Multiobjective Learning in the Model Space for Time Series Classification

Zhichen Gong, *Student Member, IEEE*, Huanhuan Chen, *Senior Member, IEEE*, Bo Yuan, *Member, IEEE*, and Xin Yao, *Fellow, IEEE*

***Abstract*—A well-defined distance is critical for the performance of time series classification. Existing distance measurements can be categorized into two branches. One is to utilize handmade features for calculating distance, e.g., dynamic time warping, which is limited to exploiting the dynamic information of time series. The other methods make use of the dynamic information by approximating the time series with a generative model, e.g., Fisher kernel. However, previous distance measurements for time series seldom exploit the label information, which is helpful for classification by distance metric learning. In order to attain the benefits of the dynamic information of time series and the label information simultaneously, this paper proposes a multiobjective learning algorithm for both time series approximation and classification, termed multiobjective model-metric (MOMM) learning. In MOMM, a recurrent network is exploited as the temporal filter, based on which, a generative model is learned for each time series as a representation of that series. The models span a non-Euclidean space, where the label information is utilized to learn the distance metric. The distance between time series is then calculated as the model distance weighted by the learned metric. The network size is also optimized to learn parsimonious representations. MOMM simultaneously optimizes the data representation, the time series model separation, and the network size. The experiments show that MOMM achieves not only superior overall performance on uni/multivariate time series classification but also promising time series prediction performance.**

***Index Terms*—Echo state network (ESN), learning in the model space, multiobjective learning, time series classification.**

I. Introduction

TIME series are ubiquitously created and exploited in scientific [1], engineering [2], [3], medicine [4], entertainment [5], stock market [6], etc. The pervasiveness of time series inspires machine learning techniques for time series analysis. One of the key issues is to define the distance between time series.

Euclidean distance (ED) compares time steps one-by-one and is most suitable for equal length time series. When time series have variable length, dynamic time warping (DTW) [7] is usually employed to align the series to the same length by allowing many-to-one or one-to-many comparisons of time steps. Although ED and DTW have demonstrated their usefulness in measuring the distance of time series on many tasks [8], [9], previous researchers argued that they are not suitable when dealing with time series that are high-dimensional, long, or noisy [10]. Besides, time series usually have long or short temporal dependencies, which characterize the dynamics of time series. However, ED and DTW essentially ignore the dynamic information [10].

In order to employ the dynamic information, some attempts have been made to utilize a generative model to approximate the time series. Fisher kernel [11], [12] utilizes the gradients of the log likelihood of a generative model as a representation of the time series. Fisher kernel assumes that time series in the same class make the parameters of a generative model change in a similar manner. However, the separation capacity (SC)[1] of the gradients is not clear. Reservoir kernel (RV) [2], [13] employs the echo state network (ESN) to learn a model for each time series. It demonstrates that the final connection weights of the ESN network are able to predict the class membership. Similar to Fisher kernel, the SC of the learned models is ignored.

Time series usually have labels. The label information is critical for classification tasks. Yet, ED and DTW [7], etc. are unsupervised distance measurements, while Fisher kernel [11], RV [13], etc. are not intuitive to employ the supervised similarity constraint, which prevents them from finding discriminative features for classification. Luckily, there have been some strategies to take advantage of the label information in discriminative learning, such as distance metric learning [14].

In this paper, our goal is to employ the dynamic information of time series to learn faithful models and employ the label information to enhance the classification performance

Manuscript received September 30, 2017; revised December 8, 2017; accepted December 29, 2017. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1000905, in part by the National Natural Science Foundation of China under Grant 61673363, Grant 91546116, Grant 61329302, and Grant 61503357, and in part by the Science and Technology Innovation Committee Foundation of Shenzhen under Grant ZDSYS201703031748284 and Grant JCYJ20170307105521943. This paper was recommended by Associate Editor H. Wang. *(Corresponding author: Huanhuan Chen.)*

Z. Gong and H. Chen are with the UBRI Laboratory, School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China (e-mail: zcgong@mail.ustc.edu.cn; hchen@ustc.edu.cn).

B. Yuan and X. Yao are with the Shenzhen Key Laboratory of Computational Intelligence, Department of Computer Science and Engineering, Southern University of Science and Technology, Shenzhen 518055, China (e-mail: yuanb@sustc.edu.cn; xiny@sustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2018.2789422

[1]For the moment, it is enough to consider the representation capacity (RC) as the capacity of a generative model to approximate time series, and the SC as the relative distances of the learned models (representations) of the time series in different classes. Please see Section III-C for more information.

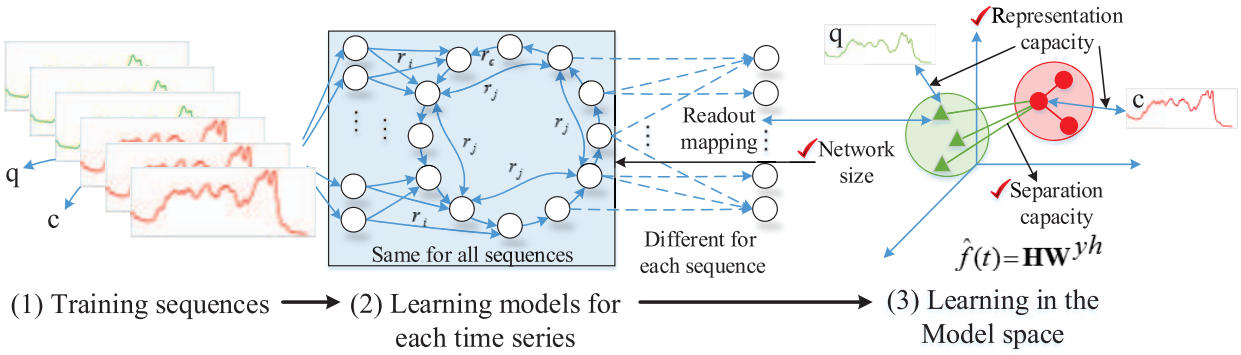


Fig. 1. Illustration of the proposed MOMM method. Given a set of time series (step 1), we first learn a model for each time series by using the ESN (step 2). The models span a space, which we term model space. Then learning algorithms are performed in the model space. To learn models, our goal is to optimize three objectives, i.e., RC of learned models, SC of learned models, and the network size (step 3). We formulate the problem as a multiobjective learning problem and implement an MOEA for optimizing the ESN.

in the model-spanned space. Model learning and class separation learning focus on different goals. Generally speaking, the objective constructed just for model learning concentrates more on the sequential dynamics and less on the classification performance. In contrast, the objective constructed just for class separation focuses more on the classification performance and less on finding the dynamic information in data. Hence, the RC and the SC might lead to conflicting results. In addition, to exploit the temporal dynamic information, this paper exploits a type of recurrent neural network. A large network may lead to additional computational cost, unwanted chaotic behaviors and overfit [6], [15]. A small network may be unable to uncover the dynamics in the data. The topology optimization usually requires intensive trial-and-error procedure [6] of a human expert. It is desired to automatically determine the network topology to make the network compact.

Due to the conflict of the objectives, a single solution that is optimal for all objectives simultaneously does not exist [6], [16], [17]. Therefore, there has to be a tradeoff among the objectives in order for good performance. Multiobjective evolutionary algorithms (MOEAs) provide a general and flexible framework for multiobjective learning problems. Based on population, it returns a number of solutions in a single run. Each solution is a tradeoff between objectives [18]. Therefore, we reformulate the learning problem of the model space into a Pareto-based multiobjective learning problem to simultaneously optimize the three objectives. The potential advantages of using MOEA include: it enables us to analyze the interactions among different objectives; and it is convenient to add or remove an objective without adjusting the algorithm.

In order to attain the benefits of data approximation and model separation simultaneously and reducing human efforts in determining a suitable network size, this paper proposes a multiobjective learning algorithm, multiobjective model-metric (MOMM) learning, to simultaneously optimize the RC, SC of series models, and the network size. The label information is utilized to learn a distance metric (a positive semidefinite matrix) to enhance the SC in learning models. In particular, we propose to represent time series that are uni-dimensional/multidimensional or possibly varying length, by learning models for them. The time series models are aware of the dynamical behaviors of the series. The models span a space, termed model space. The distance of the models is calculated to measure the distance of time series. The learning algorithms are performed in the model space. A lot of studies [1], [14], [19] have demonstrated that learning distance metric is helpful to enhance the discriminative capacity of classifiers. Motivated by these results, we learn a distance metric in the model space to facilitate the SC. In doing so, time series in the same class are represented by similar models while time series in different classes are represented by dissimilar models. Fig. 1 illustrates the key idea of the MOMM.

The MOMM has several advantages.

1) The network structure and connection weights are tuned automatically and effectively. It avoids tedious cross-validation (CV) or local optima when training the recurrent network.
2) A set of tradeoffs among different objectives can be found by using the MOEA. The objectives are effectively optimized without explicit specification of the combinational coefficients.
3) The RC, SC, and the network structure are optimized simultaneously. This simultaneous learning enables the solutions to attain the advantages of each objective and to be even better than optimizing the objectives individually.
4) We are able to obtain suitable solutions for both series representation and classification problem in one run and the obtained networks are usually of small size, which is desired for efficiency and hardware implementation.

The main contributions of this paper include the following.

1) We propose a supervised multiobjective learning algorithm MOMM to simultaneously optimize time series approximation, time series model separation and automatically accomplish network topology selection (Section III).
2) We empirically reveal the effect of the objectives on time series classification (Section IV-B) and the benefits

of including the network size as the objective to be optimized (Section IV-C).

3) By optimizing the three objectives with MOEA, we achieve superior overall classification accuracy on univariate and multivariate time series (Section IV-D).

The rest of this paper is organized as follows. Section II gives a brief review of the state-of-the-art time series classification methods and background of multiobjective learning. Section III details the MOMM approach. Section IV demonstrates the effectiveness of the MOMM and provides empirical analysis. Section V concludes this paper.

II. Background and Related Work

In this section, we briefly introduce related work on time series classification. We consider time series $\mathbf{X} = [\mathbf{X}(1); \mathbf{X}(2); \ldots; \mathbf{X}(L_{\mathbf{X}})]^T \in R^{L_{\mathbf{X}} \times d}$, where $\mathbf{X}(i) \in R^{d \times 1}$ is the observation for d variables and $L_{\mathbf{X}}$ is the length of time series $\mathbf{X}$. Each time series has a label y. Denote a training set containing M samples as $\mathbf{T} = \{\mathbf{X}^m, y^m\}_{m=1}^M$. Time series classification is to predict y given a new sample $\mathbf{X}$.

