out}\left(\sum_{j=1}^{N}\int_{t=0}^{T} w_j(t)u_j(t)dt - \theta\right) \tag{5}$$

Given the teacher signal d, the loss function is defined by Eq. 6:

$$F(\mathbf{W}(t)) = \min_{W(t)} \|y - d\|^2 \tag{6}$$

Note that learning task $F(\mathbf{W}(t))$ can be regarded as a functional problem because its arguments are the matrix $\mathbf{W}(t)$, which is made up of weight functions $\{w_i(t), i = 1, 2, \ldots, N\}$. Through this functional learning task, the spatio-temporal aggregation operator can yield the following benefits:

(1) It takes a function for its input argument, and its output is a real number; Thus, it can project temporal signals into a discrete class label.
(2) It can effectively accumulate temporal information in the time dimension due to the dynamic weights $\mathbf{W}(t)$ that consider the relative importance of temporal data at different time steps.

However, although adding the temporal aggregation into a feed forward neural network (FFNN) can work well, for a conventional RNN, learning of the weight functions is a very complicated and time-consuming task. To integrate the training simplicity of an ESN with the temporal aggregation operator, in Section 4.2, we propose a spatio-temporal aggregation learning algorithm that makes learning the weight functions possible in RNNs.

4. Functional echo state network

This section describes the proposed FESN for time series classification. The basic idea behind FESN is to introduce a temporal aggregation operator into the output layer and replace the numeric variables of the output weights in an ESN with time-varying functions, which allow the ESN to map the reservoir results from the functional field into the real number field and, thus, transforms the ESN from a regressive model into a classifier. Then, to make the learning process tractable, a spatio-temporal aggregation learning algorithm is proposed to approximate the output-weight functions with orthogonal function basis expansion, which can obtain the optimal discriminating weight function in high-dimensional functional spaces. The details of the proposed method are presented as follows.

4.1. Model formulation

The structure of FESN, which has K input units, a large, sparse and fixed reservoir with N internal units and L output units for L classes is shown in Fig. 4.

Identical to an ESN, FESN also needs to randomly initialize the input-to-reservoir weight matrix, $\mathbf{W}^{in}$, and the internal units weight matrix, $\mathbf{W}$. After the initialization stage, the input signals $\mathbf{u}(t)$ will be input into the FESN; then, the state of the reservoir is constantly updated. The updating rules are the same as in Eq. 1:

$$\mathbf{x}(t+1) = \mathbf{f}(\mathbf{W}\mathbf{x}(t) + \mathbf{W}^{in}\mathbf{u}(t+1)), (t = 0, 1, \ldots, T-1) \tag{7}$$

where $\mathbf{x}(t) = [x_1(t), x_2(t), \ldots, x_N(t)]^T$ denotes the state of the internal units at time t, each element $x_i(t)$ is the state of the ith internal unit at time t, and $\mathbf{f}(\cdot)$ is the vector of the activation function in the reservoir-to-output layer.

Here, we regard the internal neural state $\mathbf{x}(t)$ as a function of time with N dimensions in a functional space. Therefore, if we replace the numerical variables of the output weights with the time-varying functions, we can use temporal aggregation

operators on each internal unit of the reservoir and regard $\mathbf{x}(t)$ as the input $u(t)$ of the temporal aggregation operator $\int_T$ in Eq. 3. The aggregation result of the ith internal unit is

$$\eta_{ji} = \int_0^T w_{ji}(t)x_i(t)dt \tag{8}$$

where $x_i(t)$ $(i = 1, 2, \ldots, N)$ denotes the state of the ith internal unit at time step t and $w_{ji}(t)$ $(i = 1, 2, \ldots, N, j = 1, 2, \ldots, L)$ is the weight function of the ith internal unit to the jth output unit.

As shown in Fig. 4, the jth spatial aggregation result Σ_j is

$$\Sigma_j = \sum_{i=1}^{N} \eta_{ji} \tag{9}$$

Hence, the output corresponding to the jth class could be

$$y_j = f^{out}(\Sigma_j) = f^{out}\left(\sum_{i=1}^{N}\int_0^T w_{ji}(t)x_i(t)dt\right) \tag{10}$$

where the threshold θ in f^{out} is set to zero for simplicity.

To rewrite Eq. 10 in a more concise form, we define the matrix of the output-weight functions, $\mathbf{W}^{out}(t)$ as

$$\mathbf{W}^{out}(t) = \begin{bmatrix} w_{11}(t) & w_{12}(t) & \cdots & w_{1N}(t) \\ w_{21}(t) & w_{22}(t) & \cdots & w_{2N}(t) \\ \vdots & \vdots & \ddots & \vdots \\ w_{L1}(t) & w_{L2}(t) & \cdots & w_{L1}(t) \end{bmatrix}_{L\times N} \tag{11}$$

where $w_{ji}(t)$ $(i = 1, 2, \ldots, N, j = 1, 2, \ldots, L)$ is the weight function of the ith internal unit to the jth output unit, and the state vector of the internal units in the reservoir is denoted as

$$\mathbf{x}(t) = \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_N(t) \end{bmatrix}^T_{N\times 1} \tag{12}$$

so that the vector of the final output y can be given by

$$\mathbf{y} = \mathbf{f}^{out}\left(\int_{t=0}^{T} \mathbf{W}^{out}(t)\mathbf{x}(t)dt\right) \tag{13}$$

where $\mathbf{f}^{out}$ is the vector of the activation functions of the output units.

From Eq. 13, classifying time series is equivalent to seeking a mapping from functional space to real space, such that the optimization goal of FESN is to find an appropriate matrix of the output-weight functions $\mathbf{W}^{out}(t)$ representing the discriminating hyperplane in an N-dimensional functional space. We propose a spatio-temporal aggregation algorithm for learning $\mathbf{W}^{out}(t)$ in the following subsection.

4.2. Learning the weight functions

N_{train} samples of the time series $\{\mathbf{u}^{(m)}(t),\ \mathbf{d}^{(m)}\}$ $(m = 1, 2, \ldots N_{train},\ t \in 0, 1, \ldots, T-1)$ are provided as the training set, where $\mathbf{d}^{(m)}$ is an L-by-1 teacher vector obtained by the 1-of-L coding method. Under these conditions, the update process and the spatio-temporal aggregation in FESN can be given by

$$\mathbf{x}^{(m)}(t+1) = \mathbf{f}(\mathbf{W}\mathbf{x}^{(m)}(t) + \mathbf{W}^{in}\mathbf{u}^{(m)}(t+1)),\ \ t = 0, 1, \ldots, T-1 \tag{14}$$

$$y_j^{(m)} = f^{out}\left(\sum_{i=1}^{N}\int_0^T w_{ji}(t)x_i^{(m)}(t)dt\right), m = 1, 2, \ldots, N_{train}, j = 1, 2, \ldots, L \tag{15}$$

where $x_i^{(m)}(t)$ $(i = 1, 2, \ldots, N)$ is the ith internal unit state driven by the mth input signal. In this manner, for an L-class TSC problem, the loss function in FESN is given by

$$F(\mathbf{W}^{out}(t)) = \min \sum_{m=1}^{N_{train}} \|\mathbf{y}^{(m)} - \mathbf{d}^{(m)}\|^2 \tag{16}$$

Eq. 16 can be regarded as a functional operator because its input argument is the matrix $\mathbf{W}(t)$, which is made up of output-weight functions $\{w_i(t), i = 1, 2, \ldots, N\}$ that could be learned by orthogonal basis expansion.

4.2.1. Step 1: Orthogonal basis expansion of $x_i^{(m)}(t)$ and $w_{ji}(t)$

The standard orthogonal basis functions (e.g., Fourier orthogonal basis, wavelet basic function, etc.) are denoted as

$$\varphi_1(t), \varphi_2(t), \ldots, \varphi_r(t), \ldots \tag{17}$$

where $\varphi_r(t) \in C[0, T]$ and C[0,T] denote the set of all continuous real-valued functions on [0,T]. They must satisfy an orthogonality constraint such that

$$\int_0^T \varphi_n(t)\varphi_m(t)dt = \begin{cases} 1 & (n = m) \\ 0 & (n \neq m) \end{cases} \tag{18}$$

where $m, n \in \mathbb{Z}$. Specifically, we choose the Fourier orthogonal basis in this paper, which has the following forms:

$$\frac{1}{\sqrt{2\pi}}, \frac{1}{\sqrt{\pi}}\cos t, \frac{1}{\sqrt{\pi}}\sin t, \ldots, \frac{1}{\sqrt{\pi}}\cos nt, \frac{1}{\sqrt{\pi}}\sin nt, \ldots \tag{19}$$

The period T should be rescaled into 2π to satisfy the orthogonality. Subsection 5.2 discusses how to choose an appropriate orthogonal basis in FESN.

According to the orthogonal decomposition of the continuous signals, the ith internal unit state function $x_i^{(m)}(t)$ driven by the mth input signal can be expressed as the series form of the orthogonal basis expansion

$$x_i^{(m)}(t) = \sum_{r=1}^{\infty} a_{ir}^{(m)} \varphi_r(t) \tag{20}$$

where $a_{ir}^{(m)}$ is the coefficient of the rth orthogonal basis.

Here, we select the first R basis to expand $x_i^{(m)}(t)$:

$$x_i^{(m)}(t) \approx \sum_{r=1}^{R} a_{ir}^{(m)} \varphi_r(t) \tag{21}$$

where R should be a positive integer to guarantee that, for any $\varepsilon > 0$, there exists an R_0 that is large enough such that, for any $R \geq R_0$, we have

$$|x_i^{(m)}(t) - \sum_{r=1}^{R} a_{ir}^{(m)} \varphi_r(t)| \leq \varepsilon \tag{22}$$

In a similar manner, $w_{ji}(t)$ can be expanded based on the basis functions $\varphi_{r'}(t)(r' = 1, 2, \ldots, R)$ as

$$w_{ji}(t) \approx \sum_{r'=1}^{R} b_{ji(r')} \varphi_{r'}(t) \tag{23}$$

where $b_{ji(r')}$ is the coefficient of the r'th orthogonal basis.

