Neural Processing Letters (2019) 50:1789–1829
https://doi.org/10.1007/s11063-018-9957-7

Several Novel Dynamic Ensemble Selection Algorithms for Time Series Prediction

Changsheng Yao[1] · Qun Dai[1] · Gang Song[1]

Published online: 30 November 2018

Abstract

The goal to improve prediction accuracy and robustness of predictive models is quite important for time series prediction (TSP). Multi-model predictions ensemble exhibits favorable capability to enhance forecasting precision. Nevertheless, a static ensemble system does not always function well for all the circumstances. This work proposes six novel dynamic ensemble selection (DES) algorithms for TSP, including one DES algorithm based on Predictor Accuracy over Local Region (DES-PALR), two DES algorithms based on the Consensus of Predictors (DES-CP) and three Dynamic Validation Set determination algorithms. The first dynamic validation set determination algorithm is designed based on the similarity between the Predictive value of the test sample and the Objective values of the training samples. The second one is constructed based on the similarity between the Newly constituted sample for the test sample and All the training samples. Finally, the third one is developed based on the similarity between the Output profile of the test sample and the Output profile of each training sample. These proposed algorithms successfully realize dynamic ensemble selection for TSP. Experimental results on twelve benchmark time series datasets have demonstrated that the proposed DES algorithms greatly improve predictive performance when compared against current state-of-the-art prediction algorithms and the static ensemble selection techniques.

Keywords Dynamic ensemble selection (DES) · DES algorithm based on Predictor Accuracy over Local Region (DES-PALR) · Dynamic Ensemble Selection algorithm based on the Consensus of Predictors (DES-CP) · Dynamic validation set determination algorithm · Time series prediction (TSP)

1 Introduction

Time series can be defined as a set of sequential observations, of a variable of interest, recorded over a predefined period of time [1]. Time series are widely used today because we need to know the future behavior of certain relevant phenomena in order to plan, prevent,

✉ Qun Dai
daiqun@nuaa.edu.cn

[1] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China

and so on. That is, to predict what will happen with a variable in the future from the behavior of that variable in the past [2]. The word “prediction” comes from the Latin “prognosticum”, which means “I know in advance” [3]. Time series forecasting techniques could be conditionally classified into long-term time series forecasting techniques and short-term time series forecasting ones [4].

Time series prediction (TSP) is an important and active research topic in machine learning, and it has indispensable importance in many practical data mining applications. In general, time series involves a subject of research interest in various areas of knowledge engineering, such as: agriculture (the number of pigs slaughtered, sheep population, and milk production), health (suicide rates, fertility rates and number of cases of measles), finance (stocks, loans and exchange rates), and production (beer shipments, motor vehicle production and electricity production), etc.

Financial time series forecasting (such as stock forecasting and crude oil price forecasting) is one of the most popular research directions in the field of time series prediction. However, there exist many factors that will influence a stock market, including the basic situation of listed companies, national macroeconomic policies, market supply and the technical indicators of shares [5]. Investors are facing with a great challenge that they do not know how to precisely forecast price fluctuation in financial markets. It is a hard task to predict the trend of stock market because of its high volatility and the noisy environment.

There has been an increasing interest in using Neural Networks (NNs) to model and forecast time series over the last decades. NNs have been found to be a viable contender to various traditional time series models [6–8]. Lapedes and Farber [9] reported the first attempt to model nonlinear time series with NNs. Chakraborty et al. [10] conducted an empirical study on multivariate time series forecasting with NNs. What’s more, several forecasting competitions [6, 11] show that NNs could be a very useful addition to the time series forecasting toolbox.

One of the major developments in NNs over the last decade is models combining or ensemble learning. An ensemble can be formed by multiple network architectures, different initial random weights, different number of hidden nodes, or even different activation functions. Multi-model ensemble prediction systems show convincing ability to improve the forecast performance in different areas of computational science [12, 13]. In short, two heads are better than one.

Multiple Predictor Systems (MPSs) are typically composed of three stages [14]: (1) Generation; (2) Selection; and (3) Integration. In the first stage, a pool of predictors is generated. In the second stage, a single predictor (or several predictors) having better predictive prediction on the validation set than the others is (are) selected into the ensemble. We refer to the subset of predictors as Ensemble of Predictors (EoP). In the last stage, the predictions of the selected predictors are combined by some ways for the final results [15, 16].

For the second stage, there exist two types of selective ensemble paradigms, i.e., static and dynamic ensemble selection [17]. Within the static ensemble selection paradigm, the selected predictors of the ensemble will remain unchanged for the prediction of all the test samples [18–20]. While the assumption of dynamic ensemble selection (DES) paradigm is that every predictor is an expert in some specific local regions. Just the opposite to static ensemble selection, a predictor or several predictors which specializes (specialize) in conducting prediction for the new test sample will be selected into the final ensemble dynamically within the DES paradigm. Recently, several literatures show that DES is a very effective tool for TSP problems [21–23].

The important basis of DES is how to determine, with regard to the current test sample, the appropriate validation samples from the training set. This topic has received considerably limited research attention so far in TSP problems. However, for classification problems, Woods et al. [24] proposed an approach that is Dynamic Classifier Selection by Local Accuracy (DCS-LA), and the validation samples are chosen as the K-nearest neighbors (KNNs) training samples to the new test one. Smits [25] proposed the measure of Modified Local Accuracy (MLA), which is similar to DCS-LA, with the only difference being that each sample belonging to the validation set is weighted by its Euclidean distance to the test sample. And the MLA outperforms DCS-LA with respect to minimum accuracy, maximum accuracy and kappa value. Kuncheva [26] conducted a study that the validation set is determined by using the clustering techniques.

Due to the fact that the above methods, which are used to select the validation data, is prone to be limited by the quality of the region of competence defined in the training data. Hence, the decision templates (DT) [27] technique is considered to select validation set. The goal of DT is also to select samples that are close to the test instance. However, the similarity is computed over decision space rather than feature space. This is performed by transforming both the test instance and the training data into the corresponding output profiles. The output profile of one sample is a vector that consists of the predicted values obtained by the base predictors for that sample.

As mentioned above, determining a proper validation set is a critical issue for the successful implementation of DES algorithms, which has received very limited attention for the research of TSP. Therefore, we decide to carry out some innovative research work in this area, so as to further promote the research progress in the respect of TSP. We find through analysis that, there are mainly three schemes to address this issue, i.e., the dynamic determination of the validation set based upon feature space solely, decision space solely, or based on the integration of feature and decision space. The DES algorithm based on Predictor Accuracy over Local Region (DES-PALR) proposed in this work belongs to the first scheme. It performs k-means clustering algorithm on the feature space of the training set, which generates several clusters of the training samples. The cluster whose center is the nearest to the feature vector of the current test sample will be selected as the validation set dynamically.

The proposed Dynamic Validation Set determination algorithm based on the similarity between the Predictive value of the test sample and the Objective values of the training samples (DVS-PvOv) belongs to the second scheme. While the proposed Dynamic Validation Set determination algorithm based on the similarity between the **N**ewly constituted **s**ample for the test sample and All the training **s**amples (DVS-NsAs) is part of the third scheme, determining the validation set based on the integration of feature and decision space. The finally proposed Dynamic Validation Set determination algorithm based on the similarity between the Output profile of the test sample and the Output profile of each training sample (DVS-OpOp) belongs to the second scheme.

Another important point of DES is how to dynamically choose an appropriate ensemble constituted with some selected models. Adhikari et al. [16] proposed an ensemble selection method that selectively combines some of the constituent forecasting models, instead of combining all of them. And on each time series, the constituent models are successively ranked as per their past forecasting accuracies. Then the forecasts of a group of high ranked models are combined to produce the final predictive results. In the respects of selective ensemble, Zhou et al. [28] proposed an approach which trains several individual NNs and then employs genetic algorithm to select an optimum subset of individual networks to constitute the final ensemble.

On the other hand, there exist some DES techniques which are based upon other criteria, such as the degree of consensus or confidence of the ensemble decision. Dos Santos et al. [29] proposed the Margin-based Dynamic Selection (MDS) and Ambiguity-guided Dynamic

Selection (ADS). The criterion of MDS is the margin between the most voted class and the second most voted class. ADS uses the ambiguity among the base classifiers of a pool of ensembles of classifiers as the criterion for measuring its competence level. The ambiguity is determined as the number of base classifiers of an ensemble that disagree with the ensemble decision.

Inspired by the existing research work, we propose a Dynamic Ensemble Selection algorithm based on the Consensus of Predictors (DES-CP) for TSP in this paper. DES-CP is developed based on the extent of the ensemble consensus, similarly. And it works by considering a pool of ensembles of predictors (EoPs) rather than a pool of predictors. Several EoPs are generated by the Genetic Algorithm based Selective ENsemble (GASEN) [28]. And according to the different approaches employed to evaluate the extent of consensus, the proposed DES-CP algorithm is further subdivided into two algorithms: (1) for each test sample, the consensus of each EoP is evaluated by calculating its Variance (namely, DES-CP-Var); (2) conducting Clustering algorithm for evaluating the consensus of each EoP (namely, DES-CP-Clustering). The ensemble, which has the highest consensus, is chosen as the final EoP.

Our motivations behind the developments of these novel dynamic ensemble selection algorithms for the research of TSP mainly lie in that, in the domain of TSP, instead of splitting the original dataset into three disjunctive parts, i.e., training set, validation set and test set, like most of the static ensemble pruning methods, dynamically determining the validation dataset for each distinctive test sample is more reasonable. The superiority of DES over static ensemble selection on prediction performance has been verified in the literatures.

Specifically, the motivation for the proposal of DES-PALR is that, predictive accuracy is the most essential criterion for the implementation of DES for TSP. With the design of DES-PALR, the local region has more similar distribution with the test sample than other training samples. The predictors which perform better on the local region could also perform better on the test sample, while different test sample locates different region of competence, effectively achieving dynamic ensemble selection.

The proposal of the three novel algorithms, i.e., DVS-PvOv, DVS-NsAs and DVS-OpOp, further facilitates the effective implementation of DES for TSP. Their common characteristic is that, they are not limited by the quality of the local region of competence solely defined in the feature space, which is beneficial to their predictive performances.

With the proposed DES-CP algorithm, the higher the consensus among the member predictors is, the higher the level of prediction confidence will be. By maximizing the extent of ensemble consensus, the certainty that the ensemble will make a more accurate prediction is improved. Its major merit is that, it does not need any information from the region of competence, and therefore, it is not restricted by the algorithms which define the region of competence.

In Table 1, the advantages and disadvantages of the proposed five DES algorithms are listed, so as to provide a brief and clear comparison among these algorithms.

To our best knowledge, it is the first time that all these new dynamic ensemble selection algorithms are proposed for the research of TSP.

There exist two other advantages of this work. Firstly, ELM is used to be the base model; therefore our algorithms naturally inherit those salient advantages of ELM, including better generalization capability, fast learning speed and the avoidance of local minima problem. Secondly, the GASEN algorithm is employed to generate some EoPs for the proposed DES-CP algorithm. Comparing with the popular ensemble approach, i.e., averaging all, and the theoretically optimum selective ensemble approach, i.e., enumerating, GASEN has preferable

Table 1 The advantages and disadvantages of the proposed DES algorithms

Model	Advantages	Disadvantages
DES-PALR	Simple; relatively high speed testing phase; good performance	Trapped by the techniques which define the local region; influenced by the outliers
DES-CP	Higher prediction accuracy; stronger robust; high speed testing phase	Space-consuming; spend much time generating the pool of ensemble of predictors
DVS-PvOv	Not restricted by the algorithms which define the local region; extremely fast training	Only consider the similarity between the target value of training samples and the predicted value of predictor; time-consuming
DVS-NsAs	Consider both the feature of testing sample and the predicted value of predictor; extremely fast training	Time-consuming
DVS-OpOp	Do not need the information of the local domain; extremely fast training	Spend much time computing the output profiles of the training samples; trapped by the techniques which define the decision space

performance in generating EoPs with both strong generalization ability and small computational cost.