The previous time series classification methods could be predominantly categorized into the signal-based and model-based approaches.

A. Signal-Based Methods

The signal-based methods operate on raw data directly and often use handmade features to define the distance. DTW [7], [20] allows nonlinear local warping to align two time series temporally to the same length such that the accumulative distance is minimized. However, due to the nonlinear local warping, the DTW distance violates the triangle inequality. Thus it is not a properly defined distance measurement [21]. Another strategy is to symbolize time series by discretizing real values into symbols [22], which, however, causes a loss of information. Time series classification by bag-of-feature (TSBF) [10] samples subsequences from original time series as features. It is favored for flexibility and insensitiveness to nonlinear local warping. However, the classification performance is dependent on advanced classification techniques. These time series distance or representations are strong solutions for processing time series. Yet, they have limitations in leveraging the generating mechanism and the label information of time series.

B. Model-Based Methods

The model-based methods address the problem with signal-based methods by using a probabilistic generative model to approximate the time series, such as Kullback–Leibler divergence-based kernel [23], probability product kernel [24], global alignment kernel (a generalization of DTW by taking all candidate aligns into consideration) [21], Fisher kernel [11], RV [13], etc.

Fisher kernel [11], [12] usually employs a hidden Markov model (HMM) to approximate a set of series. It transforms the series into Fisher scores, which are defined as the gradient of the log-likelihood function $\nabla_\theta P(S|\theta)$ of the HMM. Fisher kernel calculates the distance between two time series $\mathbf{X^1}$ and $\mathbf{X^2}$ by $\nabla_\theta P(\mathbf{X^1}|\theta) \mathscr{I}^{-1} \nabla_\theta P(\mathbf{X^2}|\theta)$, where $\mathscr{I}$ is Fisher information metric. The inverse of the metric tensor results in intensive computational cost. To avoid this effort, the Fisher Information metric is usually ignored and the Fisher scores are simply projected into a Euclidean space, with a loss of valuable information [25]. Fisher kernel is trained by maximizing the likelihood. It may have problems when the likelihood reaches a local maximum, where the gradients are nearly zeros.

Chen *et al.* [13] proposed to use RV to learn models for time series. The distance between time series is defined by the function distance between models. However, the SC of the models is ignored and may result in inferior classification performance.

Chen *et al.* [19] proposed a model metric co-learning (MMCL) algorithm. MMCL combines the learning of models and distance metric into a single objective by a tradeoff. Iterative gradient descent are employed to optimize the network weights and the distance metric alternatively. However, due to the recurrent training of the network, it may suffer from gradient vanishing issues and be trapped in a local optimum.

Previous studies have problems in properly taking into consideration the label information. The time series models are learned either by tedious CV-based grid-search [2], [13], which has the disadvantage that it needs user interaction and selects parameters from discrete values of the parameter space; or learn models and distance metric by combining them with a coefficient and are optimized alternatively by gradient-based optimization [19], which may fall into local optima and limit the ability.

Our MOMM is in line with RV and MMCL in learning models for time series but they are essentially different. In MOMM, the RC, SC of time series models, and the network size are concurrently optimized by an MOEA. The objectives of MOMM are generalizations of that in RV and MMCL, and they are well-defined for optimization. Besides, the MOEA enables us to gain more insights into these objectives. By taking advantage of the flexible weighting coefficients and avoiding local optima of training recurrent networks, the MOMM often shows better performance than RV and MMCL.

C. Multiobjective Learning

Multiobjective learning is to find solutions by optimizing the several objectives $\{f_1(x), f_2(x), \ldots, f_N(x)\}$ simultaneously [26]. The objectives are often conflicting with each other. To compare solutions with multiple objectives, the dominance relationship is usually employed [5].

Definition 1: Define domination as (Assuming Minimization):

1) A solution x_1 dominates x_2 iff $\forall i \in \{1, 2, \ldots, N\}$, $f_i(x_1) \leq f_i(x_2)$, denoted by $x_1 \preceq x_2$.
2) When $x_1 \preceq x_2$ and $\exists j \in \{1, 2, \ldots, N\}, f_j(x_1) < f_j(x_2)$, x_1 strictly dominates x_2, denoted by $x_1 \prec x_2$.

A solution that is not dominated by any other solutions is a Pareto optimal solution. The goal of multiobjective learning is to approximate the set of Pareto optimal solutions, which is also called the Pareto front.

Esling and Agon [5] proposed to retrieve target time series by jointly minimizing a set of distances, which are calculated by multiple descriptors. To classify a time series, the predicted label is assigned based on the voting or dominance volume of the Pareto front. Krause *et al.* [27] proposed to find a small and efficient network for a pattern generator by treating the approximation accuracy and network size as two conflicting objectives. The network connectivity and weights are optimized by MOEA and a genetic algorithm, respectively. Delgado *et al.* [6] treated the time series prediction accuracy, network size and number of hidden activations as three objectives to evolve recurrent networks for prediction problem. Despite the potential overlap of the last two objectives, promising prediction accuracy is achieved.

III. MOMM for Time Series Classification

This section first introduces the ESN. The model for each time series is then introduced. Next, the optimization objectives for the ESN are explicated. Finally, the MOMM algorithm is presented.

A. Echo State Networks

To learn a parsimonious model for each time series, ESN is employed as the state space model in this paper. ESN [28] is a discrete time recurrent network. It receives inputs step-by-step and converts each input into a latent state by combining the input and the previous state. Due to this recursive nature, the ESN is able to maintain the input history information with a fading memory.

The ESN is composed of an input layer, a reservoir network, and a readout layer. The reservoir is a high-dimensional sparse network, where the neurons are connected recurrently. In a typical ESN setting, the recurrent network structure and connection weights are randomly fixed subject to echo state property, i.e., the largest eigenvalue of the reservoir weight matrix $||\rho|| < 1$ [29]. Loosely speaking, the echo state property is to make far earlier inputs have less influence on the current state. The readout layer is the only trainable part. It is usually trained through linear regression [30] by mapping from state space to the target output.

In order to avoid the randomness in designing a proper reservoir, a deterministically constructed derivative of ESN is proposed recently [31], called cycle reservoir with jumps (CRJs). Fig. 2 illustrates the architecture of the CRJ network. The N reservoir neurons of CRJ are cyclically interconnected in uni-direction. The cyclic connection weights share the same value r_c. Neurons at a distance J are bi-directionally connected, where the jump connection weights share the same value r_j. The inputs are fully connected to the reservoir. The input weights share an absolute value r_i, and the sign flips randomly [31]. This paper employs this deterministic version of ESN as the base model to ease the operations and analysis.

The reservoir state and the readout mapping of CRJ can be generalized as

$$\begin{cases} \mathbf{h}(t) = f\left(\mathbf{W}^{hh}\mathbf{h}(t-1) + \mathbf{W}^{hx}\mathbf{X}(t)\right) \\ \hat{f}(t) = \mathbf{W}^{yh}\mathbf{h}(t) \end{cases} \tag{1}$$

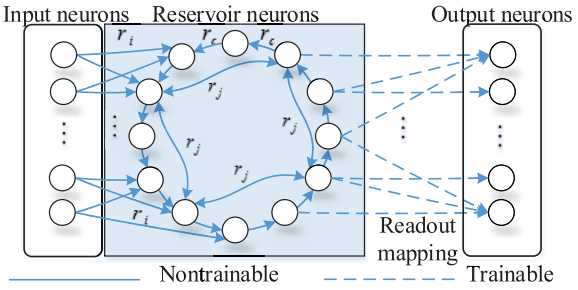


Fig. 2. Illustration for CRJ reservoir network.

where $\mathbf{X}(t) \in R^{d+1}$ is the input stream with an additional unit bias, d is the dimension of the input sequences; $\mathbf{h}(t) \in R^N$ represents the hidden state, N is the number of reservoir neurons; $\mathbf{W}^{hx} \in R^{N\times(d+1)}$ is the input weight matrix, $\mathbf{W}^{hh} \in R^{N\times N}$ is the reservoir weight matrix, $\mathbf{W}^{yh} \in R^{d'\times N}$ is the output weight matrix to be learned, d' is the dimension of output sequences; and f is the nonlinear state transition function, which is implemented with sigmoid function or tan$h$.

The input layer and the reservoir are the same for all the time series in order to provide a unique platform for providing versatile dynamic features. The output mapping assembles the state space features to learn a model for the specific input sequence. The nonlinear approximation capacity of CRJ is able to approximate the time series well, providing the input and the reservoir are designed properly. The work in this paper is to optimize the ESN in learning models for time series.

B. Models of Time Series

The model for a time series is trained by future prediction [6], which has been widely applied in language processing and video learning. In this paper, the CRJ network is employed to extrapolate one-step-ahead future observation according to the past input history. A time series is then represented by the linear readout function, i.e., (1).

Specifically, the state of the CRJ reservoir is initially set as all zeros, i.e., storing no characteristics about the data. To fill the gap of cold start, a time series is divided into two parts $\mathbf{X}(0 \sim L_0 - 1)$ and $\mathbf{X}(L_0 \sim L)$. The first part $\mathbf{X}(0 \sim L_0-1)$ is employed to wash out the stochastic initial transition. The second part $\mathbf{X}_{L_0\sim L}$ is employed for training the readout mapping. Let $\mathbf{Y}$ denote the target output sequence. For one-step prediction, $\mathbf{Y} = \mathbf{X}(L_0 + 1 \sim L) \in R^{(L-L_0-1)\times d}$. Denote $\mathbf{H} \in R^{(L-L_0-1)\times N}$ be the collections of the reservoir states over time points $\mathbf{X}(L_0 \sim L-1)$, i.e., $\mathbf{h}(L_0) \sim \mathbf{h}(L-1)$. Then the readout mapping is trained by

$$\mathbf{W}^{yh} = \arg\min_{\mathbf{W}^{yh}} \sum_{t=L_0}^{L-1} \left\| \hat{f}(t) - \mathbf{X}(t+1) \right\|^2 + \lambda ||\mathbf{W}^{yh}||^2.$$

By virtue of ridge regression, $\mathbf{W}^{yh}$ can be calculated in a closed-form

$$\mathbf{W}^{yh} = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}\right)^{-1}\mathbf{H}^T\mathbf{Y}. \tag{2}$$

Here λ needs to be adjusted for generalization. However, the learned models do not exploit the label information and cannot guarantee to be suitable for classification.