4.2.2. Step 2: Learning by linear regression

Plugging Eq. 21 and Eq. 23 into Eq. 15, we obtain

$$y_j^{(m)} = f^{out}\left(\sum_{i=1}^{N}\int_0^T (\sum_{r'=1}^{R} b_{ji(r')}\varphi_{r'}(t))(\sum_{r=1}^{R} a_{ir}^{(m)}\varphi_r(t))dt\right), m = 1, 2, \ldots, N_{train}, j = 1, 2, \ldots, L \tag{24}$$

Assuming that f^{out} is an identity function, Eq. 24 can be simplified as

$$y_j^{(m)} = \sum_{i=1}^{N}\sum_{r=1}^{R}\sum_{r'=1}^{R} a_{ir}^{(m)} b_{ji(r')} \int_0^T \varphi_r(t)\varphi_{r'}(t)dt \tag{25}$$

Because $\varphi_r(t)$ and $\varphi_{r'}(t)$ satisfy the orthogonality constraint in Eq. 18, we have

$$y_j^{(m)} = \sum_{i=1}^{N}\sum_{r=1}^{R} a_{ir}^{(m)} b_{ji(r)} \tag{26}$$

where $a_{ir}^{(m)}$ and $b_{ji(r)}$ are the coefficients of the rth orthogonal basis to expand $x_i^{(m)}(t)$ and $w_{ji}(t)$, respectively. Eq. 16, combined with Eq. 26, can be rewritten as

$$F(b_{ji(r)}) = \min_{b_{ji(r)}} \sum_{m=1}^{N_{train}}\sum_{j=1}^{L}\left(\sum_{i=1}^{N}\sum_{r=1}^{R} a_{ir}^{(m)} b_{ji(r)} - d_j^{(m)}\right)^2 \tag{27}$$

where $d_j^{(m)}$ denotes the jth value of the teacher signals $\mathbf{d}^{(m)}$.

Eq. (27) is a least squares problem, and simple linear regression methods (e.g., the pseudo-inverse method) can be used to solve it. For example, we can use the pseudo-inverse method to obtain the optimal estimation of the parameters $b^*_{ji(r)}$ with

$$\mathbf{W}^{out} = \mathbf{M}^{\dagger}\mathbf{D} \tag{28}$$

where the matrix $\mathbf{W}^{out} \in \mathbb{R}^{NR\times L}$ has elements $b^*_{ji(r)}$ $(j = 1, 2, \ldots, L, i = 1, 2, \ldots, N, r = 1, 2, \ldots, R)$, the matrix $\mathbf{M} \in \mathbb{R}^{N_{train}\times NR}$ has elements $a^{(m)}_{ir}$ $(i = 1, 2, \ldots, N, r = 1, 2, \ldots, R, m = 1, 2, \ldots, N_{train})$, and the teacher signals matrix $\mathbf{D} \in \mathbb{R}^{N_{train}\times L}$ has elements $d^{(m)}_j$ $(j = 1, 2, \ldots, L, m = 1, 2, \ldots, N_{train})$. The algorithm for updating and training the FESN is as follows:

Algorithm 1 Updating and Training algorithm of FESN

Input:

Training series and their labels, $\{\mathbf{u}^{(m)}(t), \mathbf{d}^{(m)}\}, t = 1, 2, \ldots, T,\ m = 1, 2, \ldots, N_{train}$
Number of orthogonal basis, R
Spectral radius of reservoir, ρ
Sparsity of reservoir, α
Size of reservoir, N
Categories of time series data, L

Output: The output matrix $\mathbf{W}^{out}$ in the Equation (28)

Transform teacher labels $\{\mathbf{d}^{(m)}\}$ into K by L matrix $\mathbf{D}$
Using the $\{\mathbf{u}^{(m)}(t)\}$, ρ, α, N, L to initialize the FESN, including the matrix $\mathbf{W}$ and $\mathbf{W}^{in}$
for each time series $\mathbf{u}^{(m)}(t)$ $m = 1, 2, \ldots, N_{train}$ **do**
  $T \leftarrow$ the length of $\mathbf{u}^{(m)}(t)$
  $\mathbf{X}^{(m)} \leftarrow$ generate a N by $(T + 1)$ zero matrix
  for each reservoir unit i in $[1, N]$ **do**
    for each time t in $[1, T]$ **do**
      Update states: $\mathbf{X}^{(m)}(i, t) \leftarrow \mathrm{f}(\mathbf{W}(i, :)*\mathbf{X}^{(m)}(i, t-1) + \mathbf{W}^{in}(i, :)*\mathbf{u}^{(m)}(t))$
    end for
    $\mathbf{M}(m, R*(i-1)+1: R*i) \leftarrow$ return the first R coefficients of the orthogonal basis to expand $\mathbf{X}^{(m)}(i, :)$
  end for
end for
$\mathbf{W}^{out} \leftarrow \mathbf{M}^{\dagger}\mathbf{D}$

4.3. Classification testing

With the new series $\mathbf{u}^{new}(t)$ applied, the result of the trained FESN is

$$y = \arg\max_j y_j = \arg\max_j f^{out}\left(\sum_{i=1}^{N}\sum_{r=1}^{R} b^*_{ji(r)} a_{ir}\right), j = 1, 2, \ldots, L \tag{29}$$

where a_{ir} is the coefficient of the rth orthogonal basis and $b^*_{ji(r)}$ is the result from Algorithm 1. The algorithm for testing the new series $\mathbf{u}^{new}$ is shown in Algorithm 2.

4.4. Time complexity analysis

The time complexity of FESN is mainly determined by three processes: updating the internal states, orthogonal expansion and training. In these processes, several parameters influence FESN's time complexity, including N_{train}, the number of training time series; T, the length of each series; L, the number of categories of training data; N, the size of the reservoir; and R, the length of the orthogonal expansion. We describe the time complexity of FESN by Theorem 1.

Theorem 1. *Given an L-class TSC problem, with a training data size of N_{train} and a sample length of T, the update complexity of FESN with an N-unit reservoir is $O(N_{train}NT)$ and the orthogonal decomposition complexity is $O(N_{train}NR\log_2 R)$, where the Fourier orthogonal basis is used as the basis of orthogonal expansion. Training the FESN is equivalent to solving the linear system equations $Ax = b$, $A \in \mathbb{R}^{N_{train}\times NR}$ and $b \in \mathbb{R}^{N_{train}\times L}$, where x is the solution.*

As can be observed from Theorem 1, the first two processes of FESN's time complexity are linear with T and N_{train}, which means that FESN is scalable with regard to the number of time series. In addition, the training of FESN is not related to the parameter T; therefore, it is amenable to learning time series of unequal lengths. Although the time complexity of FESN is proportional to the size of the reservoir, N, and the length of the orthogonal expansion, R, it does not require large N and R values to achieve good performance. In fact, setting $N \leq 1000$ and $R \leq 900$ is sufficient in the following experiments.

Algorithm 2 Testing algorithm of FESN

Input: Testing series data, $\mathbf{u}^{new}(t)$

Output: Result of classification , y

Loading the trained model fesn including the matrices $\mathbf{W}$, $\mathbf{W}^{in}$ and $\mathbf{W}^{out}$
$T \leftarrow$ the length of $\mathbf{u}^{new}(t)$
$\mathbf{X} \leftarrow$ generate a zero matrix of $1 - by - (T + 1)$
for all reservoir units i in $[1, N]$ **do**
  for all time t in $[0, T]$ **do**
    Update states: $\mathbf{X}(i, t) \leftarrow \mathrm{f}(\mathbf{W}(i, :) * \mathbf{X}(i, t-1) + \mathbf{W}^{in}(i, :) * \mathbf{u}^{new}(t))$
  end for
  $\mathbf{M}(1, R*(i-1)+1: R*i) \leftarrow$ return the first R coefficients of the orthogonal decomposition of $\mathbf{X}(i, :)$
end for
for all output units j from 1 to L **do**
  y_j = $\mathbf{M}^*\mathbf{W}^{out}(:, j)$
end for
$y = \arg\max_j y_j$

5. Experiments and results

To verify the effectiveness of FESN for time series classification (TSC), we performed a series of experiments on publicly available time series datasets from the "UCR Time Series Data Mining Archive" [14]. These datasets exhibit different background knowledge in a wide range of application domains, which is useful when testing the learning ability of different time series classifiers. The error rate of classifiers is defined as shown in Eq. 30

$$Error\ rate = \frac{total\ number\ of\ misclassification\ data}{total\ number\ of\ testing\ data} \tag{30}$$

and the accuracy rate of classifiers is calculated as follows:

$$Accuracy\ rate = 1 - Error\ rate \tag{31}$$

5.1. Comparison in time series classification

In these experiments, we compare FESN with four representative types of techniques: (1) distance-based methods, (2) feature-based methods, (3) ensemble-based methods and (4) ESN-based methods. While we did not implement these methods, we can compare the results of FESN with the reported performances of these methods on the same UCR datasets. As comparison data, we used the best reported results from the available literature. Due to the lack of consistency across publications with regard to the number of datasets used when testing these methods, we list their intersection datasets in the following result tables.

For the distance-based methods, FESN is compared with 6 time series classification methods: 1-Nearest Neighbor (1NN) with Euclidean distance (ED) (the standard basic method for TSC), 1-Nearest Neighbor with Dynamic Time Warping (DTW), Sequence Weighted Alignment (Swale), Spatial Assembling Distance (SpADe), a variant of the Longest Common Subsequence using the first and second derivatives ($2DD_{LCSS}$), and Geometric Template Matching (GeTeM). The classification results for 1NN-ED and 1NN-DTW are provided in [14], and the results of Swale and SpADe are reported in [58]. For $2DD_{LCSS}$ and GeTeM, we cite the results reported in [33] and [26], respectively.