To verify the efficacy of the proposed six DES algorithms, we conducted experiments to compare the predictive performance of the proposed algorithms with three static ensemble selection algorithms and three other state-of-the-art models on twelve benchmark time series prediction datasets. The experimental results indicate that, in most cases, our proposed DES algorithms significantly outperform their competitors on the twelve datasets.

The remainder of the paper is organized as follows. Section 2 introduces some important concepts of ELM, and presents some theoretical analysis on ensemble pruning for time series forecasting tasks. Section 3 gives the novel ideas and details of the proposed DES algorithms for TSP. Section 4 reports and discusses the experimental results. Finally, Sect. 5 concludes this paper and suggests directions for future work.

2 Background Knowledge of ELM and Theoretical Analysis on Ensemble Pruning for TSP

2.1 Preview with ELM

Recently, extreme learning machine (ELM) has become an increasingly significant research topic for machine learning and artificial intelligence, due to its unique advantages, i.e., good generalization performance, extremely fast training speed and universal approximation ability. ELM is a valid solution for single-hidden layer feedforward neural networks (SLFNs). SLFNs have relatively strong ability of nonlinear approximation for regression problems, and can form separable decision regions for classifying arbitrary dimensional data.

It was Huang et al. [30] who originally developed the ELM algorithm. It possesses the same architecture as the SLFN, as illustrated in Fig. 1. The most outstanding feature of ELM

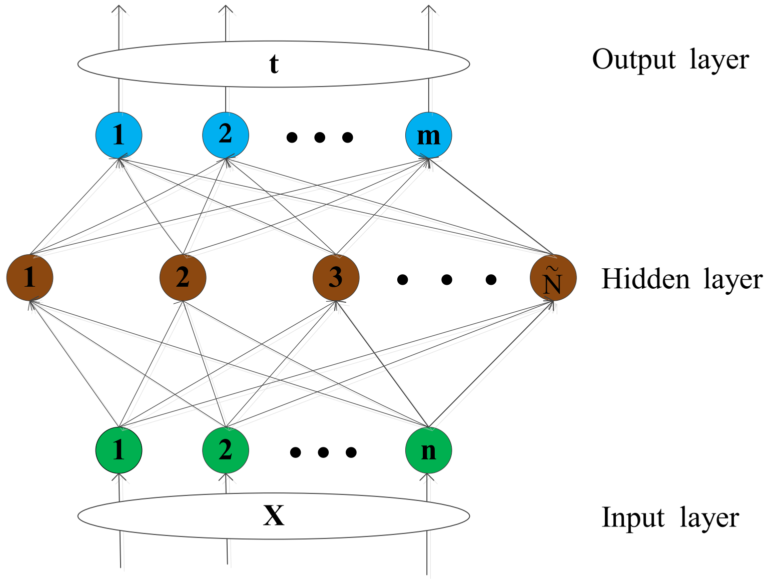


Fig. 1 ELM network topology

lies in the random initialization of its input weights and hidden layer biases, while its output weights are calculated simply by performing a matrix inversion on the hidden layer output matrix. ELM gets its name from its extremely fast learning speed, while it is also extremely easy to implement [31].

Suppose there are N arbitrarily different samples $(\mathbf{x}_i, \mathbf{t}_i)$, where $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{in}]^T \in \mathbf{R}^n$ and $\mathbf{t}_i = [t_{i1}, t_{i2}, \ldots, t_{im}]^T \in \mathbf{R}^m$, a normal SLFNs with $\tilde{N}$ hidden nodes and activation function $g(x)$ is analytically modeled by:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_j) = \mathbf{t}_j, \quad j = 1, \ldots, N \tag{1}$$

where $\mathbf{w}_i = [w_{i1}, w_{i2}, \ldots, w_{in}]^T$ denotes the connection weight vector between the ith hidden node and the input nodes, $[\beta_{i1}, \beta_{i2}, \ldots, \beta_{in}]^T$ denotes the connection weight vector between the ith hidden node and the output nodes, and b_i represents the bias of the ith hidden neuron.

Equation (1) can also be written compactly as:

$$\mathbf{H\beta} = \mathbf{T} \tag{2}$$

where $\mathbf{H}(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}}, \mathbf{x}_1, \ldots, \mathbf{x}_N) =$
$$\begin{bmatrix} \mathrm{g}(\mathbf{w}_1 \cdot \mathbf{x}_1 + \mathrm{b}_1) & \ldots & \mathrm{g}(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_1 + \mathrm{b}_{\tilde{N}}) \\ \vdots & & \vdots \\ \mathrm{g}(\mathbf{w}_1 \cdot \mathbf{x}_N + \mathrm{b}_1) & \cdots & \mathrm{g}(\mathbf{w}_{\tilde{N}} \cdot \mathbf{x}_N + \mathrm{b}_{\tilde{N}}) \end{bmatrix}_{N \times \tilde{N}}, \boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \text{ and } \mathbf{T} = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}_{N \times m}.$$

Here, $\mathbf{w}_i \cdot \mathbf{x}_j$ denotes the inner product of $\mathbf{w}_i$ and $\mathbf{x}_j$. As named by Huang et al. [32], $\mathbf{H}$ is termed the hidden layer output matrix of the neural network; the ith column of $\mathbf{H}$ is the ith hidden node output in relation to inputs $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$.

Traditionally, to train a SLFN, we might expect to acquire specific $\hat{\mathbf{w}}_i$, b_i, $\hat{\boldsymbol{\beta}}(i = 1, \ldots, \tilde{N})$ to minimize $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|$. Specifically,

$$\left\| H(\hat{\mathbf{w}}_1, \ldots, \hat{\mathbf{w}}_{\tilde{N}}, \hat{b}_1, \ldots, \hat{b}_{\tilde{N}})\hat{\beta} - \mathbf{T} \right\| = \min_{\mathbf{w}_i, b_i, \beta} \left\| H(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}})\beta - \mathbf{T} \right\| \tag{3}$$

which amounts to minimizing the cost function

$$E = \sum_{j=1}^{N} \left(\sum_{i=1}^{\tilde{N}} \boldsymbol{\beta}_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i) - \mathbf{t}_j \right)^2. \tag{4}$$

The gradient-decent learning algorithms are typically implemented to optimize these parameters, which update parameter vector $\mathbf{w}$ iteratively according to:

$$\mathbf{w}_k = \mathbf{w}_{k-1} - \eta \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} \tag{5}$$

Here η is learning rate. The most popular algorithm utilized is the BP learning algorithm. However, BP algorithm faces a series of problems, such as the difficulty to find an appropriate learning rate η; easy to fall into local minima; easy to overfitting; and rather time-consuming.

However, ELM resolves the above issues. It sets input weights $\mathbf{w}_i$ and hidden neuron bias b_i at random, after which matrix $\mathbf{H}$ is calculated directly. Then the problem of minimizing cost function in Eq. (3) equates acquiring a least-squares solution $\hat{\boldsymbol{\beta}}$ of the linear system $\|\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|$:

$$\left\| H(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}})\hat{\boldsymbol{\beta}} - \mathbf{T} \right\| = \min_{\mathbf{w}_i, b_i, \beta} \left\| H(\mathbf{w}_1, \ldots, \mathbf{w}_{\tilde{N}}, b_1, \ldots, b_{\tilde{N}})\boldsymbol{\beta} - \mathbf{T} \right\| \tag{6}$$

If the number $\tilde{N}$ of hidden nodes is equal to the number N of distinct training samples, $\tilde{N} = N$, matrix $\mathbf{H}$ is square and invertible when the input weight vectors $\mathbf{w}_i$ and the hidden biases b_i are randomly chosen, and these training samples can be approximated with zero error. However, in most cases, the number of hidden nodes is much less than the number of distinct training samples, $\tilde{N}$, $\mathbf{H}$ is nonsquare matrix and there may not exist $\mathbf{w}_i$, b_i, $\beta_i(i = 1, \ldots, \tilde{N})$ such that $\mathbf{H}\boldsymbol{\beta} = \mathbf{T}$. ELM learns the output weights $\boldsymbol{\beta}$ with the use of a Moore–Penrose generalized inverse of the matrix $\mathbf{H}$, denoted as $\mathbf{H}^\dagger$ [32]. The smallest norm least squares solution of the above linear system is

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{T} \tag{7}$$

The solution $\hat{\boldsymbol{\beta}}$ defined in Eq. (7) has the norm minimum over all the solutions of the least squares solutions of linear system in Eq. (2). Thus, $\hat{\boldsymbol{\beta}}$ deserves the best generalization performance across all the other least squares solutions [33].

A few apparent advantages exist in ELM: (1) the extremely fast learning speed; (2) the obviously better generalization performance; (3) without problems like local minima and slow rate of convergence, etc. [30].

2.2 Theoretical Analysis on Ensemble Pruning for TSP

Provided an equidistant sampled time series $\{\alpha_v\}_{v=1,\ldots,N}$, one m-dimensional state space vector $\boldsymbol{x}_t$ is constructed as the form:

$$\boldsymbol{x}_t = (\alpha_{t+1}, \alpha_{t+2}, \ldots, \alpha_{t+m}) \tag{8}$$

$$\alpha'_{t+m+s} = f(\boldsymbol{x}_t) \tag{9}$$

where s denotes the scope of prediction, m represents the time window size (TWS), and the function $f : R^m \to R$ is called the approach function.

As only one-step-ahead prediction is focused on in this paper, s is set as one. And the TSP problem here is considered as a special case of the function approximation problem. Suppose an ensemble comprising M base models is used to approximate the function $f : R^m \to R$, and the predictions of M base models are combined through weighted averaging for developing the final result. The weight $w_i (i = 1, 2, \ldots, M)$ that satisfies both Eqs. (10) and (11) is assigned to the i-th base model f_i.

$$0 \le w_i \le 1 \tag{10}$$

$$\sum_{i=1}^{M} w_i = 1 \tag{11}$$

The outcome of the ensemble is computed according to Eq. (12):

$$\tilde{f} = \sum_{i=1}^{M} w_i f_i \tag{12}$$

For convenience of discussion, here it is supposed that all the base models possess identical weights, just as in Eq. (13):

$$w_i = 1/M \quad (i = 1, 2, \ldots, M) \tag{13}$$

Then Eq. (12) becomes Eq. (14):

$$\tilde{f} = \frac{1}{M} \sum_{i=1}^{M} f_i \tag{14}$$

According to Zhou et al.'s analyses [28, 34], the reason why combining a proper subset of the original ensemble might outperform combining the entire one is explained as follows:

Suppose $\boldsymbol{x}_\mathrm{t} \in \mathrm{R}^\mathrm{m}$ is randomly sampled on the basis of a distribution $p(\boldsymbol{x}_t)$. The generalization error of the i-th individual model and that of the entire ensemble on $\boldsymbol{x}_\mathrm{t}$ are respectively:

$$E_i = (f_i(\boldsymbol{x}_t) - \alpha_{t+m+1})^2 \tag{15}$$

$$\tilde{E} = \left(\tilde{f}(\boldsymbol{x}_t) - \alpha_{t+m+1}\right)^2 \tag{16}$$

The correlation between the i-th and the j-th base models is calculated as below:

$$C_{ij} = \int d\boldsymbol{x}_t p(\boldsymbol{x}_t)(f_i(\boldsymbol{x}_t) - \alpha_{t+m+1})\left(f_j(\boldsymbol{x}_t) - \alpha_{t+m+1}\right) \tag{17}$$

Then, it can be obtained that:

$$\tilde{E} = \sum_{i=1}^{M} \sum_{j=1}^{M} C_{ij} \Big/ M^2 \tag{18}$$

If the k-th base model is removed from the pool of base models, the generalization error of current ensemble will be:

$$\tilde{E}' = \sum_{\substack{i=1 \\ i \neq k}}^{M} \sum_{\substack{j=1 \\ j \neq k}}^{M} C_{ij} \Big/ (M-1)^2 \tag{19}$$

Considering Eqs. (18) and (19), it can be gotten that:

$$\tilde{E} - \tilde{E}' = \frac{2\sum_{\substack{i=1\\ i\neq k}}^{M} C_{ik} + E_k - (2M-1)\tilde{E}}{(M-1)^2} \tag{20}$$

In this circumstance, the generalization error of the pruned ensemble is clearly smaller than that of the original entire ensemble, i.e., $\tilde{E}' < \tilde{E}$. Consequently, the conclusion could be reached that, aggregating an appropriate subensemble of the original ensemble might achieve preferable generalization performance, compared with aggregating the entire one.