C. Objectives of MOMM

The goals are to learn the time series models that are faithful for future prediction and with good class distribution for classification, by a CRJ which is as compact as possible. Due to the highly constrained CRJ structure, the connection weights (r_i, r_c, r_j), the network topology (N, J) and the regularization parameter of ridge regression (λ) fully determine the performance [31]. Denote the optimizing parameters collectively as $\theta = \{r_i, r_c, r_j, J, N, \lambda\}$.

1) Representation Capacity: Given a time series $\mathbf{X}$, the time series model is trained by one-step-ahead future prediction (as stated in Section III-B). Despite the assembling ability of the readout layer, our goal is to optimize the general-purpose reservoir so that it memorizes the input history information and approximates the dynamics of the time series effectively. In particular, the reservoir network is optimized so that the likelihood of time series $\mathbf{X}$ is maximized given the initial value $\mathbf{X}(1)$ and the learned model $\hat{f}_\theta(t)$

$$\begin{aligned} &P\Big(\mathbf{X}(2), \mathbf{X}(3), \ldots, \mathbf{X}(L) | \hat{f}_\theta(t)\Big) \\ &\quad = P\Big(\mathbf{X}(L) | \mathbf{X}(L-1), \ldots, \mathbf{X}(2), \mathbf{X}(1), \hat{f}_\theta(t)\Big) \\ &\qquad \times P\Big(\mathbf{X}(3) | \mathbf{X}(2), \mathbf{X}(1), \hat{f}_\theta(t)\Big) P\Big(\mathbf{X}(2) | \mathbf{X}(1), \hat{f}_\theta(t)\Big) \end{aligned} \quad (3)$$

where $\hat{f}_\theta$ is the time series model trained with a CRJ network parameterized by θ and $P(\mathbf{X}(i)|\mathbf{X}(i-1), \ldots, \mathbf{X}(2), \mathbf{X}(1), \hat{f}_\theta)$ is the likelihood that $\hat{f}_\theta$ outputs $\mathbf{X}(i)$ given the previous inputs.

We employ normalized mean square error (NMSE) on the training set to evaluate the RC

$$\mathbf{RC}: \ \arg\min_\theta \left(\frac{1}{M} \sum_{m=1}^{M} \sum_{t=1}^{L_m - 1} \frac{\left\| \hat{f}_\theta^m(t) - \mathbf{X}^m(t+1) \right\|^2}{\mathrm{var}(\mathbf{X}^m)} \right) \quad (4)$$

where M is the number of training samples. var denotes the variance. The smaller the RC, the better the learned model approximates the original time series.

2) Separation Capacity: Let $\hat{f}^m(t)$[2] be the model for time series indexed by m. Then the distance of two time series q and c can be measured by the distance of their models [2]

$$\begin{aligned} D^2\big(f^q(t), f^c(t)\big) &= \left(\int_{\mathscr{H}} \big(f^q(t) - f^c(t)\big)^2 d\mu(\mathbf{h}) \right) \\ &= \left(\int_{\mathscr{H}} \left(\left\| \mathbf{W}_{qc}^{yh} \mathbf{h} \right\|^2 + 2\mathbf{b}_{qc}^T \mathbf{W}_{qc}^{yh} \mathbf{h} + \left\| \mathbf{b}_{qc} \right\|^2 \right) d\mu(\mathbf{h}) \right) \end{aligned}$$

where $\mathscr{H}$ is the integral range of the reservoir state. $\mu(\mathbf{h})$ is the probability distribution of the state $\mathbf{h}$. $\mathbf{W}_{qc}^{yh} = \mathbf{W}_q^{yh} - \mathbf{W}_c^{yh}$ and $\mathbf{b}_{qc} = \mathbf{b}_q - \mathbf{b}_c$.

We set the state transition function f as *tanh*, then the state range $\mathscr{H} \in [-1, 1]^N$. Assuming $\mu(\mathbf{h})$ is uniform distribution, the middle term in the second row can be omitted because for any fixed $\mathbf{b}$ and $\mathbf{W}$, $\int_{\mathscr{H}} \mathbf{b}^T \mathbf{W}^{yh} \mathbf{h} \, d\mu(\mathbf{h}) = 0$. Then the above equation can be rewritten as

$$D^2(f^q(t), f^c(t)) = \frac{2^N}{3} \sum_{j=1}^{N} \sum_{i=1}^{d'} w_{i,j}^2 + 2^N \|\mathbf{b}\|^2 \quad (5)$$

$$\propto \sum_{j=1}^{N} \sum_{i=1}^{d'} \frac{1}{3} w_{i,j}^2 + \|\mathbf{b}\|^2 \quad (6)$$

where $w_{i,j}$ is the (i,j)th element of $\mathbf{W}_{qc}^{yh}$, N is the number of reservoir neurons, and d' is the dimension of the output sequence. Since N is the same for all the time series, thus 2^N can be omitted. We rescale the original orientation to eliminate the weight $1/3$ in (5). Then we concatenate $\mathbf{W}^{yh}$ and $\mathbf{b}$ into a vector $\mathbf{w} \in R^{(N+1)d'}$. The distance between models is generalized as the ED[3] between the scaled model parameters $\mathbf{w}$. In the following, we denote the distance of two models $f^q(t)$ and $f^c(t)$ as $D(\mathbf{w_q}, \mathbf{w_c})$.

The ED is not adapted to the characteristics of data and cannot guarantee to be optimal. As a mitigation, we induce a Mahalanobis distance metric $\mathbf{Q}$ to let models in the same class have small distance while models in different classes have large distance. To learn the distance metric, we follow neighborhood component analysis (NCA) [14]. In principle, NCA learns a linear transformation of original space so that a stochastic variant of K-nearest neighbor classifier is optimized. The positive semidefinite matrix $\mathbf{Q} = \mathbf{A}^T\mathbf{A}$, where $\mathbf{A}$ is the learned transformation. We restrict $\mathbf{A}$ to be diagonal such that it measures the relative importance of different reservoir neurons

$$\begin{aligned} D^2\big(\mathbf{w}_q, \mathbf{w}_c\big)_{\mathbf{A}} &= \big(\mathbf{w}_q - \mathbf{w}_c\big)^T \mathbf{Q} \big(\mathbf{w}_q - \mathbf{w}_c\big) \\ &= \big(\mathbf{w}_q - \mathbf{w}_c\big)^T \mathbf{A}^T \mathbf{A} \big(\mathbf{w}_q - \mathbf{w}_c\big) \end{aligned}$$

where $\mathbf{w}_q$ and $\mathbf{w}_c$ are the scaled readout parameters. Let

$$P_{q,c|\mathbf{A},\theta} = \frac{e^{-D^2\left(\mathbf{w}_{q|\theta}, \mathbf{w}_{c|\theta}\right)_{\mathbf{A}}}}{\sum_{k=1, k\neq q}^{M} e^{-D^2\left(\mathbf{w}_{q|\theta}, \mathbf{w}_{k|\theta}\right)_{\mathbf{A}}}}, \ q \neq c$$

be a soft nearest neighbor selection rule. $P_{q|\mathbf{A},\theta} = \sum_{m|y^q = y^m} P_{q,m|\mathbf{A},\theta}$ measures the likelihood of the time series q being correctly classified in the transformed space.

The SC is to maximize the likelihood of all the training time series being correctly classified

$$\mathbf{SC}: \ \arg\min_{\boldsymbol{\theta}} -E(\boldsymbol{\theta}) = \arg\min_{\boldsymbol{\theta}} -\left(\sum_{m=1}^{M} P_{m|\mathbf{A},\boldsymbol{\theta}} \right). \quad (7)$$

The distance metric is learned as a subroutine in our MOEA algorithm. The projection mapping $\mathbf{A}$ is learned by maximizing the above equation through gradient descent [14]

$$\begin{aligned} \nabla_{\mathbf{A}} E(\theta) &= -2\mathbf{A} \sum_{m=1}^{M} \sum_{y^q = y^m} P_{m,q|\mathbf{A},\theta} \begin{pmatrix} \mathbf{w}_{mq}\mathbf{w}_{mq}^T - \\ \sum_{k=1}^{M} P_{m,k|\mathbf{A},\theta} \mathbf{w}_{mk}\mathbf{w}_{mk}^T \end{pmatrix} \\ &= 2\mathbf{A} \sum_{m=1}^{M} \begin{pmatrix} P_{m|\mathbf{A},\theta} \sum_{k=1}^{M} P_{m,k|\mathbf{A},\theta} \mathbf{w}_{mk}\mathbf{w}_{mk}^T \\ - \sum_{y^q = y^m} P_{m,q|\mathbf{A},\theta} \mathbf{w}_{mq}\mathbf{w}_{mq}^T \end{pmatrix} \end{aligned} \quad (8)$$

where $\mathbf{w}_{mq} = \mathbf{w}_m - \mathbf{w}_q$.

[2]We omit θ here for compactness.

[3]As stated, we have scaled the parameters. Strictly speaking, the distance between models puts less weight $(1/3)$ to the orientation than the offset, so it is different from ED.

3) Reservoir Size: The number of neurons in the reservoir determines the memory capacity and the dynamics the network could provide [29], [30]. A large reservoir may introduce unnecessary computational cost and unwanted chaotic behaviors. Besides, the large reservoir is hard to train and may easily overfit, resulting in deteriorated generalization performance. However, a small reservoir is unable to capture the dynamic characteristics of input sequences. RC and SC are network performance concerning different aspects, i.e., generating mechanism and pattern discrimination. Previous studies have demonstrated that the increase of reservoir size will improve RC [28], [31] and SC [17]. This implies a trade-off between the network performance, i.e., RC and SC, and the reservoir size. The determination of a suitable reservoir size usually needs the trial-and-error procedure, which requires a lot of training time and expert knowledge [30]. Luckily, the ability to take the network topology and complexity into consideration in a remarkable advantage for applying MOEA in neural networks [6], [15]. The network structure optimization has been widely performed in evolutionary neural networks and receives promising benefits [3], [6], [16]. It makes possible to evaluate networks with different structures inside the training process, without an extra loop to adapt the network topology.