For the feature-based methods, 4 appealing time series classification methods have been proposed in recent years. They are TSBF [5] based on a bag-of-features representation, time series forest (TSF) combined with entrance [21], a linear feature-based classifier [29] and a method that uses DTW distances as features (Feature-DTW-DTW-R) [39]. Furthermore, because shapelet transform is a recently proposed and effective method that has good explanatory power [44], we separately compare shapelet transform with FESN. Shapelet transform can be regarded as a preprocessing technique based on features called time series shapelets. It is easy to combine shapelet transform with common classifiers; therefore, in the following experiments, we compare shapelet transform combined with the classifiers of decision trees, Naive Bayes, random forest and SVM, all constructed on the shapelet transform reported in [44].

For the ensemble-based methods, 3 main ensemble schemes are compared with FESN: Shapelet Ensemble (SE) [3], Elastic Ensemble (EE) [43] and the collective of transformation-based ensembles (COTE) [3]. SE is a shapelet transformation based on a heterogeneous ensemble; EE is a combination of 1NN classifiers based on 11 elastic distance measures via a voting scheme; and COTE pools 35 different classifiers into a single ensemble based on features from time and frequency domains. We compare the results of FESN with those of SE, EE and COTE reported in [3,43].

Among the ESN-based methods, one is a simple ESN based on the first strategy mentioned in Section 1, and the other is the metric co-learning model MMCL [11]. In the experiments, the simple predictive classifier ESN is set as follows: the

Table 1
Error rates of 6 distance-based methods and FESN on 32 UCR datasets.

Data Set	♯ of classes	♯ train	♯ test	length	1NN–ED	1NN–DTW	Swale	SpADe	$2DD_{LCSS}$	GeTeM	FESN
50words	50	450	455	270	0.369	0.310	0.281	0.341	**0.251**	0.286	0.376
Adiac	37	390	391	176	0.389	0.396	0.408	0.438	0.575	**0.274**	0.384
Beef	5	30	30	470	0.333	0.333	0.384	0.500	0.467	0.367	**0.000**
CBF	3	30	900	128	0.148	0.003	0.013	0.044	0.012	0.041	**0.000**
ChlorineConcentration	3	467	3840	166	0.350	0.352	0.374	0.439	0.439	0.281	**0.124**
Coffee	2	28	28	286	**0.000**	**0.000**	0.270	0.185	0.107	0.143	**0.000**
DiatomsizeReduction	4	16	306	345	0.065	0.033	0.028	**0.016**	0.118	0.065	0.020
ECG200	2	100	100	96	0.120	0.230	0.170	0.256	0.130	0.200	**0.070**
ECGFiveDays	2	23	861	136	0.203	0.232	0.290	0.265	0.056	**0.012**	0.019
FaceFour	4	24	88	350	0.216	0.170	0.134	0.250	0.182	**0.034**	**0.034**
FacesUCR	14	200	2050	131	0.231	0.095	**0.030**	0.315	0.085	0.085	0.099
Fish	7	175	175	463	0.217	0.177	0.171	0.150	**0.057**	0.063	0.160
GunPoint	2	50	150	150	0.087	0.093	0.066	**0.007**	0.033	0.013	0.013
Haptics	5	155	308	1092	0.630	0.623	0.581	0.736	0.682	**0.542**	0.558
ItalyPowerDemand	2	67	1029	24	0.045	0.050	0.082	0.233	0.067	0.079	**0.028**
Lighting2	2	60	61	637	0.246	**0.131**	0.160	0.272	0.180	0.246	0.148
Lighting7	7	70	73	319	0.425	0.274	**0.090**	0.167	0.452	0.575	0.288
MALLAT	8	55	2345	1024	0.086	0.066	0.279	0.557	**0.060**	0.074	0.296
MedicalImages	10	381	760	99	0.316	**0.263**	0.348	0.434	0.345	0.267	0.343
MoteStrain	2	20	1252	84	0.121	0.165	0.073	0.103	0.152	0.104	**0.066**
OliveOil	4	30	30	570	0.133	0.167	0.097	0.207	0.833	0.300	**0.000**
OSULeaf	6	200	242	427	0.479	0.409	0.403	0.212	**0.141**	**0.141**	0.517
SonyAIBORobotSurface	2	20	601	70	0.305	0.275	0.205	0.195	0.133	0.180	**0.118**
SonyAIBORobotSurfaceII	2	27	953	65	0.141	0.169	0.281	0.322	0.143	0.090	**0.072**
StarLightCurves	3	1000	8236	1024	0.151	0.093	0.120	0.142	**0.039**	0.040	0.160
SwedishLeaf	15	500	625	128	0.211	0.208	0.140	0.254	0.106	0.138	**0.093**
Syntheticcontrol	6	300	300	60	0.120	**0.007**	0.060	0.150	0.060	0.123	0.023
Trace	4	100	100	275	0.240	**0.000**	0.108	0.000	0.040	0.010	**0.000**
TwoLeadECG	2	23	1139	82	0.253	0.096	0.149	0.017	0.069	**0.004**	0.077
TwoPatterns	4	1000	4000	128	0.090	**0.000**	**0.000**	0.052	0.001	0.154	0.247
wafer	2	1000	6174	152	0.005	0.020	**0.004**	0.018	0.005	0.006	0.010
yoga	2	300	3000	426	0.170	0.164	0.430	0.130	**0.123**	0.131	0.198
Win/Lose/Tie					8/23/1	11/19/2	8/24/0	10/21/1	10/22/0	14/16/2	–
Average rank					4.750	3.875	4.125	5.078	3.766	3.406	3.000
Rank difference					1.750	0.875	1.125	2.078	0.766	0.406	–
♯ of best errors					1	6	4	2	6	6	13

size of the reservoir, N, is in the range [300, 1500], the spectral radius of the reservoir, ρ, is in the range [0.9, 1], and the sparsity, α, is 0.01.

In the following experiments, FESN has a reservoir with internal units ranging from 80 to 1000 and L output units corresponding to L-class TSC tasks. The learning strategy of the orthogonal basis in FESN is the Fourier series expansion. The sparsity of the reservoir, α is 0.01, the spectral radius, ρ, is in the range [0.8, 1.2], and the length of orthogonal expansion, R, is in the range [100, 900].

5.1.1. FESN vs. distance-based methods

The classification errors of FESN and 6 distance-based methods on 32 UCR tasks are collated in Table 1.

As shown in Table 1, FESN achieves much higher accuracy than any of the other methods on 13 of the 32 datasets. GeTeM is the most accurate on 6 of the 32 datasets, while 1NN-DTW is the most accurate on 6 of the 32 datasets. In particular, there are some sets such as Beef, Olive Oil, Chlorine Concentration on which the distance-based methods cannot work well but on which FESN performs with lower error rates. For the set Beef, FESN works well, achieving an error rate of 0.0%, but the best results of other classifiers have error rates of 33.3% by 1NN ED, 33.3% by 1NN-DTW and 36.7% by GeTeM. In the Olive Oil task, FESN has a 0.0% error rate, but others have the error rates of only 9.70% (Swale), 13.3% (1NN-ED) and 16.7% (1NN-DTW).

Apart from the error rates, we count the win/loss/tie numbers, and FESN outperforms the other 5 methods. For example, FESN is better on 19 of the 32 datasets, equel on 2 and worse on 11 compared to 1NN-DTW. Compared to $2DD_{LCSS}$, FESN is better on 22 of the 32 data sets and worse on 10. Over all 32 datasets, FESN is comparable to GeTeM, whose win/loss/tie numbers are 14/16/2, respectively.

In addition, we conducted non-parametric tests (Friedman tests, Nemenyi tests) [20] to make statistical comparisons. As shown in Table 1, The average rank of FESN is significantly higher than all the other methods (tested using the Friedman rank test). GeTeM achieves second place, and $2DD_{LCSS}$ ranks third. At a 0.05 significance level, Friedman tests indicate that the differences of each of the models in Table 1 are statistically significant. A critical difference diagram comparing the results of FESN to the other distance-based TSC methods is shown in Fig. 5.

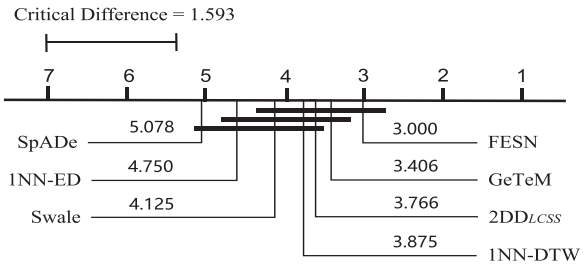

Fig. 5. Comparison of FESN with 6 distance-based classifiers on the Nemenyi test. Groups of classifiers that are not significantly different (at p = 0.05) are connected.

Table 2
Error rates of 4 feature-based methods and FESN on 19 UCR datasets.

Data Set	♯ of classes	♯ train	♯ test	length	TSBF-Unif	TSF Entrance	Feature-linear	F-DTW-DTW-R	FESN
50words	50	450	455	270	**0.211**	0.266	0.453	0.255	0.376
Adiac	37	390	391	176	0.295	**0.230**	0.355	0.325	0.384
Beef	5	30	30	470	0.460	0.233	0.500	0.533	**0.000**
CBF	3	30	900	128	0.004	0.026	0.289	**0.000**	**0.000**
Coffee	2	28	28	286	0.007	0.036	0.179	0.071	**0.000**
ECG200	2	100	100	96	0.207	0.080	**0.010**	0.090	0.070
FaceFour	4	24	88	350	0.048	**0.023**	0.261	0.136	0.034
FaceAll	14	560	1690	131	0.196	0.233	0.292	0.138	**0.096**
Fish	7	175	175	463	**0.056**	0.154	0.171	0.171	0.160
GunPoint	2	50	150	150	0.015	0.047	0.073	0.047	**0.013**
Lighting2	2	60	61	637	0.334	0.180	0.197	**0.112**	0.148
Lighting7	7	70	73	319	0.370	0.260	0.438	**0.233**	0.288
OliveOil	4	30	30	570	0.167	0.067	0.100	0.267	**0.000**
SwedishLeaf	15	500	625	128	**0.088**	0.106	0.227	0.101	0.093
syntheticControl	6	300	300	60	**0.009**	0.027	0.037	0.013	0.023
Trace	4	100	100	275	0.020	0.020	0.010	**0.000**	**0.000**
TwoPatterns	4	1000	4000	128	0.004	0.054	0.074	**0.000**	0.247
wafer	2	1000	6174	152	0.003	0.005	**0.000**	0.006	0.010
yoga	2	300	3000	426	0.156	0.151	0.226	**0.141**	0.198
Win/Lose/Tie					8/11/0	8/11/0	4/15/0	8/9/2	–
Average rank					2.711	2.842	4.132	2.790	2.526
Rank difference					0.148	0.316	1.605	0.263	–
♯ of best errors					4	2	2	6	7

The critical difference in Fig. 5 is 1.593, such that two classifiers are significantly different when their rank difference is at least 1.593. Therefore, we can conclude that FESN is significantly better than 1NN-ED and SpADe and slightly better than 1NN-DTW, Swale and 2DD$_{LCSS}$. The rank difference between GeTeM and FESN is 0.406; consequently, these two methods exhibit comparable performance.