3 The Proposed Several Novel Dynamic Ensemble Selection Algorithms for TSP

According to the literature, it can be found that the predictive precision of ELM is usually better than BP and SVM in many fields. However, the input weights and biases of ELM are randomly assigned, which will produce much uncertainty, especially when every new sample owns its particular property. Therefore, they should not be treated equally without discrimination [30]. Hence, an ensemble of ELMs rather than a single ELM is used in this work for the research of TSP, as an ensemble has better adaptability and stronger robustness compared to a single ELM.

In this work, several novel dynamic ensemble selection algorithms are proposed specifically for the research of TSP. Let $P = \{p_1, p_2, \ldots, p_M\}$ be the initial pool of predictors, where $p_i, i = 1, 2, \ldots, M$ denote the base predictors belonging to the initial pool P, and M is the size of P. The aim of dynamic ensemble selection is to find a subensemble $P' \subseteq P$ that is composed of the proper set of predictors to predict a specific test sample. Figure 2 shows an overview of a dynamic predictors selection system. The details about the proposed algorithms are presented in the following subsections.

The research of this paper is mainly focused on the DES paradigm that allows for dynamically selecting ensemble members for each forecast rather than the training of each predictor. However, for completeness and continuity, the generation of an ELMs ensemble is introduced briefly as follows.

Let $\boldsymbol{x}_t = \{\alpha_{t+1}, \alpha_{t+2}, \ldots, \alpha_{t+m}\}$ be the input values of the time series, and $y_t = \alpha_{t+m+1}$ be the target value of the time series, where m denotes the size of time window. Let $g(x)$ denote the activation function, such as sigmoid function in Eq. (21), sine function in Eq. (22), hard limit transfer function in Eq. (23), triangular basis transfer function in Eq. (24), and radial basis transfer function in Eq. (25). Let $\tilde{N} \in [1, 20]$ be the range of the number of hidden nodes. Although when the number of hidden nodes increases, the predictive accuracy on the training set will be improved, however, this is prone to the over-fitting problem. We found by trial-and-error that it is appropriate to set the range of the numbers of hidden nodes as [1, 20].

$$g(x) = \frac{1}{1+e^{-x}} \tag{21}$$

$$g(x) = sin(x) \tag{22}$$

$$g(x) = \begin{cases} 1, if \;\; x > 0; \\ 0, otherwise. \end{cases} \tag{23}$$

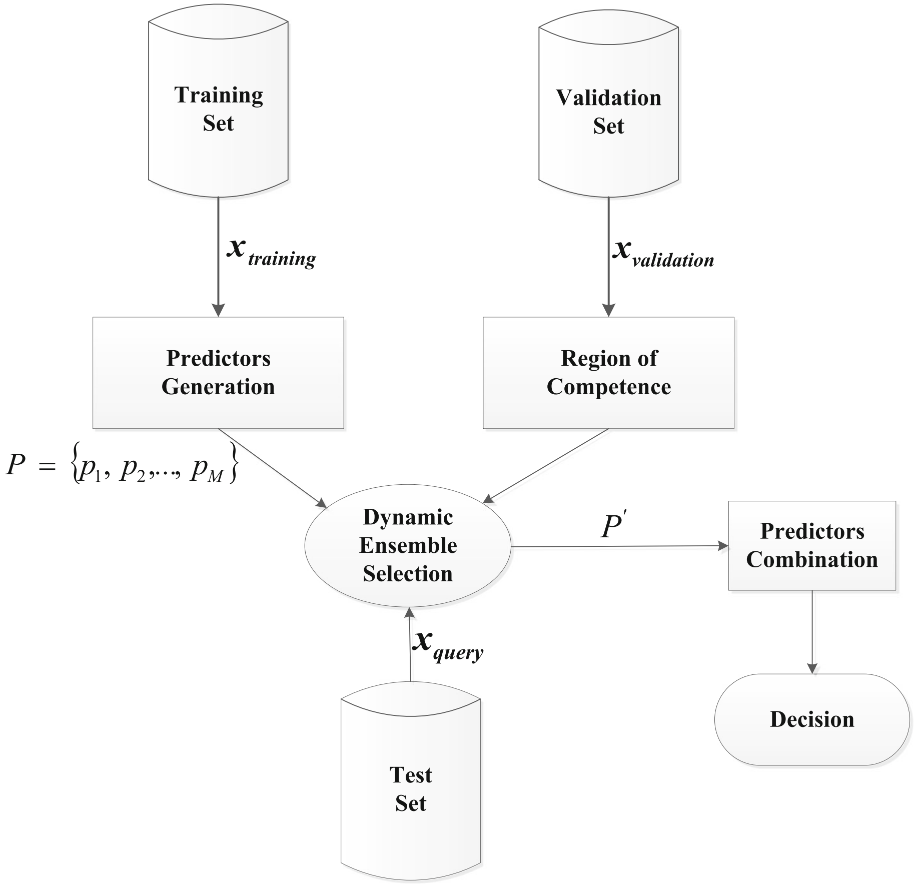


Fig. 2 Overview of a dynamic predictors selection system

$$g(x) = \begin{cases} 1 - |x|, if \quad -1 \leq x \leq 1; \\ 0, \qquad otherwise. \end{cases} \tag{24}$$

$$g(x) = e^{-x^2} \tag{25}$$

One hundred ELMs are generated based on the above five kinds of activation functions, with the number of hidden nodes varying from one to twenty. Let $P = \{p_1, p_2, \ldots, p_M\}$ denote the pool of generated ELM predictors, the proposed algorithms are introduced in detail in the following.

3.1 DES Algorithm Based on Predictor Accuracy over the Local Region (DES-PALR)

The first algorithm proposed in this work is the DES algorithm based on Predictor Accuracy over the Local Region (DES-PALR). Predictor accuracy is the most common criterion for

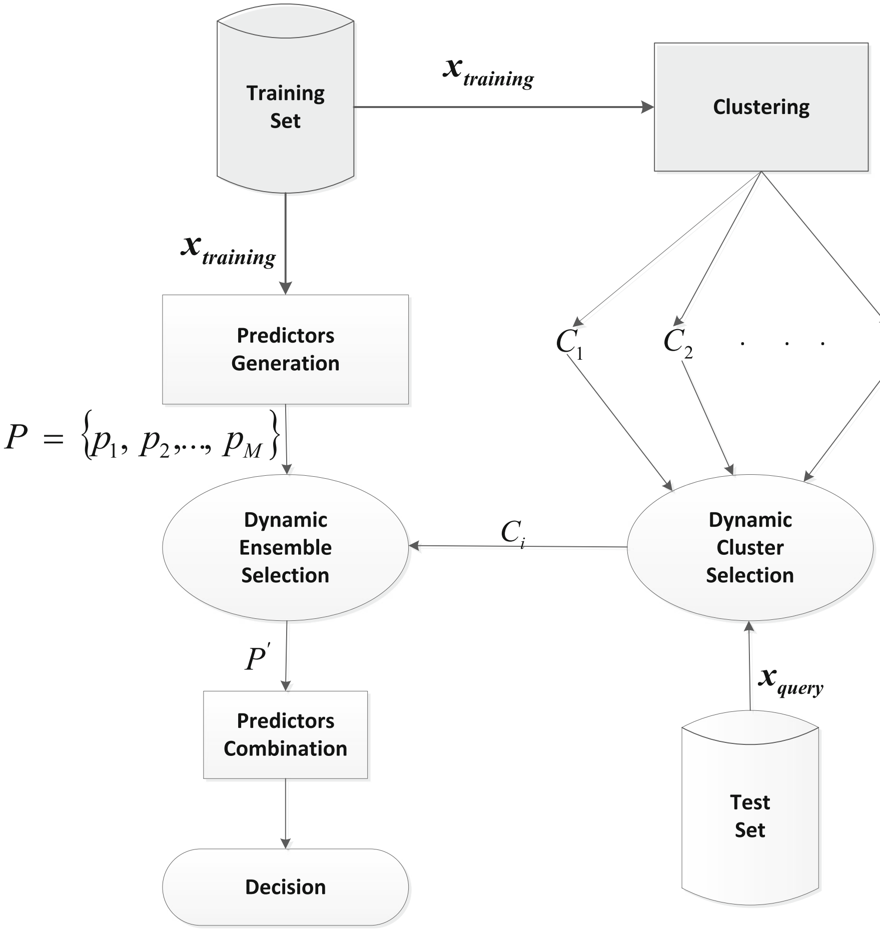


Fig. 3 Overview of the DES-PALR algorithm

the implementation of DES [24, 35, 36]. For this criterion, a small region in the training data surrounding the given testing instance $(\boldsymbol{x}_j, y_j)$ is defined. The region can be computed using the K-NN algorithm [24, 35], or can be computed by using other clustering algorithms [26, 36]. The samples in this region constitute the validation dataset for DES. Figure 3 shows an overview of this algorithm.

The proposed DES-PALR algorithm is shown as follows:

Algorithm 1 The DES algorithm based on Predictor Accuracy over the Local Region (DES-PALR)

```
Input:
    $(\boldsymbol{x}_j, y_j)$— the new testing sample;
    $P$— the pool of $M$ base predictors;
    $Tr$— the training set;
Output:
    $P^{'}$— the ensemble of best predictors found based on the validation set;
1, Perform the k-means clustering algorithm on the feature space of training set, denote the outcome clustering result as $C = \{c_1, c_2, \ldots, c_k\}$ and the set of centers of each cluster as $U = \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_k\}$;
2, Determine $c_{i^*}$ according to $i^* = \underset{1 \le i \le k}{argmin} \|\boldsymbol{x}_j - \boldsymbol{\mu}_i\|$ with respect to $(\boldsymbol{x}_j, y_j)$;
3, Choose all the samples in $c_{i^*}$ as the validation set;
4, $Z$ predictors are selected from $P$ according to the corresponding root mean square error on the validation set, and the set of the selected $Z$ predictors is denote as $P^{'}$ ($P^{'} \subseteq P$);
5, Return $P^{'}$;
```

The time complexities of DES-PALR algorithm in training and testing phase are $O(M + tkNn)$ and $O(k + MlogZ)$, respectively, where M represents the size of initial pool of predictors, t represents the number of iterations of k-means algorithm, k denotes the number of clusters of k-means algorithm, N represents the size of training set, n denotes the dimension of training samples, and Z represents the number of selected predictors.

Before the explanation of the proposed DES-PALR algorithm, we would like to introduce the k-means clustering algorithm briefly [37, 38]. The k-means clustering algorithm is one of the most widely used clustering algorithms. In k-means algorithm, the center of each class is computed as the average of samples belonging to that cluster, which well reflects the geometric and statistical significance of the clustering, with the computational complexity being $O(n)$ [37, 38]. A key problem of k-means algorithm lies in that, the number of clusters is required to be set in advance. However, for the time series problems of great majority, it is difficult to determine the number of clusters.

In [39], the clusters number is determined by optimizing a clustering validity function, which is defined by the ratio of scatter between-class to scatter within-class. This method is called the rule of variance ratio criterion (VRC):

$$max \frac{tr[S_B]/(K-1)}{tr[S_W]/(l-K)}, \tag{26}$$

where l is the samples number, $tr[S_B]$ denotes the trace of scatter between-class matrix and $tr[S_W]$ denotes the trace of scatter within-class matrix. VRC well embodies the compactness and separability of clustering results, and is closely related to the number of clusters. Therefore, VRC is widely used to measure the appropriateness of the number of clusters. Here, we take this criterion to determine K_{opt} ($K_{opt} \leq \sqrt{l}$), similar as in [40]. When the rule of VRC is utilized to determine the number of clusters, it is required to check all of the possible numbers of clusters. Therefore, a reasonable search scope to the clusters numbers is required to be set, so as to reduce the amount of calculation. The greater l is, the greater the search scope becomes. Moreover, when K_{opt} is too large, the samples number in each class becomes very small, which will cause the decline of the model generalization ability. Therefore, in this work by trial-and-error, K_{opt} is set to values between 2 and 4, i.e., $2 \leq K_{opt} \leq 4$.