In this paper, the minimization of the reservoir size is treated as an objective. A small network is preferred from the consideration of efficiency and hardware implementation

$$\mathbf{RS}: \min N. \tag{9}$$

The inclusion of the RS produces a search space which is different from using only RC and SC. Our experiments show that taking the reservoir size as the third objective could be beneficial to the overall performance, as well as reducing human interaction in determining a proper reservoir size.

D. Objective Relation

In a multiobjective problem, a unique solution that optimizes all objectives simultaneously does not exist. Otherwise, the problem would be solved by optimizing a single objective. In this paper, the RC and SC are dependent on the amount of memory capacity the reservoir could provide. Simultaneously optimizing RC/SC and RS requires the network to provide satisfactory performance with a minimum number of neurons.

We now analyze the relation between RC and SC. RC pursues approximation to the generating mechanism of the training sequences. SC pursues separation among different classes. We review the following considerations.

1) RC and SC could complement each other [32]. The RC result provides generating mechanism information of input sequences and may help improve the recognition performance. The SC exploits the sequence label information and provides the guidance for understanding the variation structures in data.
2) RC and SC could be conflicting. For example, considering a dataset where the examples in two classes are overlapped, if we only consider models that approximate the data well, the class distribution in the model space may not be separable either. Whereas, by taking the separation ability into consideration, the models can be learned to be separable according to the label information.

Therefore, it would be desired to simultaneously optimize the RC and SC by MOEA to attain the benefits of both domains.

Although the three objectives are different quantities and concern different aspects of desired solutions, they are all determined by the same set of CRJ network parameters. In this paper, we employ a nondominated sorting evolutionary algorithm to tune the network parameters.

TABLE I
INITIALIZATION RANGES FOR THE PARAMETERS OF THE CRJ NETWORK

$\{r_i, r_c, r_j\}$	J	N	λ
$[0.01, 1]$	$[2, \frac{N}{2}]$	$[10, 100]$	$\{10^x \mid x = -6, -5, ..., 2\}$

E. Multiobjective Learning of Time Series Model Space

In this paper, we consider a population of CRJ networks, which have three optimizing objectives. Initialization of the CRJ network parameters θ is set randomly with uniform distribution in the range given in Table I.

To optimize the reservoir network, we employ a multiobjective nondominated sorting [33] and local selection evolutionary algorithm. In particular, our method is based on NSGAII [26]. NSGAII algorithm sorts the population by their objective values and assigns fitness. To generate new solutions, parents are selected based on their fitness. Since the fitness range is broad and the selection is global, some solutions may be frequently selected. Therefore, these solutions reproduce themselves quickly. It results in the search being limited in an area that is guided by a subset of the population. In order to avoid this problem, in this paper, we propose to cluster the solutions in each front according to their fitness by k-means algorithm. The clustering algorithm employs ED. As the selection operator, we first select a cluster randomly, then select a solution in the cluster. In this way, we block out the influence of good solutions by the range of a cluster. Each cluster on the fronts has equal probability of generating offsprings. In doing so, we are able to generate solutions with more diversity. The parent solutions perform crossover and mutation to generate new solutions. Next, we describe the crossover and mutation operators in the MOMM.

1) Crossover: There are many crossover methods in the literature, for example one-point crossover, two-point crossover [26]. For CRJ models, we define two kinds of crossovers.

1) *Weight Crossover:* Take parameter r_i as an example. Given parents p and q, the crossover is defined as follows:

$$\begin{aligned} r_i^{\text{0ffs1}} &= r_i^p \times \gamma + r_i^q \times (1-\gamma) \\ r_i^{\text{0ffs2}} &= r_i^q \times \gamma + r_i^p \times (1-\gamma) \end{aligned} \tag{10}$$

where γ is the random coefficient and r_i^{0ffs1} and r_i^{0ffs2} are the offsprings of the parameter r_i.

2) *Structure Crossover:* It is also known as one-point crossover. Randomly select one cut point and swap the CRJ components after the cut point. The two crossover operations are performed randomly in implementation.

2) Mutation: A gene is first selected randomly, then a Gaussian noise $\mathcal{N}(0, \sigma)$ is added to the gene's original value

$$r_i^{\text{mutant}} = r_i + N\big(0, \sigma_{r_i}\big). \tag{11}$$

σ_{r_i} is the standard deviation of the Gaussian noise for parameter r_i. Similarly, the other genes are mutated in a similar way. Note that if the mutation results in $||\rho|| \geq 1$, then r_c and r_j have to be regenerated to guarantee echo state property (see Section III-A).

As the termination condition for the evolution, we set a threshold value for each objective. The threshold values for RC, SC, and RS are set as 10^{-3}, 10^{-3}, and 1^4 in this paper. This decision is made based on some preliminary experiments and not meant to be optimal. When the difference between the best fitness values of parent population and the offspring falls below the threshold, the evolution ends; otherwise, continue. The maximum iteration number is set as 200.

When the algorithm terminates, a final population has been generated. The first front contains nondominated solutions with different tradeoffs among objectives. We select an efficient solution in the nondominated front by tenfold CV to maximize the classification performance of an SVM classifier. More specifically, for solution selection, the training set is partitioned into ten folds randomly. The SVM classifier is tested on each fold after training on the remaining nine folds. The parameters of SVM are provided in the experiment. The final solution is selected as the solution with the least average error over ten folds. After obtaining the solution, the SVM is retrained on the training data before applying it to test data. We also consider some alternative solution selection strategies and compare them with the CV-based strategy in Section IV-E1.

The MOMM algorithm is summarized in Algorithm 1.

Algorithm 1 MOMM Learning

Initialization Let $i = 1$, create an initial population of CRJ reservoir networks randomly. The network parameters are uniformly generated (see Table I for the parameter range).

Objective Evaluation Train initial CRJ network population and then compute RC (Equation 4), SC (Equation 7) and RS (Equation 9) for each individual. The distance metric is learned using gradient descent as a plug-in procedure (Equation 8) when evaluating SC.

Non-dominated Ranking and clustering Sort the population by non-domination sorting and assign fitness. Then partition each front by the k-means clustering algorithm.

repeat

Crossover operations are repeated for desired times to generate a sub-population $P_i^{crossover}$. Two clusters are first selected randomly. Then parents are selected by binary tournament from the two clusters, respectively.

Mutation operations are repeated for desired times to generate a sub-population $P_i^{mutation}$. The parent is randomly generated by binary tournament selection.

Keep elites as $\forall q \in P_i^{crossover} \cup P_i^{mutation}$,if $\nexists x \in P_i, x \prec q$, then $P_i = (P_i - \{x' | q \prec x'\}) \cup \{q\}$, where $\prec$ is domination operator. Calculate fitness for P_i and keep the top N_{pop} solutions as new population P_{i+1}. Let $i = i + 1$.

until Iteration reaches the maximum or the termination condition is met.

Output Solution Select one solution in the final nondominated front by 10-fold cross-validation as the output.

IV. Experimental Studies

In this section, we evaluate the MOMM method in order to demonstrate the following.

1) The effect of different objectives for time series classification (Section IV-B).
2) The influence of the network size optimization and distance metric learning on the solution searching process (Section IV-C).
3) The effectiveness of MOMM on time series classification in comparison with the state-of-the-art (Section IV-D).
4) The effectiveness of solution selection from the nondominated front and the convergence (Section IV-E).

Before that, we first detail our experimental setup.

A. Experimental Setup

1) Compared Methods: The compared methods include DTW [7], TSBF [10], Fisher kernel[5] [12], RV [13], and MMCL [19]. We include HV-MOTS[6] and MESN [27] in order to compare MOMM with multiobjective methods on time series classification. To be fair, we also use the multiobjective learning algorithm to optimize RV for comparison, denoted by MRV. In MRV, we take the future prediction error and 1NN classification error of leave-one-out CV as two objectives to be minimized.

2) Parameter Setting: In our implementation, the global constraint of DTW is optimized using leave-one-out CV [7]. The minimum subsequence length and minimum interval size of TSBF are set following [10]. The number of hidden states of HMM in Fisher kernel [12], the network weights in RV [13], the tradeoff coefficient in MMCL [19], the ridge regression parameter λ, etc. are set by tenfold CV on the training set. For MESN, we set the population size as 200 and the rest parameters are the same as [27]. All points on the nondominated front are optimized to minimize the approximation error. We select one solution by tenfold CV that minimizes the CV error [27]. The search ranges for all these parameters are presented in Table II.

In MOMM, the population size is set as 200. The ratios for crossover and mutation are set as 0.8 and 0.2, respectively. The standard deviation of Gaussian noise in mutation is set as 20% of the range of the selected parameter (see Table I). The first 1/3 points of a sequence are employed as the wash-out part for the reservoir. We set $k = 5$ for k-means algorithm with

[4]The number of neurons in the reservoir is unchanged.

[5]http://homepage.tudelft.nl/19j49/fisher/Fisher_Kernel_Learning.html

[6]http://repmus.ircam.fr/esling/hvmots-datasets.html

TABLE II
PARAMETERS FOR COMPARED APPROACHES EMPLOYED IN THIS PAPER

Approaches	*Parameters*	*Parameter range*
TSBF	z, w_{min}	$z \in \{0.1, 0.25, 0.5, 0.75\}, w_{min} \in \{1, 2, \cdots, 10\}$
Fisher	$\#states$	$\#states \in \{1, 2, ..., 10\}$
RV	$r_i, r_c, r_j, \lambda, N$	$r_i, r_c, r_j \in \{0.01, 0.05, ..., 1\}, \lambda \in \{10^{-6}, 10^{-5}, ..., 10^{2}\}$, $N \in \{10, 20, \cdots, 100\}$
MMCL	α, λ, N	$\alpha \in \{0, 10^{-1}, ..., 10\}, \lambda \in \{10^{-6}, 10^{-5}, ..., 10^{2}\}$, $N \in \{10, 20, \cdots, 100\}$

z and w_{min} are parameters determining the length of the minimum subsequence and minimum interval. $\#states$ is the number of HMM states. r_i,r_c,r_j are the parameters for CRJ network, λ is the regularization parameter in ridge regression, α is the trade-off parameter for MMCL.

random seeds. These parameters are determined arbitrarily and may not be optimal for MOMM.