5.1.2. FESN vs. feature-based methods

Table 2 shows the classification error rates of FESN and the feature-based methods, including TSBF-Unif, TSF Entrance, Feature-based linear, and Feature-DTW-DTW-R, on 19 UCR sets.

Compared with Feature-linear, FESN performs better on 15 of the 19 datasets, ties on 0 and performs worse on 4. Compared with TSBF-Unif, FESN performs better on 11 of the 19 datasets and worse on 8. Over all 32 datasets, FESN performs comparably to Feature-DTW-DTW-R, whose win/loss/tie numbers are 8/9/2, respectively. Similar to the distance-based methods, the Beef and Olive Oil tasks are also difficult for the feature-based methods. FESN has the highest average rank of 2.526, and Feature-linear has the lowest average rank of 4.132. As the results of Nemenyi tests in Fig. 6(a) show, the difference between FESN and Feature-linear is statistically significant with a critical difference of 1.399, and FESN achieves slightly better performance than TSBF-Unif, TSF Entrance and Feature-DTW-DTW-R.

We compared FESN with shapelet transform methods independently. Table 3 provides the error rates of FESN and 4 classifiers constructed on the shapelet transform on 17 UCR sets. FESN achieves the highest average rank of 1.471 and the best performance in 14 out of 17 TSC tasks in Table 3, showing that it performs better than the other classifiers constructed on the shapelet transform. For example, FESN is better on 15 of the 17 datasets, equal on 1 and worse on 1 when compared with the shapelet transform with random forest. As shown in Fig. 6(b), FESN is significantly better than Shapelet Tree, Naive Bayes and Random Forest at a 0.05 significance level and performs slightly better than shapelet SVM. These results indicate that, with the orthogonal decomposition (e.g., Fast Fourier Transform) in the high-dimensional functional spaces,

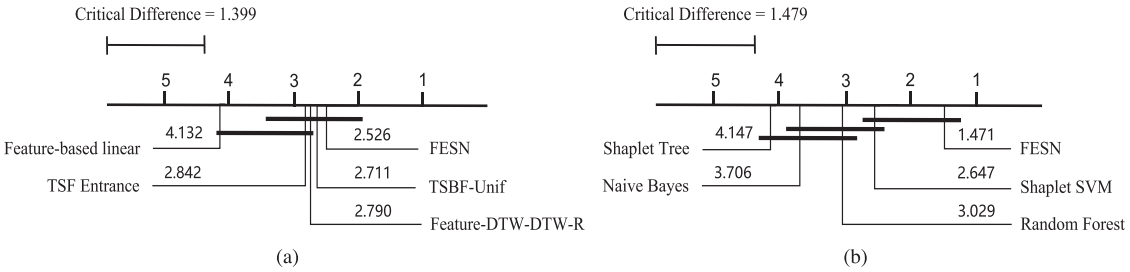

Fig. 6. Comparison of FESN with feature-based classifiers on the Nemenyi test. Groups of classifiers that are not significantly different (at p = 0.05) are connected. (a) shows the comparison of FESN with TSBF-Unif, TSF Entrance, Feature-linear, Feature-DTW-DTW-R. (b) shows the comparison of FESN with classifiers constructed on the shapelet transform: Shapelet Tree, Naive Bayes, Random Forest and Shapelet SVM.

Table 3
Error rates of FESN and 4 classifiers constructed on the shapelet transform on 17 UCR datasets.

Data Set	♯ of classes	♯ train	♯ test	length	Shapelet Tree	Naive Bayes	Random Forest	Shaplet SVM	FESN
Adiac	37	390	391	176	0.708	0.719	0.696	0.762	**0.384**
Beef	5	30	30	470	0.500	0.267	0.400	0.133	**0.000**
ChlorineConcentration	3	467	3840	166	0.412	0.540	0.424	0.439	**0.124**
Coffee	2	28	28	286	0.036	0.071	**0.000**	**0.000**	**0.000**
DiatomSizeReduction	4	16	306	345	0.278	0.212	0.196	0.078	**0.020**
ECGFiveDays	2	23	861	136	0.225	0.036	0.067	**0.011**	**0.019**
Face(four)	4	24	88	350	0.159	**0.023**	0.125	**0.023**	0.034
GunPoint	2	50	150	150	0.107	0.080	0.040	**0.000**	0.013
ItalyPowerDemand	2	67	1029	24	0.108	0.075	0.070	0.079	**0.028**
Lighting7	7	70	73	319	0.507	0.425	0.356	0.301	**0.288**
MedicalImages	10	381	760	99	0.512	0.826	0.492	0.475	**0.343**
MoteStrain	2	20	1252	84	0.175	0.112	0.154	0.113	**0.066**
SonyAIBORobotSurface	2	20	601	70	0.155	0.210	0.148	0.133	**0.118**
SyntheticControl	2	50	150	150	0.057	0.220	0.110	0.127	**0.023**
Trace	4	100	100	275	0.020	0.020	0.020	0.020	**0.000**
TwoLeadECG	2	23	1139	82	0.149	0.009	0.039	**0.007**	0.077
ElectricDevices	7	8296	7711	96	0.451	0.746	0.440	0.758	**0.416**
Win/Lose/Tie					0/17/0	2/15/0	1/15/1	4/12/1	–
Average rank					4.147	3.706	3.029	2.647	1.471
Rank difference					2.677	2.235	1.559	1.177	–
♯ of best errors					0	1	1	5	14

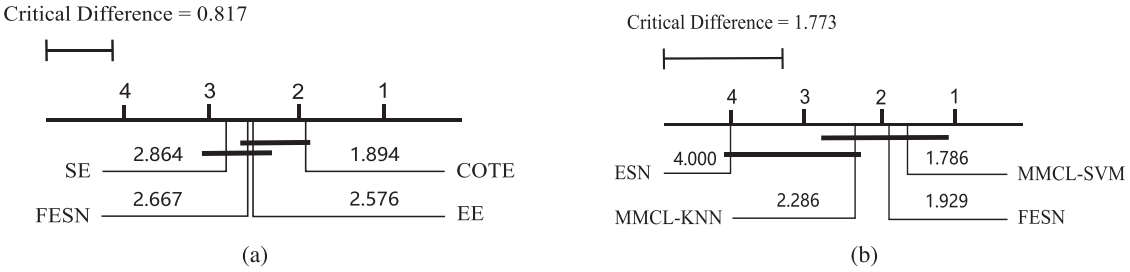

Fig. 7. (a) Nemenyi test comparisons of FESN with 3 ensemble-based classifiers. (b) Nemenyi test comparisons of FESN with 3 ESN-based classifiers. Groups of classifiers that are not significantly different (at p = 0.05) are connected.

FESN can extract more representative features than the classifiers constructed on the shapelet transform. We will illustrate some interpretable features extracted by FESN in the case study in subsection 5.4.

5.1.3. FESN vs. ensemble-based methods

The comparison results of FESN and three ensemble-based methods on 33 UCR tasks are shown in Table 4.

As shown in Table 4, we found that, compared with the ensemble-based methods (SE, EE and COTE), FESN achieves the best performance on 12 of the 33 UCR datasets. COTE won on 13 datasets, while SE and EE's performances won on 4 and 8 of 33 tasks, respectively. COTE has the best overall performance. The average rank of FESN is 2.667, lower than SE but higher than COTE and EE. To further analyze the performance, we used Binomial tests are to measure the significance of the differences between FESN and the three ensemble-based methods. Fig. 7(a) shows the critical difference diagram.

Table 4
Error rates of 3 ensemble-based methods and FESN on 33 UCR datasets.

Data Set	♯ of classes	♯ train	♯ test	length	SE	EE	COTE	FESN
50words	50	450	455	270	0.281	**0.180**	0.191	0.376
Adiac	37	390	391	176	0.435	0.353	**0.233**	0.384
Beef	5	30	30	470	0.167	0.367	0.133	**0.000**
CBF	3	30	900	128	0.003	0.002	0.001	**0.000**
ChlorineConcentration	3	467	3840	166	0.300	0.360	0.314	**0.124**
CinC_ECG_torso	4	40	1380	1639	0.154	**0.062**	0.064	0.652
Coffee	2	28	28	286	**0.000**	**0.000**	**0.000**	**0.000**
DiatomsizeReduction	4	16	306	345	0.124	0.059	0.082	**0.020**
ECGFiveDays	2	23	861	136	0.001	0.178	**0.000**	0.019
FaceAll	14	560	1690	131	0.263	0.152	0.105	**0.099**
FaceFour	4	24	88	350	0.057	0.091	0.091	**0.034**
FacesUCR	14	200	2050	131	0.087	0.063	**0.057**	0.099
Fish	7	175	175	463	**0.023**	0.034	0.029	0.160
GunPoint	2	50	150	150	0.020	**0.007**	**0.007**	0.013
Haptics	5	155	308	1092	0.523	0.584	**0.481**	0.558
ItalyPowerDemand	2	67	1029	24	0.048	0.039	0.036	**0.028**
Lighting2	2	60	61	637	0.344	**0.115**	0.164	0.148
Lighting7	7	70	73	319	0.260	**0.233**	0.247	0.288
MALLAT	8	55	2345	1024	0.060	0.050	**0.036**	0.296
MedicalImages	10	381	760	99	0.396	**0.245**	0.258	0.343
MoteStrain	2	20	1252	84	0.109	0.114	0.085	**0.066**
OliveOil	4	30	30	570	0.100	0.133	0.100	**0.000**
OSULeaf	6	200	242	427	0.285	0.194	**0.145**	0.517
SonyAIBORobotSurface	2	20	601	70	**0.067**	0.293	0.146	0.118
SonyAIBORobotSurfaceII	2	27	953	65	0.115	0.124	0.076	**0.072**
StarLightCurves	3	1000	8236	1024	**0.024**	0.079	0.031	0.160
SwedishLeaf	15	500	625	128	0.093	0.085	**0.046**	0.093
Syntheticcontrol	6	300	300	60	0.017	0.010	**0.000**	0.023
Trace	4	100	100	275	0.020	0.010	0.010	**0.000**
TwoLeadECG	2	23	1139	82	0.004	**0.000**	0.015	0.077
TwoPatterns	4	1000	4000	128	0.059	0.067	**0.000**	0.247
wafer	2	1000	6174	152	0.002	0.003	**0.001**	0.010
yoga	2	300	3000	426	0.195	0.121	**0.113**	0.198
Win/Lose/Tie					16/15/2	18/14/1	19/13/1	–
Average rank					2.864	2.576	1.894	2.667
Rank difference					0.197	−0.091	−0.773	–
♯ of best errors					4	8	13	12