When K_{opt} is determined, and the training set has been clustered into K_{opt} clusters, c_{i^*}, whose center is the nearest to the current test sample $\left(\boldsymbol{x}_j, y_j\right)$, will be determined. The constituent samples in c_{i^*} are defined as the region of competence for this specific test sample. Based on the region of competence, the local predictor accuracy of a base predictor is evaluated. The predictor possessing the highest predictive accuracy is considered the most competent one. Because the local region has more similar distribution with the testing sample than other samples in the training datasets, therefore, the predictors which have better performance on the local region could perform better on the testing sample than other predictors, and will be selected into the final ensemble. What's more, different testing sample locates different region of competence, yielding effective realization of dynamic ensemble selection.

The main issue with DES-PALR arises from the fact that it depends on the performance of the technique that defines the region of competence, such as K-NN algorithm or other clustering techniques. The algorithm is inclined to commit errors when outlier instances exist around the testing sample. Only using the local accuracy information alone is not sufficient to achieve results close to the Oracle. Moreover, any difference between the distribution of validation and test datasets may negatively affect the DES-PALR algorithm performance. Consequently, more information should be considered during dynamic ensemble selection for its successful application on time series prediction.

3.2 A Group of Three Novel Algorithms for Dynamic Validation Set Determination for TSP

The determination of an appropriate validation set is crucial for the performance of DES algorithms. In this section, a group of three novel algorithms for selecting validation set dynamically and effectively is proposed.

First of all, the training set is denoted as $Tr = \{(\boldsymbol{x}_t, y_t)|\boldsymbol{x}_t = \{\alpha_{t+1}, \ldots, \alpha_{t+m}\}, y_t = \alpha_{t+m+1}\}$. And the output profile of the test instance $\left(\boldsymbol{x}_j, y_j\right)$ is denoted as $\tilde{\boldsymbol{y}}_j = \left(\tilde{y}_{j,1}, \tilde{y}_{j,2}, \ldots, \tilde{y}_{j,M}\right)$, where $\tilde{y}_{j,i}$ is the predictive decision yielded by the base predictor p_i for the sample $\left(\boldsymbol{x}_j, y_j\right)$.

3.2.1 The Dynamic Validation Set Determination Algorithm Based on the Similarity Between the Predictive Value of the Test Sample and the Objective Values of the Training Samples (DVS-PvOv)

In this algorithm, specifically, the dynamic validation set is determined based on the similarity between the predictive value of the test sample and the objective values of the training samples. We term this algorithm as DVS-PvOv, for short. All the training samples $(\boldsymbol{x}_{t^*.i}, y_{t^*.i})$ $(1 \leq i \leq M)$ satisfying Eq. (27) are selected into the validation set.

$$t^* = \underset{1 \leq t \leq N}{argmin} \left| y_{t.i} - \tilde{y}_{j,i} \right|, \quad 1 \leq i \leq M \tag{27}$$

The algorithm can be described as follows:

```
Algorithm 2 The Dynamic Validation Set determination algorithm based on the similarity between the Predictive value of the test sample and the Objective values of the training samples (DVS-PvOv)
Input:
    $(\boldsymbol{x}_j, y_j)$— the testing sample;
    $P$— the pool of $M$ base predictors;
    $Tr$— the training set;
Output:
    $P'$— the ensemble of best predictors found based on the validation set;
1, Perform the $M$ base predictors on the testing sample $(\boldsymbol{x}_j, y_j)$ and generate the output profile $\tilde{\boldsymbol{y}}_j = (\tilde{y}_{j,1}, \tilde{y}_{j,2},\ldots, \tilde{y}_{j,M})$;
2, Select the training sample whose target value is most similar to each element $\tilde{y}_{j.i}$, $1 \le i \le M$ in $\tilde{\boldsymbol{y}}_j$ in turn according to Eq.(27) to constitute the validation set;
3, Select $Z$ predictors that have better performance on validation set from $P$ to constitute $P'$;
4, Return $P'$.
```

The time complexities of DVS-PvOv algorithm in its training and testing phase are $O(M)$ and $O(M + MN + MlogZ)$, respectively, where M represents the size of initial pool of predictors, N represents the size of training set, and Z represents the number of selected predictors.

3.2.2 The Dynamic Validation Set Determination Algorithm Based on the Similarity Between the Newly Constituted Sample for the Test Sample and All the Training Samples (DVS-NsAs)

This algorithm determines the validation set not only based on the predictive value, but also based on the input values of the testing sample. A completely new sample $(\boldsymbol{x}_j, \tilde{y}_{j,i})$ is constituted by $\tilde{y}_{j,i}$ and the input vector $\boldsymbol{x}_j$ of the testing sample $(\boldsymbol{x}_j, y_j)$. In this algorithm, in particular, the dynamic validation set is determined based on the similarity between the newly constituted sample $(\boldsymbol{x}_j, \tilde{y}_{j,i})$ for the test sample and all the training samples. Therefore, we term this algorithm as DVS-NsAs, for short. All the training samples $(\boldsymbol{x}_{t^*.i}, y_{t^*.i})$ $(1 \le i \le M)$ satisfying Eq. (28) are selected into the validation set.

$$t^* = \underset{1\le t\le N}{argmin}\sqrt{\left(\boldsymbol{x}_{t.i} - \boldsymbol{x}_j, y_{t.i} - \tilde{y}_{j,i}\right)\left(\boldsymbol{x}_{t.i} - \boldsymbol{x}_j, y_{t.i} - \tilde{y}_{j,i}\right)^T}, \quad 1 \le i \le M \tag{28}$$

The DVS-NsAs algorithm is showed as follows:

Algorithm 3 The **D**ynamic **V**alidation **S**et determination algorithm based on the similarity between the **N**ewly constituted **s**ample for the test sample and **A**ll the training **s**amples (**DVS-NsAs**)

```
Input:
    $(\boldsymbol{x}_j, y_j)$— the testing sample;
    $P$— the pool of $M$ base predictors;
    $Tr$— the training set;
Output:
    $P'$— the ensemble of best predictors found based on the validation set;
1, Perform M base predictors on the testing sample $(\boldsymbol{x}_j, y_j)$ and generate the output profile $\tilde{\boldsymbol{y}}_j = (\tilde{y}_{j,1}, \tilde{y}_{j,2},..., \tilde{y}_{j,M})$;
2, Construct a completely new sample $(\boldsymbol{x}_j, \tilde{y}_{j,i})$ using the input values of the testing sample and $\tilde{y}_{j,i}$;
3, Select the most similar sample to $(\boldsymbol{x}_j, \tilde{y}_{j,i})$ from the training set to join the validation set;
4, Evaluate the performance of each base predictor on the validation set and select $Z$ best predictors to construct the final ensemble $P'$;
5, Return $P'$.
```

The time complexity of DVS-NsAs algorithm is $O(M)$ in its training phase and $O(M + MN + MlogZ)$ in its testing phase, where M represents the size of initial pool of predictors, N represents the size of training set, and Z represents the number of selected predictors.

The peculiarity of the DVS-NsAs algorithm lies in: it establishes a relationship between the predictive value of every predictor and the input vector of the testing sample. The DVS-NsAs algorithm selects validation samples, not only depend on the characteristic of current test sample but also take the special field in which each predictor is proficient into consideration. Hence, the predictors that possess the better performance than others on the selected validation samples could have higher predictive accuracy.

3.2.3 The Dynamic Validation Set Determination Algorithm Based on the Similarity Between the Output Profile of the Test Sample and the Output Profile of Each Training Sample (DVS-OpOp)

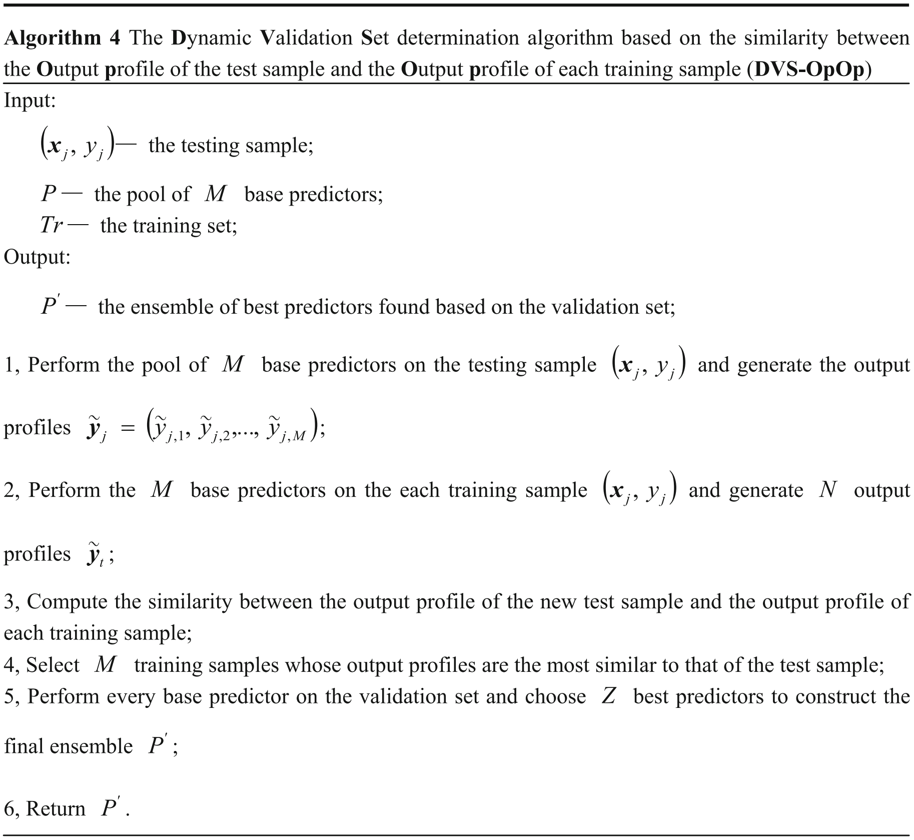
This algorithm dynamically determines the validation set based on the similarity between the output profile of the specific test sample and the output profile of each training sample, which is abbreviated as DVS-OpOp. In this algorithm, the set of output profiles of all the training samples are computed, which is denoted as $Y = \{\tilde{\boldsymbol{y}}_t\}_{t=1}^{N}$, where $\tilde{\boldsymbol{y}}_t = (\tilde{y}_{t,1}, \tilde{y}_{t,2}, \ldots, \tilde{y}_{t,M})$.

The similarity is evaluated based on the Euclidean distance between the two corresponding output profiles, computed as below:

$$dist(t,j) = \sqrt{\left(\tilde{\boldsymbol{y}}_t - \tilde{\boldsymbol{y}}_j\right)\left(\tilde{\boldsymbol{y}}_t - \tilde{\boldsymbol{y}}_j\right)^T}, \quad 1 \le t \le N \tag{29}$$

All the training samples are ranked according to the Euclidean distances between their output profiles with the output profile of the specific test sample, and an appropriate number of top-ranking training samples are selected into the validation set.

The algorithm is showed in detail as follows:



Algorithm 4 The **D**ynamic **V**alidation **S**et determination algorithm based on the similarity between the **O**utput **p**rofile of the test sample and the **O**utput **p**rofile of each training sample (**DVS-OpOp**)

```
Input:
    $(\boldsymbol{x}_j, y_j)$— the testing sample;
    $P$— the pool of $M$ base predictors;
    $Tr$— the training set;
Output:
    $P'$— the ensemble of best predictors found based on the validation set;
1, Perform the pool of $M$ base predictors on the testing sample $(\boldsymbol{x}_j, y_j)$ and generate the output profiles $\tilde{\boldsymbol{y}}_j = (\tilde{y}_{j,1}, \tilde{y}_{j,2},..., \tilde{y}_{j,M})$;
2, Perform the $M$ base predictors on the each training sample $(\boldsymbol{x}_j, y_j)$ and generate $N$ output profiles $\tilde{\boldsymbol{y}}_t$;
3, Compute the similarity between the output profile of the new test sample and the output profile of each training sample;
4, Select $M$ training samples whose output profiles are the most similar to that of the test sample;
5, Perform every base predictor on the validation set and choose $Z$ best predictors to construct the final ensemble $P'$;
6, Return $P'$.
```

The time complexities of DVS-OpOp algorithm in the training and testing phase are $O(M)$ and $O(M + MN + NlogM + MlogZ)$, respectively, where M represents the size of initial pool of predictors, N represents the size of training set, and Z represents the number of selected predictors.