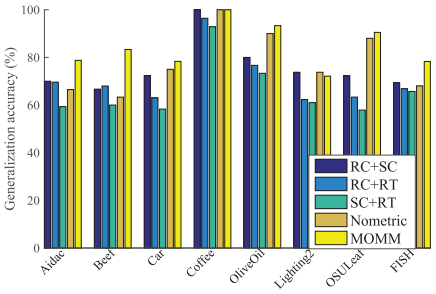
DTW employs 1NN classifier, which has been shown to be competitive for time series classification. Other compared methods employ SVM for classification. An extensively acknowledged implementation of SVM—LIBSVM is employed in the experiment [34]. We also perform tenfold CV to determine the slack-weight regularization parameter $C \in \{10^{-3}, 10^{-2}, \ldots, 10^{3}\}$ and kernel width $\gamma \in \{10^{-6}, 10^{-5}, \ldots, 10\}$ in SVM. For datasets that do not have enough training samples for tenfold CV, such as ECG5, we employ fivefold CV. As the default setup in LIBSVM, one-versus-one strategy is adopted in multiclass classification. After the model selection, the selected model is retrained on all training data before evaluated on the test data.

3) Datasets: The experiments are performed on both univariate and multivariate time series. We employ 23 univariate datasets from the UCR Time Series Repository [9]. The dataset information is presented in Table III. The datasets are already divided into standard training and test sets. These datasets include binary-class/multiclass classification, short-/long-term series, image outline classification and sensor reading classification tasks, etc. These characteristics motivate us to select these datasets to evaluate the proposed method.

We also evaluate the MOMM algorithm on three multivariate time series datasets from UCI Machine Learning Repository (http://archive.ics.uci.edu/ml), i.e., Brazilian sign language (Libras), handwritten characters (Hand), and Australian sign language signs (AUS). Note that Hand and AUS are also variable lengths. Libras dataset contains video-recorded 15 class of hand movement patterns. Each class has 24 instances. The centroid pixels of the hand are found in 45 evenly sampled frames and compose the movement curve with 45 points. The Hand dataset contains 3-D pen tip velocity trajectories. Each character is represented by 2-D coordinates and pen tip force. The AUS dataset collects 95 video-recorded Australian sign language signs from a native signer. There are 27 samples per sign. These datasets are divided into training and test set randomly by 70%/30%.

B. Effect of Objectives on Time Series Classification

To study the relative influence of different objectives on classification results, our strategy is to leave out one objective to study its effect. In particular, we consider five situations.



	RC	SC	RS
RC		8/8	8/8
SC			7/8
RS			

Fig. 3. Left plot illustrates the performance of different objective combinations. It demonstrates that two-objective optimization is less effective than three-objective optimization significantly. The right table presents the pairwise comparison of RC, SC, and RS regarding two-objective classification performance on eight univariate datasets. The numerical value in (i, j)th entry means the number of wins of i in comparison with j. The result indicates the relative effect of three objectives: RC$>$ SC $>$ RS.

1) *RC+SC:* Optimize two objectives RC and SC, no matter how large reservoir it needs.
2) *RC+RS:* Optimize two objectives RC and RS. The network is trained only for prediction and no label information is exploited.
3) *SC+RS:* Optimize two objectives SC and RS. Only the separation among sequence models is considered. Data approximation is not considered.
4) *No-Metric:* Optimize all three objectives RC, SC and RS. The distance metric $\mathbf{Q}$ is set as the identity matrix and is not optimized.
5) *MOMM:* Optimize three objectives RC, SC, and RS.

1) Compare Three-Objective Optimization With Any Two-Objective Optimization: Fig. 3 presents the average classification results on eight datasets. It is clear that optimizing two objectives usually yields poorer results than three objectives. Wilcoxon signed rank test is evaluated to see whether the pairwise difference between the performance of any two objectives and three objectives is significant. The p-values are 0.016/0.008/0.008/0.016, respectively, indicating the significant difference. The result demonstrates that optimizing the model space with three objectives indeed leads to significantly better performance than optimizing any two objectives.

2) Comparison of Two-Objective Optimization: First, we restrict our attention on the two-objective optimization cases, i.e., RC+SC, RC+RS, and SC+RS. In particular, RC+SC achieves the best performance on seven out of eight datasets. SC+RS maintains the worst results on all eight datasets. RC+RS achieves intermediate classification performance. It is slightly better on Beef dataset than RC+SC and SC+RS. We can make three main observations from Fig. 3.

1) Despite that the network is not constrained, it is observed that RC+SC still has quite good performance. We attribute this observation to the size range ([10, 100]) of the CRJ reservoir in the experiment. When the potential reservoir size is set even larger than that in our experiments, RS may become more critical to the overall performance. Because in that case, the reservoir size is prone to grow large and tends to over-fit [28].
2) RC+RS achieves better performance than SC+RS but worse than RC+SC. This result confirms that the

TABLE III
GENERALIZATION ACCURACY OF MOMM AND COMPARED METHODS ON UNIVARIATE TIME SERIES DATASETS

Dataset	#Class	#train	#test	Len	DTW	TSBF	MESN	Fisher	RV	MRV	MMCL	MOMM
50words	50	450	455	270	75.82	75.82	73.41	46.59	75.82	78.24	77.58	**79.34**
Aiac	37	390	391	176	60.89	75.51	65.24	68.03	72.63	70.08	73.40	**78.77**
Beef	4	30	30	577	66.67	71.33	77.45	58.00	66.67	80.00	83.33	**86.67**
Car	5	60	60	470	76.67	71.67	75.22	65.00	76.67	72.33	76.67	**78.33**
CBF	3	30	900	128	**99.56**	97.78	86.22	94.47	89.33	90.19	93.56	99.44
Coffee	2	28	28	286	**100.00**	96.43	89.29	81.43	**100.00**	**100.00**	**100.00**	**100.00**
DiatomSize	4	16	306	345	93.46	87.18	88.24	93.14	97.71	91.5	91.04	**98.69**
ECG	2	100	100	96	**88.00**	84.50	77.42	84.33	82.87	83.71	84.87	87.12
ECG5	2	23	861	136	79.67	81.79	79.54	85.37	65.51	82.11	86.76	**99.77**
FaceFour	4	24	88	350	**88.64**	94.32	77.27	**88.64**	85.23	73.87	87.5	86.59
FISH	7	175	175	463	84.60	**92.00**	71.43	77.14	78.29	70.28	87.43	89.71
GunPoint	2	50	150	150	91.33	98.67	89.52	95.33	90.67	91.33	96.67	**99.33**
Lighting2	2	60	61	637	86.85	74.34	86.89	84.10	87.05	74.59	85.41	**91.80**
Lighting7	7	70	73	319	71.23	**73.97**	69. 86	71.23	75.34	67.12	72.93	73.29
OliveOil	4	30	30	570	86.67	91.00	85.33	86.67	86.67	81.51	**93.33**	**93.33**
OSULeaf	6	200	242	427	61.17	67.12	60.33	54.96	69.83	72.33	85.12	**90.50**
SyntheticControl	6	300	300	60	98.30	**99.20**	89.67	85.34	79.67	90.34	89.11	98.00
Trace	4	100	100	275	99.00	98.00	89.00	**100**	**100**	**100**	98.94	**100**
CricketX	12	390	390	300	77.18	72.20	68.92	78.72	61.03	71.79	77.18	**78.97**
CricketY	12	390	390	300	76.15	73.59	65.13	**77.69**	71.79	74.35	71.47	75.89
CricketZ	12	390	390	300	74.62	71.54	67.77	68.85	63.08	71.03	**76.15**	74.11
SonyRobot	2	20	601	70	69.55	82.36	63.73	67.55	56.57	83.44	80.59	**96.51**
SonyRobotII	2	27	953	65	85.83	80.38	79.01	85.62	71.14	84.16	83.69	**89.72**
				Mean gap	6.79	5.97	11.83	10.86	10.62	8.42	4.14	

Comparison of MOMM with DTW, TSBF, MESN, Fisher kernel, RV, MRV and MMCL on univariate datasets, in which the best accuracy is boldfaced. MOMM achieves the best classification performance on 15/23 datasets. The mean performance gap for MOMM against the remaining methods is provided.

model training only by prediction is important to capture the generating mechanism of the sequence. Whereas, it may not lead to the best representations for classification.

3) SC+RS is not a good choice on all datasets. It pursues SC of models on the training set. However, without considering the dynamic information in the data, SC+RS generalizes poorly on the test set. Therefore, SC is an ancillary objective that helps improve the classification performance.

Let us take a step further. Actually, by comparing the classification performance of two-objective optimization, we obtain the pairwise comparison of three objectives. For example, when comparing the performance of RC+SC and RC+RS, we indeed compare SC and RS. The right table of Fig. 3 presents the pairwise comparison of three objectives. The result confirms the relative importance of RC, SC, and RS. First of all, RC consistently achieves better performance than SC and RS. This result reveals that RC is the primary objective and influences the classification results the most. As expected, without understanding the nature of data, optimizing SC and RS may not yield good generalization performance. Second, SC wins on seven out of eight datasets than RS. This result indicates that learning a distance metric is important to improve the classification performance.

3) No-Metric: No-metric achieves three wins and two ties than two objective combinations. Especially, no-metric wins six out eight datasets than RC+RS. No-metric has one more objective SC than RC+RS. RC+RS does not make use of label information while no-metric does. Hence, this observation reveals that the label information is useful in training models for classification.

No-metric is inferior to MOMM. The distance in the model space is critical to the performance of learning algorithms. This observation confirms that learning an elaborate distance metric does lead to performance gains.

C. Effects of RS and the Distance Metric

Previous research mainly determines satisfactory reservoir parameters [13], [31] by grid search or trial-and-error. Thus it is useful to study the influence of objectives on the reservoir parameters. We conduct 30 repetitions for three cases: 1) MOMM; 2) RC+SC; and 3) no-metric, respectively, on the Beef dataset, to see the effect of RS and the distance metric. In particular, we store the up-till-now best SC values for different parameters during evolution. This could be viewed as sampling from the parameter space using evolution method [35]. The result for sampling RC is omitted here because the variance is too small to be identified for visualization. Fig. 4 demonstrates the generalization performance of SC by sampling the reservoir size N and r_c tuples during the evolution process. We can make three observations.