As the results of the Nemenyi test in Fig. 7 show, bold lines indicate groups of methods that are not significantly different. The critical difference is 0.817, which means the differences between FESN, COTE and EE are not significant. COTE is slightly better than FESN. FESN has nearly the same performance as EE and has slightly better performance than SE. SE, EE and COTE use collections of many different classifiers (both distance-based and feature-based) that improve their accuracies on the TSC tasks (such as image outline classification tasks: Adiac, OSULeaf, yoga, etc) that either the distance-based methods or feature-based methods are good at. For the sensor reading classification tasks (including Beef, Coffee, and Chlorine Concentration), FESN has a clear advantage. For example, the error rate of FESN on Beef is 0, but COTE is 0.133.

However, although ensemble-based methods can achieve higher accuracies than many single-algorithm methods, they suffer from high complexity and will become computationally inefficient when faced with the big data volumes common in real-world applications. In contrast, although it is a single-algorithm method, FESN has competitive accuracy and high computational capabilities inherited from ESN, as discussed in Theorem 1.

5.1.4. FESN vs. ESN-based methods

In the comparison with the ESN-based methods, we focus on one ESN and two MMCL variants, MMCL-kNN and MMCL-SVM, which are proposed in [11]. Table 5 shows their classification errors on 7 UCR datasets.

In Table 5, the error rates of MMCL-kNN and MMCL-SVM are the best results reported in [11]. FESN has the lowest error rates on 4 of the 7 datasets (boldface figures), while MMCL-kNN and MMCL-SVM have the lowest error rates on 2 of the 7 datasets. ESN has no best results on any of the 7 datasets. MMCL-SVM has an average rank of 1.786, which is slightly better than FESN's average rank of 1.929. MMCL-kNN ranks third, with 2.286, and ESN has the lowest average rank, 4.0. From the Nemenyi test results shown in Fig. 7(b), we found a critical difference of 1.773. According to this critical difference, the rank difference between FESN and MMCL-SVM is -0.143; in other word, there is no significant difference between them.

However, MMCL-SVM is a complex model with adaptive input weights and recurrent weights that must be combined with an SVM for final classification. From this perspective, by inheriting the fixed-weights reservoir and end-to-end training with linear regression from ESN, FESN is more efficient than MMCL-SVM.

Table 5
Error rates of FESN and 3 ESN-based predictive classifiers on 7 UCR sets.

Data Set	♯ of classes	♯ train	♯ test	length	ESN	MMCL-kNN	MMCL-SVM	FESN
OSULeaf	6	200	242	427	0.682	**0.120**	0.149	0.517
OliveOil	4	30	30	570	0.600	0.133	0.067	**0.000**
Lightning2	2	60	61	637	0.344	0.295	0.246	**0.148**
Beef	5	30	30	470	0.400	0.367	0.173	**0.000**
Fish	7	175	175	463	0.629	**0.114**	0.126	0.160
Coffee	2	28	28	286	0.321	0.107	**0.000**	**0.000**
Adiac	37	390	391	176	0.798	0.277	**0.266**	0.384
Win/Lose/Tie					0/7/0	3/4/0	3/3/1	–
Average Rank					4.000	2.286	1.786	1.929
Rank difference					2.071	0.357	−0.143	–
♯ of best errors					0	2	2	4

Table 6
Error rates of FESN with different orthogonal bases.

Data Set	Walsh	Haar	Daubechies	Coiflets	Symlets	Fourier
Lighting2	0.230	0.262	0.230	0.213	0.246	**0.148**
Lighting7	0.370	0.295	0.279	**0.262**	0.295	0.288
ItalyPowerDemand	0.036	0.033	0.039	0.084	0.040	**0.028**
SonyAIBORobotSurface	0.118	**0.205**	0.213	0.208	0.233	0.210
SwedishLeaf	0.131	0.131	0.133	0.122	0.141	**0.093**
FacesUCR	**0.079**	0.117	0.119	0.105	0.115	0.099
GunPoint	0.020	0.020	0.020	0.027	0.020	**0.013**
OSULeaf	0.533	0.537	0.533	**0.512**	0.517	0.517
Win/Lose/Tie	2/6/0	1/7/0	1/7/0	3/5/0	0/7/1	–
Average Rank	3.250	4.063	4.188	3.000	4.563	1.938
Rank difference	1.313	2.125	2.250	1.063	2.625	–
♯ of best errors	1	1	0	2	0	4

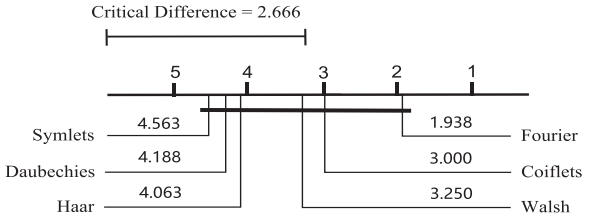

Fig. 8. Comparison of FESN with different orthogonal bases with Nemenyi tests. Groups of classifiers that are not significantly different (at p = 0.05) are connected.

5.2. Choosing an orthogonal basis

In this section, we analyze the effects of different orthogonal bases for FESN. Popular orthogonal bases include the Fourier orthogonal basis, Walsh functions basis, and various orthogonal wavelets including haar, daubechies, coiflets and symlets. The fast Fourier transform (FFT) is a method for rapidly decomposing a function into its frequency representations, but it ignores temporal information. In contrast, the wavelet transform (WT) is localized in both temporal and spatial spaces, but it is more complex than FFT. In the experiments, we fixed the wavelet transform level at 2 and fixed the length of the Fourier expansion to 120.

We selected 8 datasets from UCR, four of which (Lighting2, Lighting7, Italy Power Demand, and Sony AIBO Robot Surface) are from sensor reading classification problems, three (Swedish Leaf, Faces UCR and Adaic) belong to image outline classification tasks, and the remaining dataset, Gun Point, is a Motion Classification task. The classification results of the 6 orthogonal bases on these 8 UCR datasets are shown in Table 6.

As shown in Table 6, FESN with Fourier basis has the best performance, with an average rank of 1.938. The second-best is Coiflets, with 3.000, and the third-best is the Walsh function basis with 3.250. In the Nemenyi test results in Fig. 8, the critical difference is 2.666, which means there are no significant differences among the Fourier, Walsh function and other wavelet bases.

In addition to evaluating the accuracy of classification, we also conduct experiments to evaluate computational complexity of 6 different orthogonal bases. The time consumption results of FESN when learning with the 6 different orthogonal bases are shown in Fig. 9.

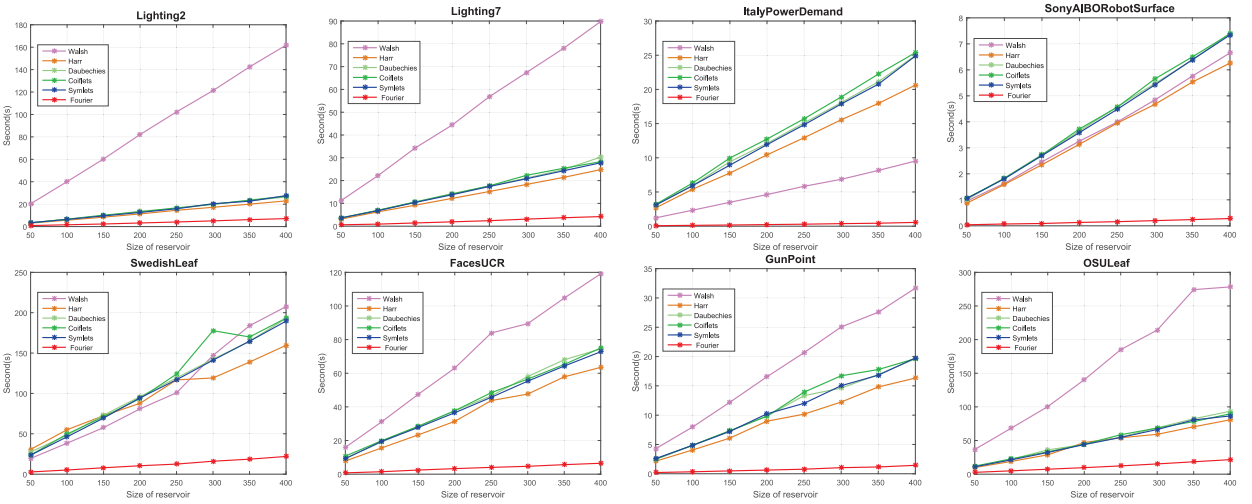

Fig. 9. Time complexity analysis of FESN with different orthogonal bases.