The aim of the proposed DVS-OpOp is also to select samples that are close to the test sample. However, the similarity is computed over the decision space through the concept of decision templates rather than feature space.

The proposed three new algorithms, i.e., DVS-PvOv, DVS-NsAs and DVS-OpOp, give impetus to the effective realization of DES for TSP further. Their common advantage lies in that, they are not limited by the quality of the local region of competence defined solely in the feature space.

3.3 Dynamic Ensemble Selection Based on Consensus of Predictors (DES-CP)

The fifth DES algorithm put forward by us is the DES-CP algorithm. It is designed based on the extent of consensus of the predictors, and it works by considering a pool of ensembles of predictors (EoPs) rather than a pool of predictors.

Firstly, a population of ensembles of predictors should be generated, i.e., $P^* = \left\{P_1', P_2', \ldots, P_n'\right\}$, where n is the number of the ensembles of predictors, and $P_m' \subseteq P$, $m = 1, \ldots, n$. The method employed to generate P^* can be an optimization algorithm, such as genetic algorithm or greedy search methods [41, 42]. Then, for each test sample $\left(\boldsymbol{x}_j, y_j\right)$, P_{con}', which has the highest consensus, is chosen as the final ensemble of predictors. In DES-CP, the extent of consensus among the base predictors of an ensemble is regarded as its level of competence. Figure 4 shows an overview of the DES-CP algorithm.

The difficulty of the DES-CP algorithm lies in how to generate the appropriate $P^* = \left\{P_1', P_2', \ldots, P_n'\right\}$ and make sure the quality of each $P_m^{'}$. In the following, Zhou et al.'s analyses are presented [28].

Assume that each individual predictor has been assigned an optimum weight that exhibits its importance in the ensemble. Then the predictors whose weights are bigger than a pre-set threshold λ are selected to constitute the final ensemble. And λ is set to $1/M$. The weight of the i-th predictor is denoted as w_i, which should satisfy both Eqs. (10) and (11). A weights vector is built as $\boldsymbol{w} = (w_1, w_2, \ldots, w_M)$. Since the best weights should minimize the generalization error of the new ensemble, considering Eq. (18), the best weights vector $\boldsymbol{w_{opt}}$ is expressed as:

$$\boldsymbol{w}_{opt} = argmin\left(\sum_{i=1}^{M}\sum_{j=1}^{M}\boldsymbol{w}_i\boldsymbol{w}_j C_{ij}\right) \tag{30}$$

$w_{opt,k}$, i.e. the k-th (k = 1, 2, … , M) component of $\boldsymbol{w_{opt}}$, can be solved by *lagrange* multiplier. $w_{opt,k}$ satisfies:

$$\frac{\partial\left(\sum_{i=1}^{M}\sum_{j=1}^{M} w_i w_j C_{ij} - 2\lambda\left(\sum_{i=1}^{M} w_i - 1\right)\right)}{\partial w_{opt,k}} = 0 \tag{31}$$

Equation (31) can be simplified as:

$$\sum_{\substack{j=1\\ j\neq k}}^{M} w_{opt,k} C_{kj} = \lambda \tag{32}$$

Considering that $w_{opt,k}$ satisfies Eq. (11), we get:

$$w_{opt,k} = \frac{\sum_{j=1}^{M} C_{kj}^{-1}}{\sum_{i=1}^{M}\sum_{j=1}^{M} C_{ij}^{-1}} \tag{33}$$

Though Eq. (33) is enough to solve $\boldsymbol{w_{opt}}$ in theory, it rarely works well in real-world applications. Neither does it work well in this work, because many ELM models have homogeneous performance in most tasks, which makes the correlation matrix $\left(C_{ij}\right)_{M\times M}$ of the ensemble become irreversible or ill-conditioned. In such circumstances, Eq. (33) cannot be solved directly.

Since Eq. (30) can be viewed as an optimization problem, considering the success that has been obtained by genetic algorithms in optimization area [43], the Genetic Algorithm based Selective ENsemble (GASEN) algorithm proposed by Zhou et al. [28] is utilized in this

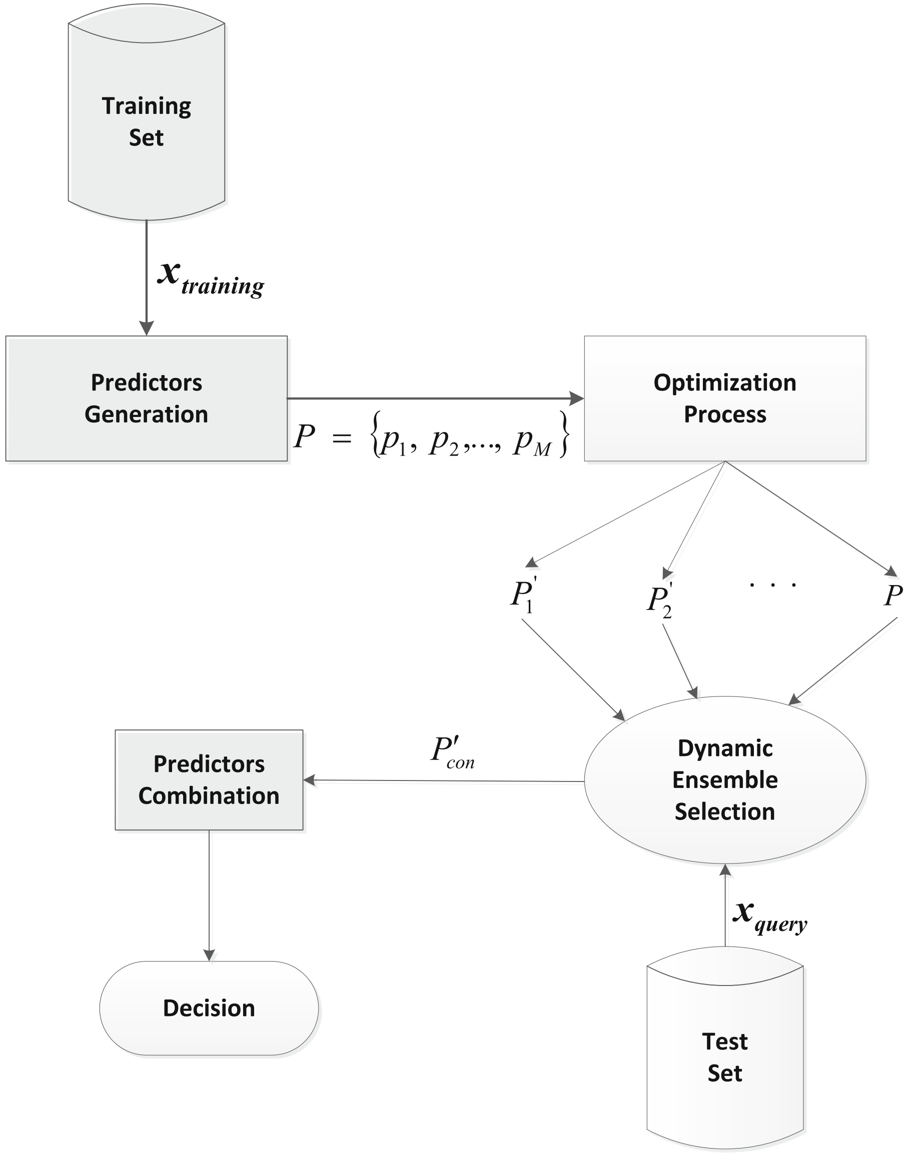


Fig. 4 Overview of the DES-CP algorithm

work. GASEN is used for finding out the subsets of P, which employs the standard genetic algorithm (GA) to evolve the optimum weight vector $\boldsymbol{w_{opt}}$ [43]. Next, the base ELMs, whose corresponding optimum weights are bigger than λ, are chosen to constitute $P_m^{'}$. We denote the validation set as V, the estimated value of the correlation between the i-th ELM and j-th ELM is computed as:

$$C_{ij}^{V} = \frac{\sum_{\boldsymbol{x}_t \in V} (f_i(\boldsymbol{x}_t) - y_t)\big(f_j(\boldsymbol{x}_t) - y_t\big)}{|V|} \tag{34}$$

The estimated generalization error of the ELMs ensemble corresponding to the base weights vector $\boldsymbol{w}$ in the evolving population is:

$$E_w^V = \sum_{i=1}^{M}\sum_{j=1}^{M} w_i w_j C_{ij}^V = \boldsymbol{w}^T C^V \boldsymbol{w} \tag{35}$$

Equation (35) shows the goodness of $\boldsymbol{w}$. The smaller the $E_{\boldsymbol{w}}^V$ is, the better quality the $\boldsymbol{w}$ possesses. So the standard GA uses $f(\boldsymbol{w}) = 1/E_{\boldsymbol{w}}^V$ as the fitness function. And the evolved optimum $\boldsymbol{w}$ is required to be normalized, so that its components can be compared with λ. A simple normalization scheme is used here:

$$w_{opt.i} = w_i \Bigg/ \sum_{i=1}^{M} w_i \tag{36}$$

What's more, GASEN has randomness of itself and initializes $\boldsymbol{w}$ randomly for every run, so it will generate a different pruning ensemble $P_m^{'}$ for each time. The difference mainly lies in that the number of the selected base models is different. And even though the numbers are the same, it cannot be guaranteed that the selected base models remain completely same.

The GASEN algorithm is performed n times to generate $P^* = \{P_1', P_2', \ldots, P_n'\}$. When a new test sample is to be predicted, the consensus of each ensemble of predictors is calculated, and then the ensemble of predictors which possesses the highest consensus is chosen as the final ensemble.

In regard to the problem of how to evaluate the consensus of an ensemble, two methods are designed. Let $P_m' = (p_{m.1}', p_{m.2}', \ldots, p_{m.s}')$ denote the m-th ensemble of selected predictors, and s denotes the size of the m-th ensemble $P_m^{'}$. $P_m'(\boldsymbol{x}_j) = (p_{m.1}'(\boldsymbol{x}_j), p_{m.2}'(x_j), \ldots, p_{m.s}'(\boldsymbol{x}_j))$ denote the predicted values of all the predictor in the m-th ensemble. One method of evaluating the consensus of the m-th ensemble is to calculate the variance of the predicted values made by its constituent predictors. The algorithm developed in this way is termed **D**ynamic **E**nsemble **S**election based on **C**onsensus of **P**redictors evaluated with predictions **Var**iance (DES-CP-Var), which is calculated as below:

$$var\left(P_m^{'}(\boldsymbol{x}_j)\right) = \frac{\sum_{i=1}^{s}\left(p_{m.i}^{'}(\boldsymbol{x}_j) - \frac{\sum_{a=1}^{s} p_{m.a}^{'}(\boldsymbol{x}_j)}{s}\right)}{s} \tag{37}$$

The smaller the variance is, the higher the consensus will be.

Another method which called Dynamic Ensemble Selection based on Consensus of Predictors evaluated with Clustering (DES-CP-Clustering) is also proposed as another specific implementation for the DES-CP algorithm. In this method, k-means clustering is performed on $P_m^{'}(\boldsymbol{x}_j)$, and all the predicted values are divided into k clusters. And the criterion is originally designed as the margin between the size of the cluster (s_1) containing the most predicted values and the size of cluster (s_2) containing the second most predicted values, i.e., the margin is calculated as $\Gamma = |s_1 - s_2|$, originally. However, GASEN is used here for the generation of P^*. The size of each generated ensemble is uncertain, which should also be taken into account. Therefore, the margin is calculated as $\Gamma = \frac{|s_1 - s_2|}{s}$. And under this definition, the bigger the value of Γ becomes, the higher the consensus is.