1) By comparing MOMM and RC+SC, we find that taking the RS as an objective makes the search distribution in the solution space focused on the region of small reservoir size [Fig. 4(a)]. Despite the smaller reservoir for the MOMM, we observe better generalization of SC [compare Fig. 4(a) and (b)]. The observation confirms that the network size optimization is effective and is beneficial to improving the generalization performance.
2) Unexpectedly, with the objective RS to minimize the reservoir size, the cycle connection weight r_c favors large values comparing Fig. 4(a) and (c) to Fig. 4(b). Note that in previous work [31], r_c always varies in the range [0, 1]. We presume that large reservoir size would be required as a price for comparative performance in case of limited r_c.

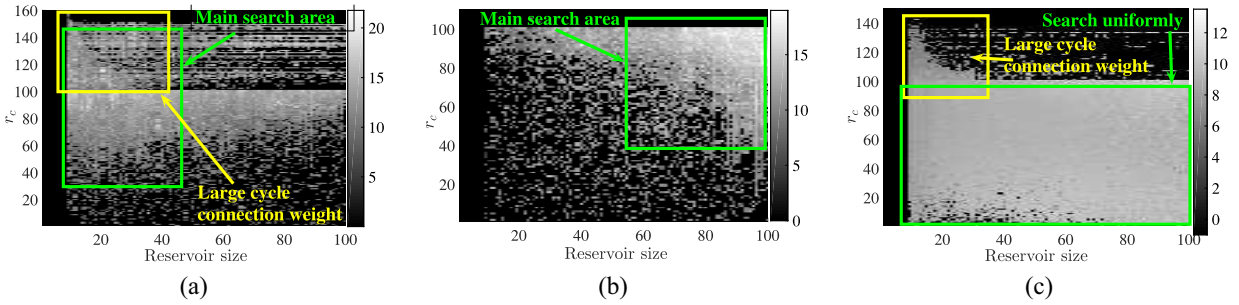


Fig. 4. Sampling objective values of SC on the test set with respect to different reservoir size and r_c ($\times 100$) while evolution on Beef. The minus sign of SC is omitted. The figure presents the result of (a) MOMM, (b) RC+SC, and (c) no-metric. We compare MOMM and RC+SC to reveal the effect of RS on the reservoir and compare MOMM and no-metric to reveal the effect of the distance metric on the reservoir. We can make three main observations. First, comparing MOMM and RC+SC [see (a) and (b)], the search for the reservoir size is successfully constrained in small size area (green box). Despite the small reservoir of the MOMM, better generalization is observed when compared with RC+SC. Second, the cycle connection weight tends to grow large (yellow box) when we restrict the reservoir size to be small [see (a) and (c)]. Third, the search area is more concentrated when the distance metric is learned [see (a)]. Whereas, without learning the distance metric, the algorithm tends to explore the parameter space more uniformly [see (c)].

3) By comparing MOMM and no-metric [Fig. 4(a) and (c)], it is observed that with the distance metric the parameter search is more concentrated and could help accelerate convergence. This observation may be because the distance metric enlarges the difference of SC values for different parameters. Hence, many areas in the parameter space are not searched, leading to the quick convergence of the MOMM (see Fig. 10).

D. Time Series Classification

1) Univariate Time Series: We compare the MOMM with the state-of-the-art methods. Table III presents the average classification accuracy over 30 times. The bottom row is the mean performance gap between other methods and MOMM. On the 23 datasets, the MOMM achieves the best classification accuracy on 15 datasets, and comparative performance with the best results on the other eight datasets. In particular, MOMM gets 99.77% accuracy on ECG5, 90.5% accuracy on OSULeaf, etc. which improves a lot compared to other approaches. We observe that the MOMM works extremely well when the training samples are rare, e.g., DiatomSize, ECG5 and SonyRobot, and SonyRobotII. In these datasets, the number of testing samples is much larger than training samples, which easily results in overfitting in classification. For example, ECG5 contains only 23 training samples but 861 test samples. The good performance of MOMM demonstrates that the co-evolution of network performance and network size helps avoid the overfitting problem. The MRV often gets inferior performance to RV. We presume this observation is because the flexibility of MOEA leads to high risk of overfitting.

The MMCL obtains the best accuracy on four datasets. However, MMCL needs to determine the combinational coefficients explicitly by CV. It requires more efforts to fine-tune the parameters in order to maintain comparative performance with the MOMM. Note that MOMM has the advantage of automatically tuning the network structure and tradeoffs according to the specific dataset. We observe that TSBF achieves excellent performance on FISH dataset. Fig. 7(f) exemplifies a typical time series in FISH dataset [36]. To explain the reason, note that FISH dataset

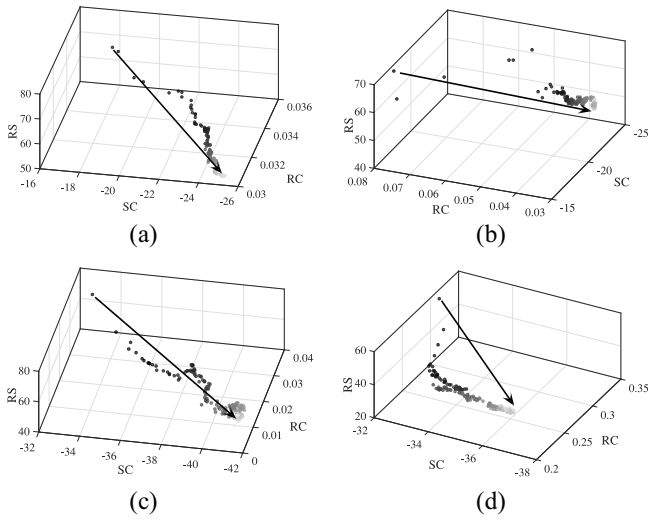


Fig. 5. Mean objective values on the nondominated front of different generations. The iteration is indicated by gray scale. The arrows point from iterations 1 to 200. (a) Beef. (b) Coffee. (c) Car. (d) Lighting2.

contains sequences of fish shape contours and we find that the sequences exhibit strong symmetric characteristics. TSBF obtains statistical properties directly by sampling subsequences. Therefore, TSBF may have more ability in finding the class-predictive features from the local structure, such as different parts of the fish. Hence, TSBF performs better than model-based approaches.

Fig. 5 illustrates the evolution of the MOMM. It presents the average objective values on the nondominated front of each iteration. We observe that the algorithm automatically determines the tradeoffs of the objective for different stages of optimization. For Coffee dataset, the algorithm first produces more pressure to optimize the SC. This is because the Coffee dataset contains two class of samples that can be easily separated in the model space. When the improvement on SC is saturated, the algorithm tends to put more emphasis on the RC and RS. For Lighting2 dataset, the MOMM first provides more priority in optimizing RC and then turns to SC.

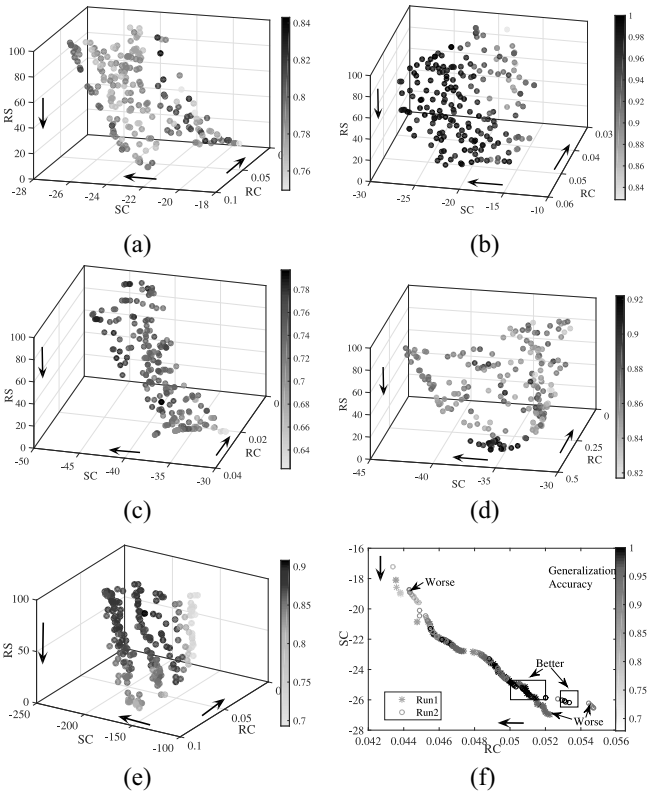


Fig. 6. Nondominated front achieved by MOMM. The axes indicate three objective values. The color indicates the generalization performance. (a) Beef. (b) and (f) Coffee. (c) Car. (d) Lighting2. (e) OSULeaf.

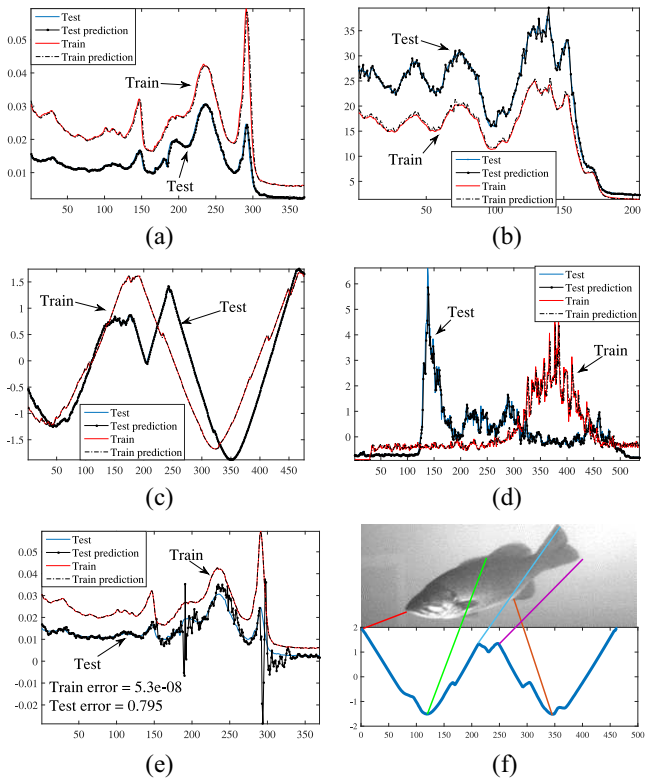


Fig. 7. Illustration of (a)–(e) prediction performance of the time series models learned on top of the optimized CRJ network and (f) Fish series.