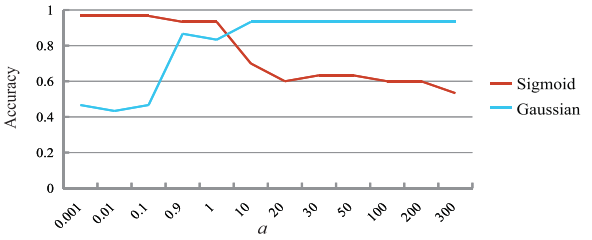

Fig. 10. Accuracy of FESN with different activation functions when varying the activation parameter, a.

The horizontal axes represent different reservoir sizes in FESN. As shown in Fig. 9, we found that the Fourier basis (red line) had the fastest computation time among the 6 orthogonal bases, while the Walsh function basis had the slowest. The differences among the wavelet orthogonal bases are not significant when the data set is small.

Briefly, considering both the classification accuracy and the learning speed, the Fourier basis is the best choice for FESN.

5.3. Effects of parameter values

Some parameters influence the performance of FESN, including the reservoir size N, the reservoir sparsity α, the reservoir spectra radius ρ, the length of the orthogonal basis expansion R, the reservoir unit activation function types (e.g., *Sigmoid* or *Gaussian*) and their parameters. We conducted experiments on the Beef dataset to illustrate the effects of these parameter values on FESN.

First, we consider the effects of the two activation functions, the *sigmoid* and *Gaussian* functions (given by Eqs. 32 and 33). The parameter is denoted as the symbol a.

$$Sigmoid(x) = \frac{1}{1+e^{-ax}} \tag{32}$$

$$Gaussian(x) = exp\left(-\left(\frac{x}{a}\right)^2\right) \tag{33}$$

For each activation function, *Sigmoid* and *Gaussian*, we chose $a \in \{0.001, 0.1, 0.9, 1, 10, 20, 30, 50, 100, 200, 300\}$ with a fixed N of 200, α of 0.01, ρ of 0.9 and R of 300. The accuracies of FESN with different activation functions when varying a are shown in Fig. 10. When $a \in [0.001, 1]$, the performance of the *Sigmoid* function is much better than that of the *Gaussian* function. In contrast, when a is in the interval from 10 to 300, the *Gaussian* function has higher accuracy than the *Sigmoid* function. When a is less than 1 in the *Sigmoid* function, FESN has higher accuracies with a slight sensitivity to a. In the Beef experiments, we choose *Sigmoid* as the activation function of FESN.

Second, after setting *Sigmoid* as the activation function, we illustrate the accuracies versus a and other parameters in Fig. 11.

On one hand, Fig. 11(a), (b) and (c) clearly show that when the value of a is in the range [0.001, 1], FESN achieves nearly the same high accuracy regardless of the values of N, α and ρ. In these results, a influences the performance of FESN in the

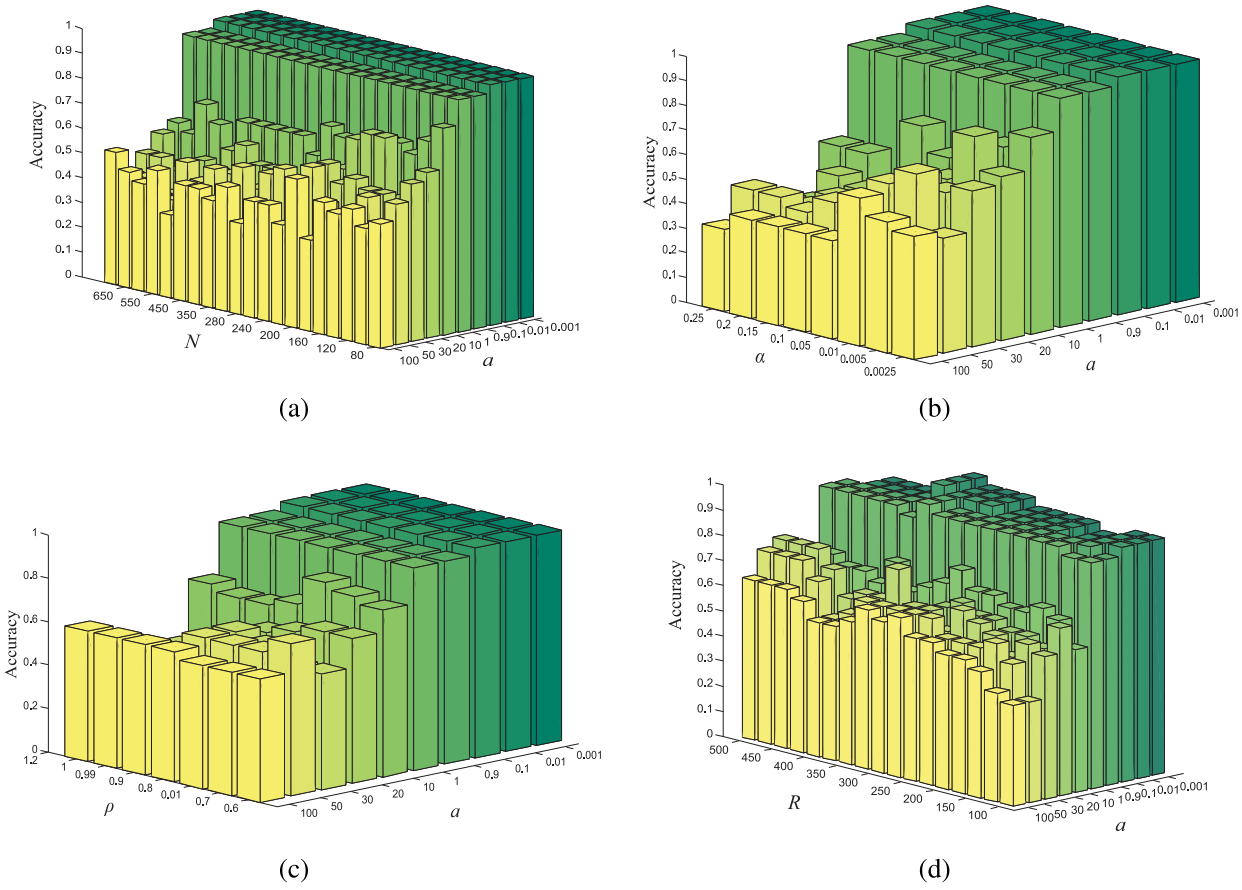

Fig. 11. Accuracies of FESN versus N, α, ρ, and R with variation of a respectively. (a) The accuracies of FESN versus N and a (α is fixed as 0.01, ρ as 0.9, R as 300); (b) The accuracies of FESN versus α and a (N is fixed as 200, ρ as 0.9, R as 300); (c) The accuracies of FESN versus ρ and a (N is fixed as 200, α as 0.01, R as 300); (d) The accuracies of FESN versus R and a (N is fixed as 200, α as 0.01, ρ as 0.9).

three cases. On the other hand, although Fig. 11(d) shows similar accuracy trends as 11(a), (b) and (c), a proper R value can improve the highest accuracy to 100% when R is in the range [300, 400].

On the whole, two arguments, the activation parameter in the *Sigmoid* function, a, and the length of the orthogonal expansion, R, have greater effects than others. These results are consistent with those in Fig. 10. In fact, our experiments obtained similar observations on other UCR datasets. We attribute this to FESN's ability to project data and extract temporal features. In other words, a nonlinear transformation by a reservoir with an appropriate activation function can enhance the separability of the input data in a high-dimensional functional space. In this space, better approximation of the output-weight functions using orthogonal basis expansion allows FESN to extract more discriminating features (e.g., frequency features by the Fourier transform) to find the optimal functional hyperplane $\mathbf{W}(t)$.

5.4. Exploratory visualization analysis of features based on a case study

To explore the interpretability of FESN, we studied the output-weight functions obtained by FESN on the Beef data set. Beef is a food spectrogram dataset used to classify food for food safety and quality assurance [2]. There are five time series categories in Beef and each class represents a different degree of contamination with offal. The spectrogram of Beef is shown in Fig. 12(a). At first glance, all five categories except class 1 are difficult to distinguish. Fig. 12(b) shows enlargements of two categories, class 2 and class 5, that have similar shapes (e.g., the location of the hump).

As mentioned above, we can note from Eq. 8 that temporal aggregation is the integral of the product of the output-weight function $w_{ji}(t)$ and the reservoirs neural state function $x_i(t)$. That means that FESN considers the relative importance of subsequences $x_i(t)$ at different time steps. Because $x_i(t)$ is generated by the input signal $u(t)$ at time step t, FESN equivalently emphasizes those subsequences of $u(t)$ that are most important for classification. Under these circumstances, the output-weight functions can be regarded as the optimal functional hyperplanes to separate the different categories of a time series in a high-dimensional functional space.

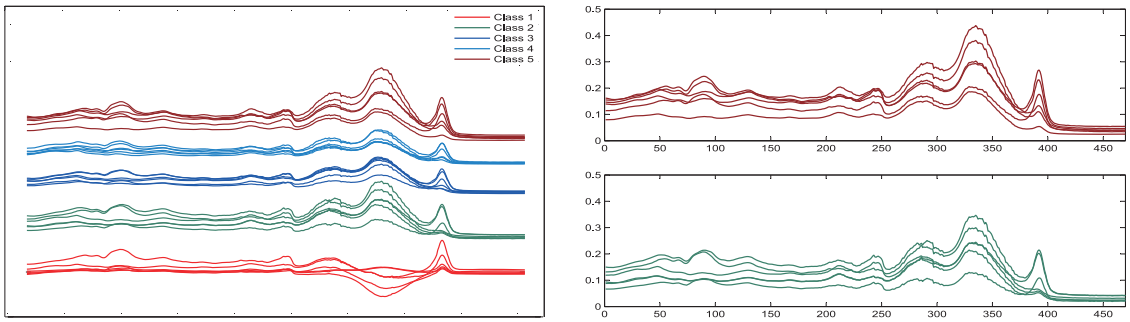

Fig. 12. (a) Spectrogram data of the five categories of Beef. (b) Enlargements of the spectrogram data of Class 2 and Class 5 in beef.