The entire algorithm is displayed as follows:

Algorithm 5 Dynamic Ensemble Selection based on the Consensus of Predictors (DES-CP)

```
Input:
    $(x_j, y_j)$ — the testing sample;
    $P$ — the pool of $M$ base predictors;
    $Va$ — the validating set;
Output:
    $P'$ — the best subensemble determined with DES-CP;
1, Run the GASEN algorithm $n$ times and produce $P^* = \{P_1', P_2', \ldots, P_n'\}$;
2, Calculate the consensus of every subensemble;
3, Choose the subensemble $P_m'$ possessing the highest consensus as the final ensemble $P'$;
4, Return $P'$;
```

$O(ngM^2(r+\frac{1}{2}r+rf)$ and O($\sum_{i=1}^{n}\left|P_i'\right|$) are the time complexities of DES-CP algorithm in its training and testing phase, respectively, where n denotes the size of P^*, M represents the size of initial pool of predictors, N represents the size of training set, Z denotes the number of selected predictors, g is the number of generations, r represents the size of population, and f denotes the length of chromosome.

The standpoint is that the higher the consensus among the constituent predictors is, the higher the level of confidence in the decision will be. Consequently, by maximizing the extent of consensus of an ensemble, the degree of certainty that it will make a better prediction is increased.

The main advantage of this technique stems from the fact that it does not require information from the region of competence. Therefore, it does not suffer from the limitation of the algorithm which defines the region of competence.

4 Empirical Analysis and Evaluation

4.1 Experimental Data and Data Pre-processing

A total of twelve benchmark datasets from Time Series Data Library [44] are conducted in comparative experiments. DES-PALR, DVS-PvOv, DVS-NsAs, DVS-OpOp, DES-CP-Var and DES-CP-Clustering models are implemented on Dow-Jones Industrial Average (DJI), St. Louis Fed Financial Stress index (STLFSI), Odonovan, Montgome, M3-U.S (MUS), Wolf River at New London (WRNL), Clearwater River at Kamiah (CRK), Mean monthly Flow in piper's hole River (MFR), Exchange rate of Australian dollar: $A for 1US dollar (EAFUS), Annual Copper Prices (ACP), Mean Annual Nile Flow (MANF), U.K. Deaths from Bronchitis, Emphysema and Asthma (UKDBEA). The concrete scale of each dataset is shown respective in Table 2.

Since the attributes of sample sets have different ranges, it is necessary to adjust the value domain of each attribute into the range between 0 and 1. This ensures that the input

Table 2 Key features of the datasets conducted in our experiments

Dataset	Metrics	Time granularity	Category
DJI	4270 fact values	Date	Finance
STLFSI	1026 fact values	Week	Finance
Odonovan	70 fact values	Time	Chemistry
Montgome	100 fact values	Time	Chemistry
MUS	398 fact values	Month	Finance
WRNL	564 fact values	Month	Hydrology
CRK	600 fact values	Month	Hydrology
MFR	348 fact values	Month	Hydrology
EAFUS	314 fact values	Month	Finance
ACP	197 fact values	Year	Micro-Economic
MANF	99 fact values	Year	Hydrology
UKDBEA	72 fact values	Month	Health

attributes with larger value do not overwhelm the smaller value inputs, and then helps to reduce prediction errors. The normalization method is presented as follows:

Each of the series value x_t is normalized by the linear interpolation as in Eq. (38):

$$x_t^{new} = \frac{x_t - x^{min}}{x^{max} - x^{min}} \tag{38}$$

where x_t^{new} =normalized value; x_t=value to be normalized; x^{min} =minimum value of the series to be normalized; x^{max} =maximum value of the series to be normalized.

4.2 Experimental Methodology

4.2.1 Performance Measurements

A performance measurement is necessary to appropriately evaluate the predictive performance of the pruned ensemble obtained by using different ensemble pruning techniques. The performance measurements are defined on the basis of prediction errors, which are computed as the difference between the real value of the series and the predicted value, just as below:

$$e_t = (target_t - output_t) \tag{39}$$

where target_t denotes the desired output of the prediction model at time t, and output_t denotes the output of the ensemble at time t. Based on the prediction errors, two performance measurements employed to evaluate the predictive performance of the pruned ensembles are described below.

Root Mean Square Error (RMSE) Root mean square error (RMSE) [45] is the most common metric used to analyze ensembles performance, and it is defined by the equation:

$$\text{RMSE} = \frac{1}{N}\sqrt{\sum_{t=1}^{N} (e_t)^2} \tag{40}$$

where N denotes the number of data values of the testing time series. Obviously, the lower the value of RMSE, the better the prediction result will be. Although RMSE is quite common as a performance measurement, it does not provide complete and convincing evidence about the accuracy obtained by the predictive model. Therefore, another metric of Mean Absolute Error is also used to evaluate the performance of the proposed algorithms.

Mean Absolute Error (MAE) In statistics, mean absolute error (MAE) [46] is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. MAE is defined as:

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N} |e_t| \tag{41}$$

Clearly, the lower the value of MAE, the closer is the desired result from the predicted one.

4.2.2 Experimental Setup

First of all, 100 well-trained ELMs are constructed by changing the type of activation function and the number of hidden neurons, where the former is represented by Eqs. (21–25), and the latter is in the range of [1, 20]. These well-trained ELMs are utilized to form the initial pool of basic models.

Every dataset is spilt into two distinctive parts, i.e., a training set and a testing set, with 70% and 30% of the initial dataset, respectively. With respect to the time window size (TWS), considering the size of the datasets and by trial-and-error, the feasible value is set to 5. Considering the randomness of the input weights and biases, which will lead to instability of ELM, the proposed algorithms are run repeatedly for 20 times. The final performance measurements are obtained by averaging the performances of these 20 rounds.

In our experiments, the proposed algorithms are compared with Static Averaging All (AA), Static Ensemble Selective (SES), GASEN, ELM, Hierarchical Extreme Learning Machine (H-ELM), BP, Deep representations via Extreme Learning Machine (DrELM), Algebraic Prediction External Smoothing (APES) [4], Algebraic Prediction Internal Smoothing (APIS) [4], and Algebraic Prediction Mixed Smoothing (APMS) [4].

4.3 Experimental Results

In this section, the experimental results of RMSE and MAE obtained by the proposed six algorithms, static ensemble selection methods and some other state-of-the-art methods on the twelve benchmark time series datasets are listed out in Tables 3, 4, 5, 6, 7, 8, 9, 10, 11, 12 and 13.

Tables 3, 4, 5 and 6 give the detailed RMSE performance and the ranking results based on RMSE on the twelve time series. From the results, it is obviously shown that, the proposed DES algorithms achieve higher ranks than other state-of-the-art algorithms, i.e., GASEN, AA, SES, ELM, H-ELM, BP and DrELM. At the same time, the algorithms that obtain the best performance on each time series are all the proposed DES algorithms. According to the average ranking results shown in column 3 of Table 11 that, DES-CP-Clustering achieves the best average ranking on the twelve time series based on RMSE, while ELM and BP get the worst ones.

Tables 7, 8, 9 and 10 show the detailed MAE performance and the ranking results based on MAE on the twelve time series. It can be observed from Tables 7, 8, 9 and 10 that, only

Table 3 RMSE and rankings based on RMSE on DJI, STLFSI and Odonovan time series

RMSE	DJI	Ranks	STLFSI	Ranks	Odonovan	Ranks
Dynamic ensemble selection						
DES-PALR	0.0129	2	0.0142	6	0.2579	9
DVS-PvOv	0.0154	7	0.0123	3	0.2500	7
DVS-NsAs	0.0161	9	0.0122	2	0.2351	4
DVS-OpOp	**0.0128**	**1**	0.0234	7	0.2513	8
DES-CP-Var	0.0131	3	**0.0115**	**1**	0.2364	5
DES-CP-Clustering	0.0133	4	0.0130	5	**0.2266**	**1**
Static ensemble selection						
GASEN	0.0144	5	0.0367	9	0.2283	2
AA	0.1028	11	0.0445	10	0.2350	3
SES	0.0184	10	0.0125	4	0.2461	6
The state-of-the-art methods						
ELM	0.2960	13	0.2515	12	0.5991	11
H-ELM	0.0157	8	4.9508	13	1.6485	13
BP	0.2957	12	0.1700	11	0.6401	12
DrELM	0.0147	6	0.0267	8	0.2943	10

Remark The boldface indicates the algorithm which performs the best on each time series

Table 4 RMSE and rankings based on RMSE on Montgome, MUS and WRNL time series

RMSE	Montgome	Ranks	MUS	Ranks	WRNL	Ranks
Dynamic ensemble selection						
DES-PALR	**0.1811**	**1**	0.1367	5	**0.1123**	**1**
DVS-PvOv	0.1965	8	0.1246	4	0.1340	10
DVS-NsAs	0.1885	4	0.1556	6	0.1258	8
DVS-OpOp	0.1882	3	0.2213	8	0.1235	4
DES-CP-Var	0.1917	6	**0.0326**	**1**	0.1242	6
DES-CP-Clustering	0.1902	5	0.0391	3	0.1233	3
Static ensemble selection						
GASEN	0.1997	10	0.2550	11	0.1190	2
AA	0.1932	7	0.1643	7	0.1269	9
SES	0.1968	9	0.0329	2	0.1248	7
The state-of-the-art methods						
ELM	0.5395	12	0.8494	12	0.1843	12
H-ELM	1.3634	13	1.6632	13	0.1475	11
BP	0.4779	11	0.2473	10	0.1844	13
DrELM	0.1864	2	0.2290	9	0.1235	4

Remark The boldface indicates the algorithm which performs the best on each time series

Table 5 RMSE and rankings based on RMSE on CRK, MFR and EAFUS time series

RMSE	CRK	Ranks	MFR	Ranks	EAFUS	Ranks
Dynamic ensemble selection						
DES-PALR	0.1129	5	0.1567	3	0.0300	3
DVS-PvOv	0.1305	9	0.1566	2	**0.0253**	**1**
DVS-NsAs	0.1276	8	0.1670	9	0.0321	4
DVS-OpOp	0.1020	2	**0.1564**	**1**	0.0283	2
DES-CP-Var	0.1156	6	0.1589	5	0.0434	9
DES-CP-Clustering	**0.1072**	**1**	0.1589	5	0.0328	5
Static ensemble selection						
GASEN	0.1095	4	0.1649	8	0.0367	8
AA	0.1366	10	0.1599	7	0.0332	6
SES	0.1085	3	0.1690	10	0.0338	7
The state-of-the-art methods						
ELM	0.2415	12	0.3117	12	0.1813	12
H-ELM	0.1174	7	0.2078	11	0.0456	10
BP	0.3254	13	0.3117	12	0.1814	13
DrELM	0.1989	11	0.1587	4	0.0470	11

Remark The boldface indicates the algorithm which performs the best on each time series

Table 6 RMSE and rankings based on RMSE on ACP, MANF and UKDBEA time series

RMSE	ACP	Ranks	MANF	Ranks	UKDBEA	Ranks
Dynamic ensemble selection						
DES-PALR	0.0663	7	0.1452	8	**0.0758**	**1**
DVS-PvOv	0.0711	10	0.1332	3	0.0898	5
DVS-NsAs	0.0709	9	0.1519	9	0.0763	3
DVS-OpOp	0.0662	6	0.1415	7	0.0761	2
DES-CP-Var	**0.0626**	**1**	0.1308	2	0.0942	6
DES-CP-Clustering	0.0627	3	**0.1297**	**1**	0.0954	7
Static ensemble selection						
GASEN	0.0654	5	0.1382	5	0.0869	4
AA	0.0632	4	0.1388	6	0.0989	8
SES	**0.0626**	**1**	0.1335	4	0.1079	9
The state-of-the-art methods						
ELM	0.4126	12	0.5064	12	0.2356	11
H-ELM	0.0721	11	0.5421	13	0.4606	13
BP	0.4231	13	0.4448	11	0.2390	12
DrELM	0.0666	8	0.2335	10	0.1869	10

Remark The boldface indicates the algorithm which performs the best on each time series