Fig. 6 demonstrates the nondominated solutions achieved in the final generation of the MOMM. It is observed that the nondominated solutions obtained in the final generation have a good distribution in the objective space. The solutions with the best generalization performance may occupy different regions in the objective space. For Beef, Coffee, and Car, the best solutions lie in the interior region of the nondominated front. For Lighting2, the best solutions are achieved with only 13 neurons. It demonstrates the advantage of treating the reservoir size as the optimizing objective. Fig. 6(f) demonstrates the nondominated front of optimizing RC and SC, when RS is fixed as 50. The satisfactory solutions have relatively low RC and SC, which empirically shows the complementarity of RC and SC.

By employing the MOMM, we are able to solve time series classification and prediction in one time. To demonstrate the prediction performance, we choose the network on the nondominated front with the best RC value. Based on the selected network, the sequence model for a randomly chosen training sequence is learned. Then the trained sequence model is used for predicting a randomly chosen test sequence. The goal is to verify whether the temporal dynamics learned from the training sequence could be faithfully represented and generalize well on an unseen sequence. According to Fig. 7, despite the diverse dynamic behaviors of the sequences, the prediction results are quite satisfactory. For example, on Lighting2 dataset, even though the training sequence and test sequence show very broad and different dynamics, the prediction performance is very promising. In our experiments, we also notice that solutions trained on one dataset can be easily transferred to other datasets with promising performance. The NMSE on Beef dataset is 0.056, while employing the network learned from Coffee dataset also achieves 0.099 NMSE error. We also evaluate the prediction ability of models optimized by any two objectives. However, the prediction results are usually worse than optimizing three objectives. Besides, without treating RS as an objective, it is typical to observe a nearly perfect approximation on the training sequences, but suffers severe degradation on test sequences, as shown in Fig. 7(e). This is because the networks become too large and cause overfitting.

From the above results and supported by the previous sections, we could find: first, by optimizing the three objectives simultaneously, we could obtain better prediction and classification performance than optimizing the objectives individually. This indicates that RC and SC could complement each other. Second, the inclusion of RS as an objective indeed prevent the network from getting too complex and overfitting in learning sequence models. Third, by simultaneously optimizing three objectives in MOEA, the tedious human interaction to tune the combinational coefficient may be exempted, while satisfactory generalization performance is obtained.

2) Multivariate Time Series: The average results over 30 repetitions are presented in Table IV. The MOMM achieves the best accuracy on Libras and Hand, but slightly worse on AUS. Presumably, the inferior result is due to the short sequence length. Using a generative model to approximate the

TABLE IV
EXPERIMENTAL RESULTS FOR MULTIVARIATE AND VARIABLE LENGTH TIME SERIES

Datasets	*Dim*	*Length*	*#class*	*#Tr*	*#Te*	DTW	MESN	Fisher	RV	MRV	HV-MOTS	MMCL	MOMM
Libras	2	45	15	360	558	94.77	93.91	94.27	93.37	94.09	91.39	94.65	**95.32**
Hand	3	60∼182	20	600	2258	89.39	79.19	86.26	89.97	88.24	92.17	91.83	**92.94**
AUS	22	45∼136	95	600	1865	**97.05**	78.83	95.21	96.14	94.72	84.36	96.78	95.05
					Mean gap	0.70	10.46	2.52	1.27	2.09	5.13	0.23	

sequences has an advantage in dealing with long sequences, whereas this method may not be competent to capture the transient characteristics of this dataset. Therefore, in this situation, methods directly operating on raw data, such as DTW, may be more effective.

Up till now, the results of univariate and multivariate time series classification have demonstrated that the MOMM performs favorably in comparison with the state-of-the-art methods. Three reasons are involved.

1) The network weights and topology in the general-purpose reservoir network is critical to providing dynamic features for both approximation and discrimination. The network selection is usually a tedious trial-and-error process by restarting or CV. This requires a lot of user interactions and cannot guarantee to find optimal solutions. More advanced gradient descent-based MMCL may suffer from local optima or vanishing gradient issues when training the network recurrently. In the MOMM, the network optimization is automatically determined using a Pareto-based evolutionary algorithm. The parameters that need to be specified is the evolutionary algorithm parameters, such as population size, crossover, and mutation rates. These parameters are not so sensitive to the classification performance.
2) The distance metric is taken into consideration explicitly in the MOMM as an objective. This enables the network to capture supervised similarity information. The learned representations favor collapsing data in the same class and separating data in different classes.
3) By using the MOEA, the tradeoffs among the various objectives are effectively optimized without the need to specify the combinational coefficients.

3) Statistical Test: Friedman test is to compare multiple methods over multiple datasets. We first employ Friedman test to test whether the compared algorithms are equivalent. The p-value of Friedman test is 5×10^{-12}, which means these methods do behave differently. Then we conduct a post-hoc test, i.e., Bonferroni–Dunn test [37], to see how the MOMM is significantly different from other methods. For our case, we compare MOMM with all other methods. The significance level is set as 0.05. Fig. 8 demonstrates the mean rank of every method and their CD. The figure illustrates that the MOMM is significantly better than most state-of-the-art methods.

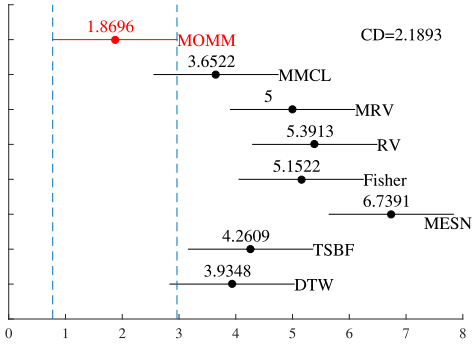


Fig. 8. CD between the MOMM and the compared methods. The numerical value above each line indicates the average rank of the corresponding method over all univariate datasets. The result indicates that MOMM achieves the best average rank. Most state-of-the-art methods are significantly different from our method under the significance level 0.05. The CD is 2.1893 for this case.

E. Discussion

1) Effectiveness of Solution Selection: How to exploit the nondominated front returned by MOEAs to output the final solution has been actively investigated [38], [39]. The most favorable solution on the nondominated front is usually determined by a decision maker. However, we note that for selecting a solution, there have been many heuristic methods proposed, which are readily available to this paper, such as trial-and-error [32], best-takes-all [3], [6], knee-based [40], hypervolume-based [5], and ensemble-based methods [41]. In our experiment, tenfold CV is used to select a solution with the best class distribution in the model space. We have also implemented three widely applied methods to compare the results.

1) Hypervolume-based selection [5] selects a solution that has the maximum dominance area on the nondominated front.
2) Ensemble-based method [41] takes into consideration all/a part of the nondominated solutions. In our method, the predicted label of each solution is the prediction of an SVM classifier trained on that solution. The final prediction of the classification is obtained through voting by all the solutions.
3) Best-takes-all [3], [6] selects the solution with the best SC value since our task is classification. If more than one solution has the same best SC value, the solution with the best RC value among them is selected.

Their results are compared with tenfold CV using boxplots to evaluate whether the significant difference exists. Fig. 9 demonstrates that the compared methods are all inferior to tenfold CV.

2) Generalization Ability: We set the stop condition as 300 iterations to observe the trend of objective values on test data. Fig. 10 demonstrates the evolution trend of RC and SC on

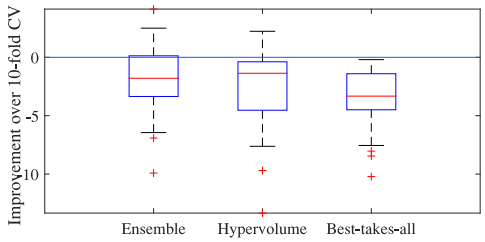
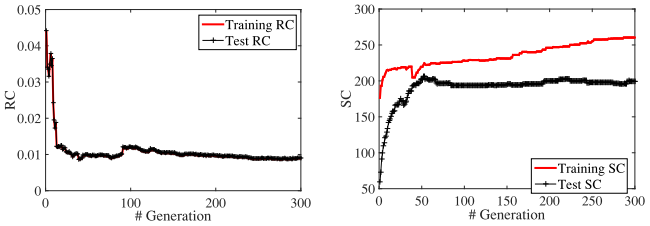


Fig. 9. Influence of different solution selection strategies. We perform ensemble [41], hypervolume-based [5], and best-takes-all [6] strategies to justify the result of the obtained nondominated front of each univariate dataset. These methods are compared with tenfold CV. The medium improvements (red line) of the three strategies are all below 0. p-value = 0.013/0.001/2.702e-05 for ensemble/hypervolume/best-takes-all in Wilcoxon signed rank test with a significance level of 0.05. Ensemble, hypervolume, and best-takes-all are significantly worse, which indicates that tenfold CV is an effective solution selection strategy in this paper.



(a) (b)

Fig. 10. Objective values of RC and SC with respect to the number of generation on Adiac dataset. The RC and SC values of each generation are averages of individuals on the nondominated front. It demonstrates that MOMM is able to converge after a few iterations and maintains good generalization performance.

the nondominated front of MOMM with respect to the number of the iteration. The generalization performance keeps steady as the evolution proceeds and does not show a tendency for overfitting. It also demonstrates that the MOMM algorithm converges quickly. For example, the fitness values on Adiac finish most of the improvement within 50 generations. Experiments on other datasets show similar results.

V. Conclusion

This paper proposes a multiobjective learning algorithm MOMM to classify time series in the model space. The problem is solved by simultaneously optimizing the network performance of time series prediction and the time series model separation, and by determining the network size automatically.

According to the experimental results, it is observed as follows.

1) The experiments on both univariate and multivariate time series datasets show that the MOMM algorithm achieves superior performance in comparison with the state-of-the-art time series classification methods.
2) Simultaneously optimizing the three objectives, i.e., the network performance of representation, SC, and the network size, demonstrates better performance than optimizing any two objectives. By optimizing the three objectives simultaneously, we could attain solutions for sequence prediction and classification in one run. The MOEA returns multiple solutions such that the user could choose a suitable solution for the task at hand.
3) For time series classification, the three objectives have different importance. Our result demonstrates that faithful future prediction capacity is critical for good generalization. The SC of learned models is also important to exploit the label information for good classification performance. The minimization of the network size produces much pressure against large networks, which helps reduce computational cost and improve the generalization ability.

We try to explain why the MOMM perform well by three main reasons.