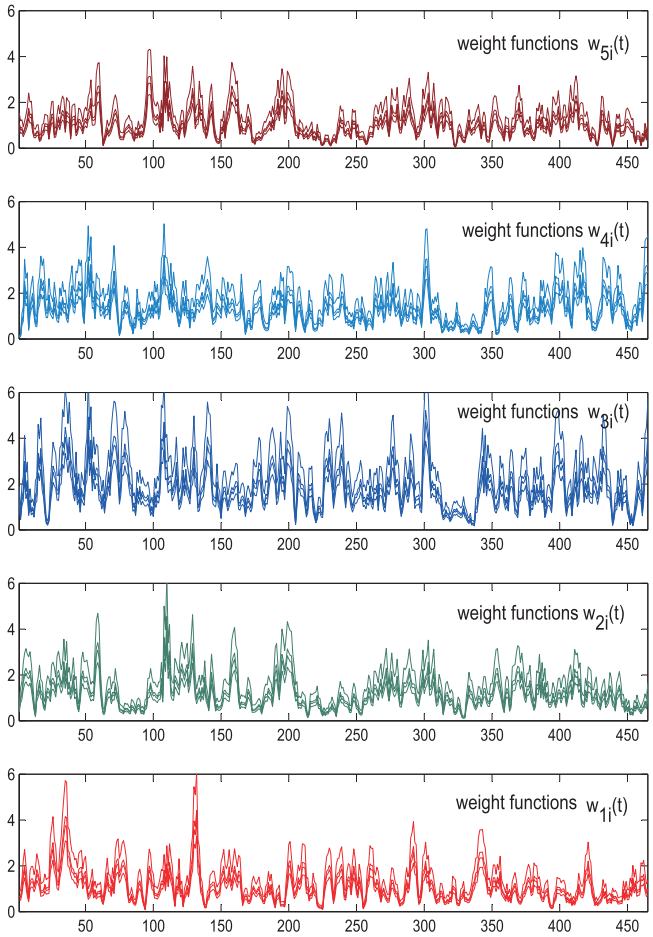

Fig. 13. Output-weight Functions from $w_{1i}(t)$ to $w_{5i}(t)$ $(i = 1, 2, \ldots, N)$ corresponding to five classes in Beef obtained by FESN.

Fig. 13 visualizes the optimal output-weight functions of the Beef task. As shown in Fig. 13, the output-weight functions corresponding to the same category have similar shapes, while those belonging to different categories have diverse shapes. Fig. 12(a) shows that class 2 and class 5 look very similar and that class 1 and class 2 have different shapes. To analyze which subsequences are important for classification in the two different cases, we compare them separately in Fig. 14(a) and (b).

From top to bottom, Fig. 14(a) shows the spectrogram of class 5, $w_{5i}(t)$, the bar of average differences between $w_{5i}(t)$ and $w_{2i}(t)$, the spectrogram of class 2, $w_{2i}(t)$, and the bar of average differences between $w_{2i}(t)$ and $w_{5i}(t)$, respectively. The bars show how large the differences of the output-weight function values are between the two classes. For example,

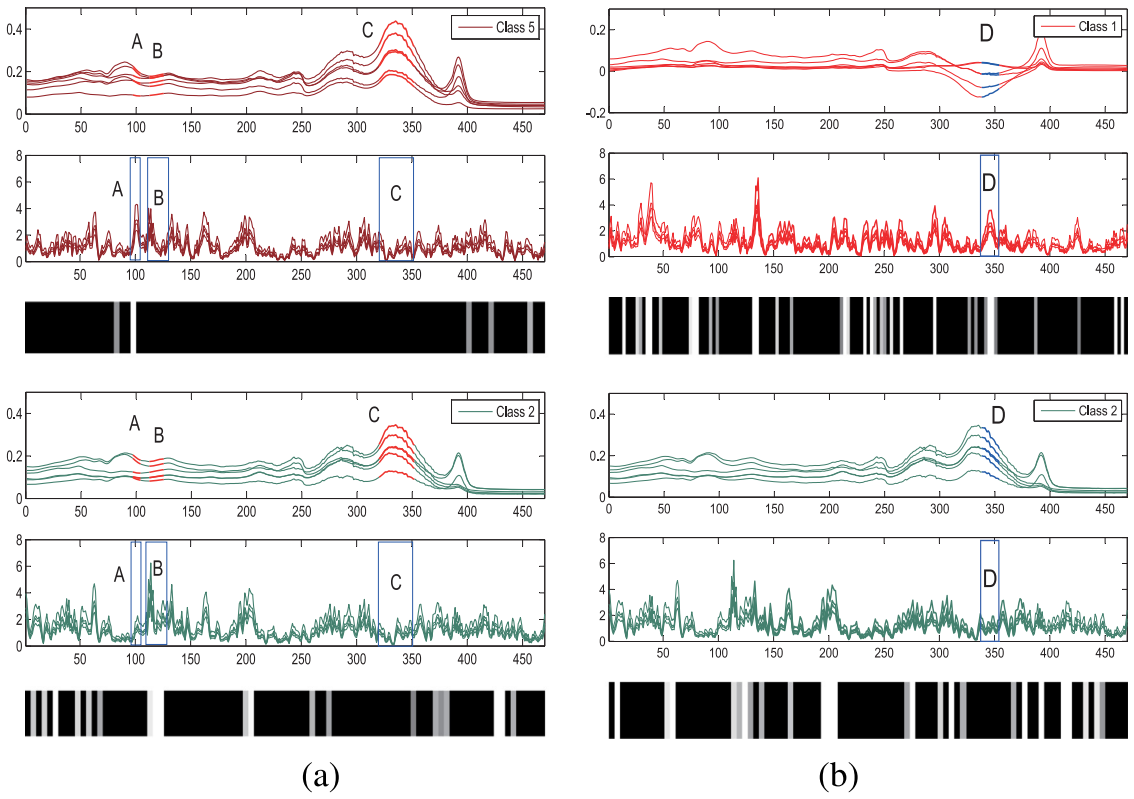

Fig. 14. Some subsequences extracted by FESN for discriminating class 1, class 2 and class 5 in Beef. (a) From top to bottom, there are the spectrogram of class 5, $w_{5i}(t)$, the bar of average differences between $w_{5i}(t)$ and $w_{2i}(t)$, the spectrogram of class 2, $w_{2i}(t)$, and the bar of average differences between $w_{2i}(t)$ and $w_{5i}(t)$, respectively. (b) is plotted in the same manner for class 1 and class 2.

in the upper bar, if the average difference between $w_{5i}(t)$ and $w_{2i}(t)$ is larger than a threshold at time step t, the color is white; otherwise, the color is black. In contrast, in the bottom bar, if the average difference between $w_{2i}(t)$ and $w_{5i}(t)$ is larger than a threshold at time step t, the color is white; otherwise, the color is black. Where two bars are all black at the same time step t, their output-weight functions have very close values for the two classes. Fig. 14(b) is plotted in the same manner for class 1 and class 2.

We highlight four positions, A, B, C and D, in Fig. 14. At position A, the color is white in the upper bar and black in the bottom bar. This means that the value of the output-weight function at A is much larger than the value at B. Therefore, subsequence A in class 5 is a more important feature than the corresponding subsequence in class 2 for separating these two classes. At position B, the color is black in the upper bar and white in the bottom bar, which means that subsequence B in class 2 is a more important feature than the corresponding subsequence in class 5 for separating these two classes. Meanwhile, for position C, the colors are black in both bars; thus, the convex subsequence has little effect on the classification of the two classes. However, to classify class 1 and class 2, the subsequence D in Fig. 14(b) at the same position as subsequence C is important because the color is black vs. white in the two bars. In other words, subsequences A, B and D are the critical features extracted by FESN to discriminate classes 1, 2 and 5, while subsequence C is not important when classifying classes 2 and 5. Therefore, FESN enhances the subsequences that are important features for classification with the weighting functions.

6. Conclusion

A novel approach to ESN named functional echo state network is presented for time series classification. By introducing a temporal aggregation operator into the reservoir units and replacing the numerical variables of the output weights in ESN with time-varying functions, FESN can map the reservoir states from the functional field into the real number field, which transforms it into a true classifier. To cope with the learning problem of time-varying output-weight functions, a spatio-temporal aggregation learning algorithm is proposed to approximate the output-weight functions with orthogonal function basis expansion, which can determine the optimal discriminating weight function in high-dimensional functional spaces. Based on theoretical analyses, FESN has an acceptable computational complexity. An empirical study on UCR datasets demonstrated that the proposed method can yields comparable or superior performance to existing distance-based, feature-based, ensemble-based and ESN-based approaches.

The most relevant advantage of FESN is that it provides an ESN-based classifier for time series classification, reaping the benefits from the mapping function of the reservoir and the accumulation of temporal information by the temporal aggregation operator. Compared with previous methods that predefine distances or design features from original time series data, FESN extracts features and measures distances between reservoir state functions $x(t)$ in a high-dimensional functional space, which can enhance the separability between different classes. Furthermore, as we evaluated in this paper, the interpretation features for time series are expressed in terms of specific patterns within the output-weight functions in FESN. Future work should investigate the big data classification in real-world applications.

Acknowledgements

The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant No. 61502174, 61402181), the Natural Science Foundation of Guangdong Province (Grant No. S2012010009961, 2015A030313215), the Science and Technology Planning Project of Guangdong Province (Grant No. 2016A040403046), the Science and Technology Programme of Guangzhou Municipal Government (Grant No. 2014J4100006), and the Fundamental Research Funds for the Central Universities (Grant No. D2153950).