Table 7 MAE and rankings based on MAE on DJI, STLFSI and Odonovan time series

MAE	DJI	Ranks	STLFSI	Ranks	Odonovan	Ranks
Dynamic ensemble selection						
DES-PALR	**0.0093**	**1**	0.0116	6	0.2020	9
DVS-PvOv	0.0109	6	0.0094	3	0.1966	6
DVS-NsAs	0.0126	8	**0.0089**	**1**	**0.1816**	**1**
DVS-OpOp	**0.0093**	**1**	0.0179	8	0.1975	8
DES-CP-Var	0.0096	3	0.0092	2	0.1911	5
DES-CP-Clustering	0.0098	4	0.0106	5	0.1822	2
Static ensemble selection						
GASEN	0.0144	9	0.0284	9	0.1850	3
AA	0.0978	11	0.0396	10	0.1860	4
SES	0.0156	10	0.0098	4	0.1962	7
The state-of-the-art methods						
ELM	0.2687	13	0.1628	12	0.2679	10
H-ELM	0.0114	7	1.1837	13	1.1834	13
BP	0.2685	12	0.1284	11	0.6187	12
DrELM	0.0103	5	0.0127	7	0.2624	11

Remark The boldface indicates the algorithm which performs the best on each time series

Table 8 MAE and rankings based on MAE on Montgome, MUS and WRNL time series

MAE	Montgome	Ranks	MUS	Ranks	WRNL	Ranks
Dynamic ensemble selection						
DES-PALR	0.1563	4	0.0987	5	**0.0756**	**1**
DVS-PvOv	0.1670	9	0.0941	4	0.1033	11
DVS-NsAs	0.1561	3	0.1237	6	0.0832	3
DVS-OpOp	0.1602	7	0.1746	8	0.0840	5
DES-CP-Var	0.1546	2	**0.0237**	**1**	0.0849	8
DES-CP-Clustering	**0.1519**	**1**	0.0312	3	0.0845	7
Static ensemble selection						
GASEN	0.1613	8	0.2271	10	0.0780	2
AA	0.1569	5	0.1499	7	0.0906	9
SES	0.1979	10	0.0293	2	0.0842	6
The state-of-the-art methods						
ELM	0.4912	12	0.7201	12	0.1217	12
H-ELM	0.7695	13	1.2291	13	0.0966	10
BP	0.4286	11	0.1952	9	0.1218	13
DrELM	0.1576	6	0.2273	11	0.0836	4

Remark The boldface indicates the algorithm which performs the best on each time series

Table 9 MAE and rankings based on MAE on CRK, MFR and EAFUS time series

MAE	CRK	Ranks	MFR	Ranks	EAFUS	Ranks
Dynamic ensemble selection						
DES-PALR	0.0731	6	0.1218	2	0.0245	5
DVS-PvOv	0.0854	9	0.1240	3	0.0201	2
DVS-NsAs	0.0735	7	0.1275	7	0.0253	6
DVS-OpOp	0.0634	3	**0.1217**	**1**	0.0228	4
DES-CP-Var	0.0715	5	0.1253	4	0.0390	11
DES-CP-Clustering	**0.0529**	**1**	0.1257	5	0.0273	8
Static ensemble selection						
GASEN	0.0618	2	0.1300	9	0.0295	9
AA	0.0858	10	0.1284	8	0.0270	7
SES	0.0698	4	0.1362	10	**0.0179**	**1**
The state-of-the-art methods						
ELM	0.1513	11	0.2575	12	0.1734	12
H-ELM	0.0776	8	0.1519	11	0.0201	2
BP	0.1988	13	0.2575	12	0.1735	13
DrELM	0.1649	12	0.1267	6	0.0370	10

Remark The boldface indicates the algorithm which performs the best on each time series

Table 10 MAE and rankings based on MAE on ACP, MANF and UKDBEA time series

MAE	ACP	Ranks	MANF	Ranks	UKDBEA	Ranks
Dynamic ensemble selection						
DES-PALR	0.0509	6	0.1144	6	0.0571	2
DVS-PvOv	0.0566	8	0.1109	3	0.0698	5
DVS-NsAs	0.0569	9	0.1198	9	**0.0541**	**1**
DVS-OpOp	0.0503	5	0.1141	5	0.0586	3
DES-CP-Var	**0.0485**	**1**	0.1029	2	0.0803	6
DES-CP-Clustering	0.0491	3	**0.1018**	**1**	0.0809	7
Static ensemble selection						
GASEN	0.0502	10	0.1190	7	0.0661	4
AA	0.0496	4	0.1125	4	0.0860	8
SES	0.0486	2	0.1194	8	0.0929	9
The state-of-the-art methods						
ELM	0.4038	12	0.4892	13	0.1891	11
H-ELM	0.0564	7	0.4064	11	0.2936	13
BP	0.4127	13	0.4306	12	0.1902	12
DrELM	0.0621	11	0.2105	10	0.1661	10

Remark The boldface indicates the algorithm which performs the best on each time series

Table 11 The average ranking of the proposed DES algorithms and the comparative methods on the twelve time series based on RMSE, MAE, and overall

Models	RMSE-ranks	MAE-ranks	Overall-ranks
Dynamic ensemble selection			
DES-PALR	4.2500	4.4167	4.3333
DVS-PvOv	5.7500	5.7500	5.7500
DVS-NsAs	6.2500	5.0833	5.6667
DVS-OpOp	4.2500	4.8333	4.5417
DES-CP-Var	4.2500	4.1667	4.2083
DES-CP-Clustering	**3.5833**	**3.9167**	**3.7500**
Static ensemble selection			
GASEN	6.0833	6.8333	6.4583
AA	7.3333	7.0833	7.2083
SES	6	6.0833	6.0417
The state-of-the-art methods			
ELM	11.9167	11.8333	11.8750
H-ELM	11.3333	10.0833	10.7083
BP	11.9167	11.9167	11.9167
DrELM	7.7500	8.5833	8.1667

Remark The boldface indicates the algorithm which achieves the highest ranking

Table 12 The detailed RMSE performance of APES, APIS and APMS on the first four time series

RMSE	DJI	STLFI	Odonovan	Montgome
APES	0.2934	0.5538	0.8938	0.4471
APIS	0.0507	0.0349	0.1933	0.2112
APMS	0.0485	0.0295	0.2097	0.2309

Table 13 The detailed MAE performance of APES, APIS and APMS on the first four time series

MAE	DJI	STLFI	Odonovan	Montgome
APES	0.1371	0.2171	0.5859	0.3521
APIS	0.0411	0.0229	0.1758	0.1801
APMS	0.0378	0.0199	0.1917	0.1987

on EAFUS time series, SES achieves the best MAE performance. Except for that, on other eleven time series datasets, the algorithms which achieve the best MAE performance are all the proposed DES methods. Table 11 lists out the average ranking of the proposed DES algorithms and the comparative methods on the twelve time series based on MAE. It is clearly shown in column 4 of Table 11 that, the top six are all the proposed six DES algorithms, while BP is the last one.

Column 5 of Table 11 also gives the average comprehensive ranking of the proposed DES algorithms and the comparative methods on the twelve time series based on both RMSE and MAE. The results demonstrate that: (1) the performances of the proposed DES algorithms are obviously better than that of the static ensemble selection methods, especially for DES-CP-Clustering and DES-CP-Var algorithms; (2) ensembles of predictors outperform single models; (3) the two deep neural networks, i.e., H-ELM and DrELM, achieve better performance than the single hidden layer neural networks; (4) ELM is slightly better than BP on the twelve time series datasets.

It can be concluded from Tables 3, 4, 5, 6, 7, 8, 9, 10 and 11 that DES-CP-Var and DES-CP-Clustering achieve more excellent performance than other proposed algorithms on most of the twelve time series datasets, and they obtain the highest overall rankings (3.75 and 4.21) within the six DES algorithms. The reason might be that, DES-CP-Clustering and DES-CP-Var algorithms are designed based on the extent of consensus of the predictors, and they work by considering a pool of EoPs generated by GASEN, rather than a pool of predictors. However, the disadvantage of these two proposed algorithms is time-consuming. Therefore, whether to choose DES-CP-Clustering or DES-CP-Var depends on the training time demanded by the specific systems. If real time response is required, DES-PALR algorithm is recommended, owing to its low time-complexity and good performance.

It can also be concluded from Table 11 that, if the criterion of performance measurement is RMSE, DVS-OpOp algorithm is a good choice (the second best). The reason is that, it costs less time than DES-CP-Clustering and DES-CP-Var algorithms. Meanwhile it is not trapped by the techniques which define the local region. Although the proposed DVS-PvOv and DVS-NsAs algorithms do not achieve better performance than the other DES algorithms proposed by us on the twelve time series datasets, they still obtain higher accuracy than the compared methods. As stated by the principle of "No Free Lunch", no algorithm performs better than any other ones on all the problems. For example, DVS-PvOv algorithm achieves the best RMSE performance on EAFUS time series dataset, while DVS-NsAs algorithm achieves the most superior MAE performance on UKDBEA, STLFSI and Odonovan time series datasets.

Tables 12 and 13 give the detailed RMSE and MAE performance of APES, APIS and APMS on the first four time series datasets, respectively. It can be concluded that, APIS achieves the best RMSE and MAE performance on the Odonovan time series only, outperforming the proposed six DES algorithms. At the same time, APES and APMS do not obtain good performance compared with the proposed DES algorithms.

Next, to ascertain whether the proposed DES algorithms are significantly better than GASEN, SES, AA, ELM, H-ELM, BP and DrELM in a statistic sense, t-tests are implemented to the rankings of all the algorithms obtained on the twelve time series datasets. However, if the rankings of algorithms are not normally distributed, t test may lead to error conclusions. Therefore, we conduct normality tests firstly, with the results shown in Table 14. The built-in function JBTEST of MATLAB is employed to implement normality tests, where the significance level ALPHA was set to 0.01. The values reported in Table 14 are the test statistic JBTEST returned by function JBTEST, where H = 0 indicates that the null hypothesus cannot be rejected at the 1% significance level, and H = 1 indicates that the null hypothesis can be rejected at the 1% level. Null hypothesis is that ranking is normally distributed.

It is clearly shown in Table 14 that, the rankings of almost all algorithms obey normal distribution, with only one exception, i.e., H-ELM.

Then, t-tests are conducted to compare the rankings of the proposed DES algorithms with those of the GASEN, SES, AA, and other state-of-the-art algorithms, with the significance level ALPHA set to 0.05 and TAIL set to left. The results are listed in Table 15. The null hypothesis H = 0 indicates that there is no significant difference between Model A and Model B. The null hypothesis H = 1 indicates that Model A is significantly better than Model B at the 5% significance level (t value ≤ -1.7139).