1) The effective network structure and weight optimization using the MOEA.
2) The incorporation of distance metric and the network size when learning representations.
3) Effective tradeoffs among objectives by considering them explicitly in a multiobjective learning algorithm.

Our methodology has a high computational complexity, as other evolutionary algorithms do. However, it is balanced by the fact of being able to search enormous parameter space and superior performance in comparison with the state-of-the-art methods. This would be important in applications where the user concerns the accuracy more than time, such as medical diagnosis [4]. The improvement in computational efficiency will be made in the future. One way is to reduce the number of fitness evaluation and comparisons by designing more efficient MOEAs. It is also possible to achieve further performance enhancement by explicitly considering diversity as an optimization objective.

Acknowledgment

The authors would like to thank the Editor-in-Chief, the Associate Editor, and the anonymous reviewers for their constructive and important comments.

References

[1] R. Goroshin, J. Bruna, J. Tompson, D. Eigen, and Y. LeCun, "Unsupervised learning of spatiotemporally coherent metrics," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, 2015, pp. 4086–4093.

[2] H. Chen, P. Tino, A. Rodan, and X. Yao, "Learning in the model space for cognitive fault diagnosis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 1, pp. 124–136, Jan. 2014.

[3] C.-F. Juang and Y.-T. Yeh, "Multiobjective evolution of biped robot gaits using advanced continuous ant-colony optimized recurrent neural networks," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2017.2718037.

[4] J. Wiens, E. Horvitz, and J. V. Guttag, "Patient risk stratification for hospital-associated C. *diff* as a time-series classification task," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 467–475.

[5] P. Esling and C. Agon, "Multiobjective time series matching for audio classification and retrieval," *IEEE Audio, Speech, Language Process.*, vol. 21, no. 10, pp. 2057–2072, Oct. 2013.

[6] M. Delgado, M. P. Cuellar, and M. C. Pegalajar, "Multiobjective hybrid optimization and training of recurrent neural networks," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 381–403, Apr. 2008.

[7] T. Rakthanmanon *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, Beijing, China, 2012, pp. 262–270.

[8] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. J. Keogh, "Querying and mining of time series data: Experimental comparison of representations and distance measures," *Proc. Very Large Database Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.
[9] Y. Chen *et al.* (2015). *The UCR Time Series Classification Archive*. [Online]. Available: http://www.cs.ucr.edu/ eamonn/time_series_data/
[10] M. G. Baydogan, G. Runger, and E. Tuv, "A bag-of-features framework to classify time series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2796–2802, Nov. 2013.
[11] T. S. Jaakkola, M. Diekhans, and D. Haussler, "Using the fisher kernel method to detect remote protein homologies," in *Proc. 7th Int. Conf. Intell. Syst. Mol. Biol.*, vol. 99, pp. 149–158, 1999.
[12] L. Van der Maaten, "Learning discriminative fisher kernels," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 217–224.
[13] H. Chen, F. Tang, P. Tino, and X. Yao, "Model-based kernel for efficient time series analysis," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2013, pp. 392–400.
[14] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. Salakhutdinov, "Neighbourhood components analysis," in *Proc. Adv. Neural Inf. Proce. Syst.*, Vancouver, BC, Canada, 2004, pp. 513–520.
[15] X. Yao, "Evolving artificial neural networks," *Proc. IEEE*, vol. 87, no. 9, pp. 1423–1447, Sep. 1999.
[16] C.-K. Goh, E.-J. Teoh, and K. C. Tan, "Hybrid multiobjective evolutionary design for artificial neural networks," *IEEE Trans. Neural Netw.*, vol. 19, no. 9, pp. 1531–1548, Sep. 2008.
[17] D. Kudithipudi, Q. Saleh, C. Merkel, J. Thesing, and B. Wysocki, "Design and analysis of a neuromemristive reservoir computing architecture for biosignal processing," *Front. Neurosci.*, vol. 9, p. 502, Feb. 2016.
[18] A. Chandra and X. Yao, "Ensemble learning using multi-objective evolutionary algorithms," *J. Math. Model. Algorithms*, vol. 5, no. 4, pp. 417–445, 2006.
[19] H. Chen, F. Tang, P. Tino, A. G. Cohn, and X. Yao, "Model metric co-learning for time series classification," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Buenos Aires, Argentina, 2015, pp. 3387–3394.
[20] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *Proc. Int. Conf. Data Min.*, 2001, pp. 1–11.
[21] M. Cuturi, J.-P. Vert, O. Birkenes, and T. Matsui, "A kernel for time series based on global alignments," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 2. Honolulu, HI, USA, 2007, pp. 413–416.
[22] J. Lin, E. Keogh, L. Wei, and S. Lonardi, "Experiencing SAX: A novel symbolic representation of time series," *Data Min. Knowl. Disc.*, vol. 15, no. 2, pp. 107–144, 2007.
[23] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A Kullback–Leibler divergence based kernel for SVM classification in multimedia applications," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1385–1392.
[24] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels," *J. Mach. Learn. Res.*, vol. 5, pp. 819–844, Jan. 2004.
[25] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
[27] A. F. Krause, V. Dürr, B. Bläsing, and T. Schack, "Multiobjective optimization of echo state networks for multiple motor pattern learning," in *Proc. 18th IEEE Workshop Nonlin. Dyn. Electron. Syst.*, Dresden, Germany, 2010, pp. 190–193.
[28] H. Jaeger, "Short term memory in echo state networks," GMD German Nat. Res. Center Inf. Technol., Bonn, Germany, GMD Rep. 152, 2001.
[29] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Comput. Sci. Rev.*, vol. 3, no. 3, pp. 127–149, 2009.
[30] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks," GMD German Nat. Res. Center Inf. Technol., Bonn, Germany, GMD Rep. 148, 2001.
[31] A. Rodan and P. Tino, "Simple deterministically constructed cycle reservoirs with regular jumps," *Neural Comput.*, vol. 24, no. 7, pp. 1822–1852, 2012.
[32] W. Cai, S. Chen, and D. Zhang, "A multiobjective simultaneous learning framework for clustering and classification," *IEEE Trans. Neural Netw.*, vol. 21, no. 2, pp. 185–200, Feb. 2010.
[33] H. Wang and X. Yao, "Corner sort for Pareto-based many-objective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 92–102, Jan. 2014.
[34] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, p. 27, 2011.
[35] J. A. G. de Visser and J. Krug, "Empirical fitness landscapes and the predictability of evolution," *Nat. Rev. Genet.*, vol. 15, no. 7, pp. 480–490, 2014.
[36] D.-J. Lee, J. K. Archibald, R. B. Schoenberger, A. W. Dennis, and D. K. Shiozawa, "Contour matching for fish species recognition and migration monitoring," in *Applications of Computational Intelligence in Biology*. Heidelberg, Germany: Springer, 2008, pp. 183–207.
[37] O. J. Dunn, "Multiple comparisons among means," *J. Amer. Stat. Assoc.*, vol. 56, no. 293, pp. 52–64, 2012.
[38] W.-Y. Chiu, G. G. Yen, and T.-K. Juan, "Minimum manhattan distance approach to multiple criteria decision making in multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 972–985, Dec. 2016.
[39] J.-H. Kim, J.-H. Han, Y.-H. Kim, S.-H. Choi, and E.-S. Kim, "Preference-based solution selection algorithm for evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 20–34, Feb. 2012.
[40] J. Branke, K. Deb, H. Dierolf, and M. Osswald, "Finding knees in multi-objective optimization," in *Proc. Int. Conf. Parallel Problem Solving From Nat.*, Birmingham, U.K., 2004, pp. 722–731.
[41] H. Chen and X. Yao, "Multiobjective neural network ensembles based on regularized negative correlation learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 12, pp. 1738–1751, Dec. 2010.

Zhichen Gong (S'17) received the B.Eng. degree in computer science from Anhui University, Hefei, China, in 2015. He is currently pursuing the master's degree with the UBRI Laboratory, School of Computer Science and Technology, University of Science and Technology of China, Hefei.

His current research interests include machine learning, dynamical learning systems, and sequential data mining.

Huanhuan Chen (SM'16) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 2004, and the Ph.D. degree in computer science from the University of Birmingham, Birmingham, U.K., in 2008.

He is currently a Professor with the UBRI Laboratory, School of Computer Science and Technology, USTC. His current research interests include statistical machine learning, data mining, fault diagnosis, and evolutionary computation.

Prof. Chen was a recipient of the 2015 International Neural Network Society Young Investigator Award, the IEEE Computational Intelligence Society Outstanding Ph.D. Dissertation Award (the only winner), the 2009 CPHC/British Computer Society Distinguished Dissertations Award (the runner up), and the IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award (bestowed in 2011 and only one paper in 2009) for his work on probabilistic classification vector machines on Bayesian machine learning. He is an Associate Editor of the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and the IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE.

Bo Yuan (M'15) received the B.Sc. and Ph.D. degrees in electronic information science and technology from the University of Science and Technology of China (USTC), Hefei, China, in 2009 and 2014, respectively.

He is currently an Assistant Professor with the Department of Computer Science and Engineering, Southern University of Science and Technology (SUSTech), Shenzhen, China. Before joining SUSTech, he was an Associate Professor with the School of Computer Science and Technology, USTC. His current research interests include evolutionary computation, electronic design automation, and machine learning.

Xin Yao (F'03) received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, in 1982, the M.Sc. degree from the North China Institute of Computing Technology, Haidian, China, in 1985, and the Ph.D. degree from USTC in 1990.

He is a Chair Professor of Computer Science with the Southern University of Science and Technology, Shenzhen, China, and a Professor of Computer Science with the University of Birmingham, Birmingham, U.K. He has been researching on multiobjective optimization since 2003, when he published a well-cited EMO'03 paper on many objective optimization. His current research interests include evolutionary computation, ensemble learning, and their applications in software engineering.

Prof. Yao was a recipient of the 2001 IEEE Donald G. Fink Prize Paper Award, the 2010, 2016, and 2017 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Awards, the 2010 BT Gordon Radley Award for Best Author of Innovation (finalist), the 2011 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award, the prestigious Royal Society Wolfson Research Merit Award in 2012, the IEEE Computational Intelligence Society (CIS) Evolutionary Computation Pioneer Award in 2013, and many other best paper awards. He was the President of the IEEE CIS from 2014 to 2015, and the Editor-in-Chief of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2003 to 2008. He is a Distinguished Lecturer of the IEEE CIS.