References

[1] A. Bagnall, A. Bostrom, J. Large, J. Lines, The great time series classification bake off: An experimental evaluation of recently proposed algorithms. extended version (2016) arXiv:1602.01711.
[2] A. Bagnall, L.M. Davis, J. Hills, J. Lines, Transformation based ensembles for time series classification., in: SDM, 12, SIAM, 2012, pp. 307–318.
[3] A. Bagnall, J. Lines, J. Hills, A. Bostrom, Time-series classification with cote: the collective of transformation-based ensembles, IEEE Trans. Knowl. Data Eng. 27 (9) (2015) 2522–2535.
[4] G.E. Batista, X. Wang, E.J. Keogh, A complexity-invariant distance measure for time series., in: SDM, 11, SIAM, 2011, pp. 699–710.
[5] M.G. Baydogan, G. Runger, E. Tuv, A bag-of-features framework to classify time series, IEEE Trans. Pattern Anal. Mach. Intell. 35 (11) (2013) 2796–2802.
[6] D.J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time series., in: KDD workshop, 10, Seattle, WA, 1994, pp. 359–370.
[7] L. Büsing, B. Schrauwen, R. Legenstein, Connectivity, dynamics, and memory in reservoir computing with binary and analog neurons, Neural comput. 22 (5) (2010) 1272–1311.
[8] K.-P. Chan, A.W.-C. Fu, Efficient time series matching by wavelets, in: Data Engineering, 1999. Proceedings., 15th International Conference on, IEEE, 1999, pp. 126–133.
[9] W.A. Chaovalitwongse, R.S. Pottenger, S. Wang, Y.-J. Fan, L.D. Iasemidis, Pattern-and network-based classification techniques for multichannel medical data signals to improve brain diagnosis, IEEE Trans. Syst. Man Cybern. Part A 41 (5) (2011) 977–988.
[10] S.P. Chatzis, Y. Demiris, Echo state gaussian process, IEEE Trans. Neural Netw. 22 (9) (2011) 1435–1445.
[11] H. Chen, F. Tang, P. Tino, A.G. Cohn, X. Yao, Model metric co-learning for time series classification, in: Proceedings of the 24th International Conference on Artificial Intelligence, AAAI Press, 2015, pp. 3387–3394.
[12] L. Chen, R. Ng, On the marriage of lp-norms and edit distance, in: Proceedings of the Thirtieth international conference on Very large data bases-Volume 30, VLDB Endowment, 2004, pp. 792–803.
[13] L. Chen, M.T. Özsu, V. Oria, Robust and fast similarity search for moving object trajectories, in: Proceedings of the 2005 ACM SIGMOD international conference on Management of data, ACM, 2005, pp. 491–502.
[14] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, The ucr time series classification archive, 2015, http://www.cs.ucr.edu/~eamonn/time_series_data/.
[15] Y. Chen, M. Nascimento, B.C. Ooi, A.K. Tung, et al., Spade: On shape-based pattern detection in streaming time series, in: Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on, IEEE, 2007, pp. 786–795.
[16] Z. Chen, Z. Wangmeng, Q. Hu, L. Liang, Kernel sparse representation for time series classification, Inf. Sci. 292 (2015) 15–26.
[17] I.G. Costa, A. Schönhuth, C. Hafemeister, A. Schliep, Constrained mixture estimation for analysis and robust classification of clinical time series, Bioinformatics 25 (12) (2009) i6–i14.
[18] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE trans. inf. theory 13 (1) (1967) 21–27.
[19] A. Deihimi, H. Showkati, Application of echo state networks in short-term electric load forecasting, Energy 39 (1) (2012) 327–340.
[20] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[21] H. Deng, G. Runger, E. Tuv, M. Vladimir, A time series forest for classification and feature extraction, Inf. Sci. 239 (2013) 142–153.
[22] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, E. Keogh, Querying and mining of time series data: experimental comparison of representations and distance measures, Proc. VLDB Endowment 1 (2) (2008) 1542–1552.
[23] A. Douzal-Chouakria, C. Amblard, Classification trees for time series, Pattern Recog. 45 (3) (2012) 1076–1091.
[24] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast subsequence matching in time-series databases, 23, ACM, 1994.
[25] O. Fink, E. Zio, U. Weidmann, Quantifying the reliability of fault classifiers, Inf. Sci. 266 (2014) 65–74.
[26] J. Frank, S. Mannor, J. Pineau, D. Precup, Time series analysis using geometric template matching, IEEE Trans. Pattern Anal. Mach. Intell. 35 (3) (2013) 740–754.
[27] E. Frentzos, K. Gratsias, Y. Theodoridis, Index-based most similar trajectory search, in: Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on, IEEE, 2007, pp. 816–825.
[28] T. Fu, C. Law, K. Chan, F. Chung, C. Ng, Stock time series categorization and clustering via sb-tree optimization, in: Fuzzy Systems and Knowledge Discovery, Springer, 2006, pp. 1130–1139.
[29] B.D. Fulcher, N.S. Jones, Highly comparative feature-based time-series classification, IEEE Trans. Knowl. Data Eng. 26 (12) (2014) 3026–3037.
[30] E.S. García-Treviño, J.A. Barria, Structural generative descriptions for time series classification, IEEE trans. cybern. 44 (10) (2014) 1978–1991.
[31] P. Geurts, Pattern extraction for time series classification, in: Principles of Data Mining and Knowledge Discovery, Springer, 2001, pp. 115–127.
[32] D. Gordon, D. Hendler, L. Rokach, Fast and space-efficient shapelets-based time-series classification, Intell. Data Anal. 19 (5) (2015) 953–981.
[33] T. Górecki, Using derivatives in a longest common subsequence dissimilarity measure for time series classification, Pattern Recog. Lett. 45 (2014) 99–105.
[34] T. Górecki, M. Łuczak, Non-isometric transforms in time series classification using dtw, Knowledge-Based Syst. 61 (2014) 98–108.
[35] X. He, S. Xu, Process neural networks: theory and applications, Springer Science & Business Media, 2010.
[36] H. Jaeger, Controlling recurrent neural networks by conceptors (2014) arXiv:1403.3369.
[37] H. Jaeger, H. Haas, Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication, Science 304 (5667) (2004) 78–80.
[38] Y.-S. Jeong, M.K. Jeong, O.A. Omitaomu, Weighted dynamic time warping for time series classification, Pattern Recog. 44 (9) (2011) 2231–2240.
[39] R.J. Kate, Using dynamic time warping distances as features for improved time series classification, Data Mining Knowl. Discovery (2015) 1–30.
[40] H. Kaya, Ş. Gündüz-Öğüdücü, A distance based time series classification framework, Inf. Syst. 51 (2015) 27–42.

[41] R. Legenstein, W. Maass, Edge of chaos and prediction of computational performance for neural circuit models, Neural Netw. 20 (3) (2007) 323–334.
[42] D. Li, M. Han, J. Wang, Chaotic time series prediction based on a novel robust echo state network, IEEE Trans. Neural Netw. Learn. Syst. 23 (5) (2012) 787–799.
[43] J. Lines, A. Bagnall, Time series classification with ensembles of elastic distance measures, Data Mining Knowl. Discovery 29 (3) (2015) 565–592.
[44] J. Lines, L.M. Davis, J. Hills, A. Bagnall, A shapelet transform for time series classification, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2012, pp. 289–297.
[45] Y. Liu, Q. Liu, W. Wang, J. Zhao, H. Leung, Data-driven based model for flow prediction of steam system in steel industry, Inf. sci. 193 (2012) 104–114.
[46] M. LukošEvičIus, H. Jaeger, Reservoir computing approaches to recurrent neural network training, Comput. Sci. Rev. 3 (3) (2009) 127–149.
[47] S. Lun, X. Yao, H. Hu, A new echo state network with variable memory length, Inf. Sci. 370 (2016) 103–119.
[48] Q.-L. Ma, W.-B. Chen, Modular state space of echo state network, Neurocomputing 122 (2013) 406–417.
[49] M.D. Morse, J.M. Patel, An efficient and accurate method for evaluating time series similarity, in: Proceedings of the 2007 ACM SIGMOD international conference on Management of data, ACM, 2007, pp. 569–580.
[50] K. Noponen, J. Kortelainen, T. Seppänen, Invariant trajectory classification of dynamical systems with a case study on ecg, Pattern Recog. 42 (9) (2009) 1832–1844.
[51] C. Orsenigo, C. Vercellis, Combining discrete svm and fixed cardinality warping distances for multivariate time series classification, Pattern Recog. 43 (11) (2010) 3787–3794.
[52] A. Rodan, P. Tiňo, Simple deterministically constructed cycle reservoirs with regular jumps, Neural comput. 24 (7) (2012) 1822–1852.
[53] M.D. Skowronski, J.G. Harris, Minimum mean squared error time series classification using an echo state network prediction model, in: Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on, IEEE, 2006, pp. 3153–3156.
[54] M.D. Skowronski, J.G. Harris, Automatic speech recognition using a predictive echo state network classifier, Neural netw. 20 (3) (2007) 414–423.
[55] D. Verstraeten, B. Schrauwen, M. dHaene, D. Stroobandt, An experimental unification of reservoir computing methods, Neural Netw. 20 (3) (2007) 391–403.
[56] M. Vlachos, G. Kollios, D. Gunopulos, Discovering similar multidimensional trajectories, in: Data Engineering, 2002. Proceedings. 18th International Conference on, IEEE, 2002, pp. 673–684.
[57] L. Wang, Z. Wang, S. Liu, An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm, Expert Syst. Appl. 43 (2016) 237–249.
[58] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, E. Keogh, Experimental comparison of representation methods and distance measures for time series data, Data Mining Knowl. Discovery 26 (2) (2013) 275–309.
[59] L. Ye, E. Keogh, Time series shapelets: a new primitive for data mining, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 947–956.
[60] D. Yu, X. Yu, Q. Hu, J. Liu, A. Wu, Dynamic time warping constraint learning for large margin nearest neighbor classification, Inf. Sci. 181 (13) (2011) 2787–2796.
[61] M.-H. Yusoff, J. Chrol-Cannon, Y. Jin, Modeling neural plasticity in echo state networks for classification and regression, Inf. Sci. 364 (2016) 184–196.
[62] Y. Zheng, Q. Liu, E. Chen, J.L. Zhao, L. He, G. Lv, Convolutional nonlinear neighbourhood components analysis for time series classification, in: Advances in Knowledge Discovery and Data Mining, Springer, 2015, pp. 534–546.
[63] L. Zhu, C. Lu, Y. Sun, Time series shapelet classification based online short-term voltage stability assessment, IEEE Trans. Power Syst. 31 (2) (2016) 1430–1439.