As shown in Table 15, for 36 of the 42 t tests (85.7%), the proposed DES algorithms achieve significant improvements on the rankings of the comparative approaches at 5% sig-

Table 14 The results of normality tests

Models	Dynamic ensemble selection						Static ensemble selection			The state-of-the-art methods			
	DES-PALR	DVS-PvOv	DVS-NsAs	DVS-OpOp	DES-CP-Var	DES-CP-Clustering	GASEN	AA	SES	ELM	H-ELM	BP	DrELM
H	0	0	0	0	0	0	0	0	0	0	1	0	0

Table 15 Results of *t*-tests applied to the rankings of the proposed DES algorithms with others on the twelve datasets

Model B	Model A											
	DES-CP-Clustering		DES-CP-Var		DVS-NsAs		DES-PALR		DVS-OpOp		DVS-PvOv	
	H	P	H	P	H	P	H	P	H	P	H	P
GASEN	1	3.5237e−04	1	0.0149	0	0.1749	1	0.0103	1	0.0124	0	0.2581
AA	1	1.0675e−06	1	1.7219e−04	1	0.0358	1	0.0025	1	0.0034	1	0.0360
SES	1	0.0025	1	0.0153	0	0.3468	0	0.0593	0	0.0818	0	0.3757
ELM	1	5.0309e−15	1	5.2850e−12	1	1.6646e−11	1	3.4092e−12	1	5.1217e−12	1	8.1840e−10
H-ELM	1	4.6754e−19	1	3.9241e−07	1	2.4295e−05	1	6.1205e−09	1	1.0498e−10	1	4.8636e−06
BP	1	5.7196e-15	1	2.6665e−14	1	1.0954e−10	1	3.3613e−−12	1	6.0252e−11	1	8.4814e-11
DrELM	1	1.1992e−05	1	4.1836e−05	1	0.0038	1	3.3044e−08	1	9.8815e−06	1	0.0087

Remark The null hypothesis H = 0 indicates that there is no significant difference between Model A and Model B. The null hypothesis H = 1 indicates that Model A is significantly better than Model B at the 5% significance level (t value ≤ − 1.7139)

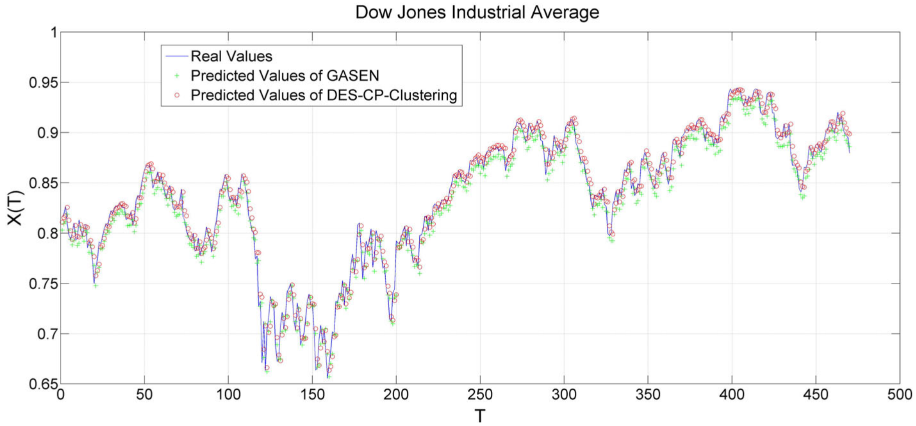


Fig. 5 Dow Jones Industrial Average (prediction values)

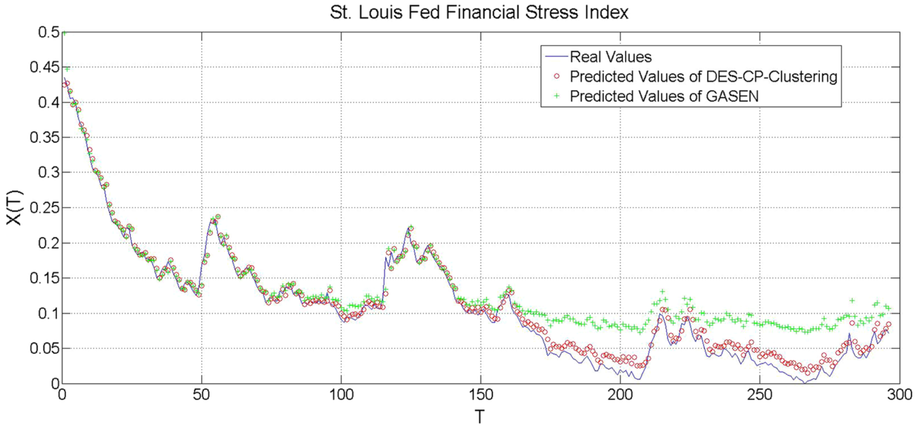


Fig. 6 St. Louis Fed Financial Stress Index (prediction values)

nificance level. These results clearly show that DES-CP-Clustering, and DES-CP-Var are significantly better than the state-of-the-art algorithms. At the same time, the proposed DES algorithms are all significantly better than ELM, H-ELM, BP and DrELM at the 5% significance level. Although it can be obviously concluded from Table 11 that, the average comprehensive of rankings of DES-PLAR, DVS-OpOp, DVS-NsAs and DVS-PvOv are superior to all the comparative algorithms, the performances of the four are not siginificantly better than the comparative algorithms at the 5% significance level, except for ELM, H-ELM, BP and DrELM in Table 15. This phenomenon is easy to understand. Since the seven comparative algorithms are all state-of-the-art algorithms in the literature, the phenomenon that the difference between DES-PLAR, DVS-OpOp, DVS-NaAs, DVS-PvOv and the comparative algorithms especially GASEN and SES are not significant is natural.

Finally, Figs. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 and 16 display the prediction values of one of the proposed DES algorithms which performs the best, i.e., DES-CP-Clustering, and GASEN on the twelve benchmark time series datasets, respectively. The prediction errors

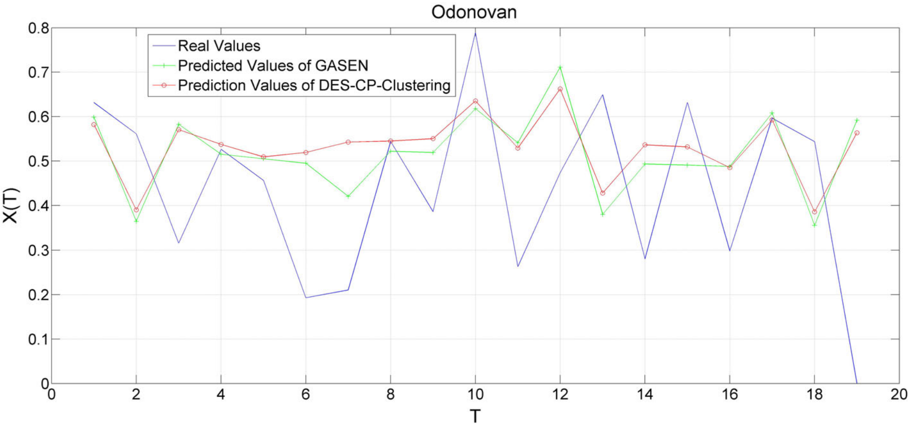


Fig. 7 Odonovan (prediction values)

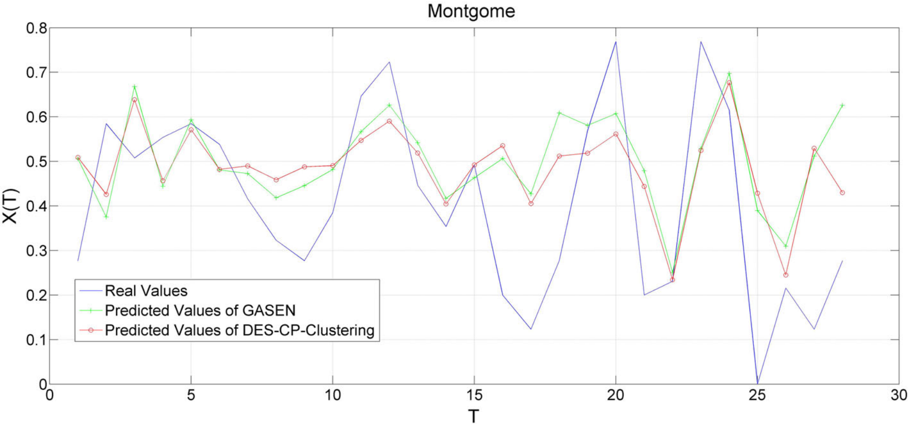


Fig. 8 Montgome (prediction values)

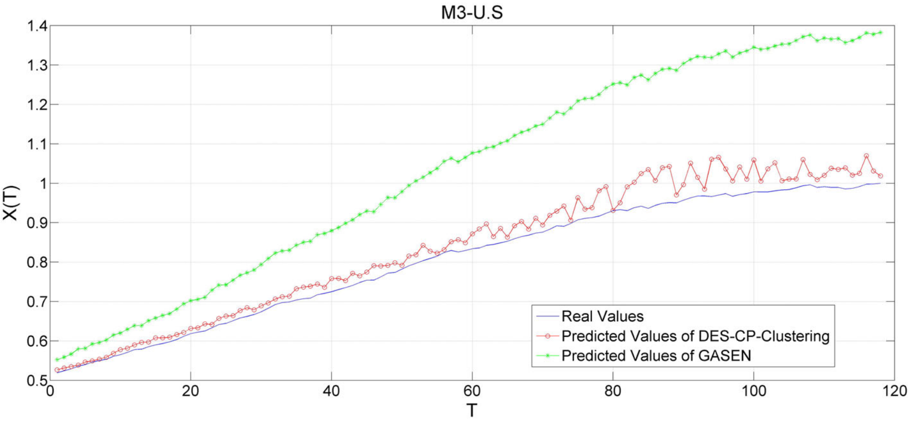


Fig. 9 M3-U.S (prediction values)

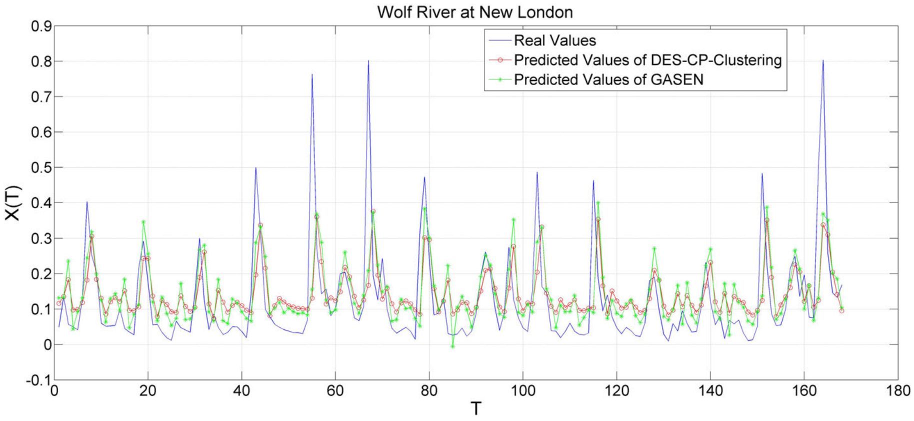


Fig. 10 Wolf River at New London (prediction values)

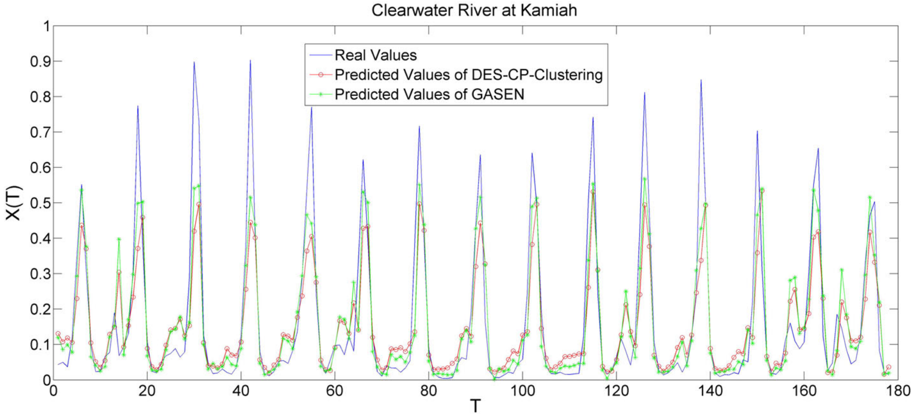


Fig. 11 Clearwater River at Kamiah (prediction values)

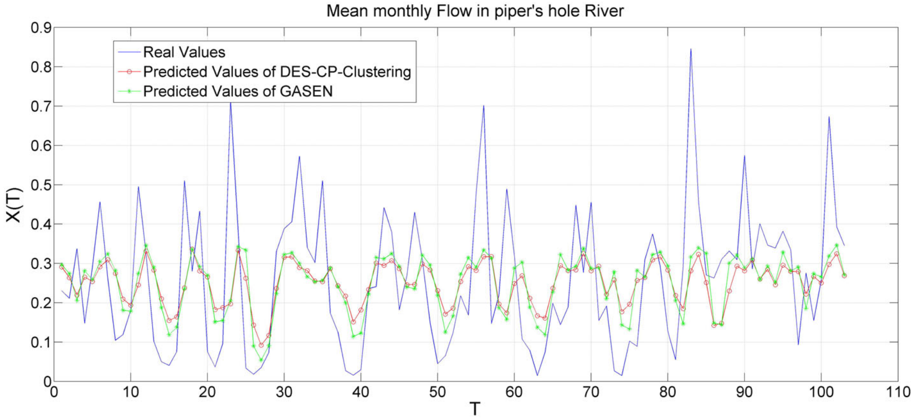


Fig. 12 Mean monthly Flow in piper’s hole River (prediction values)

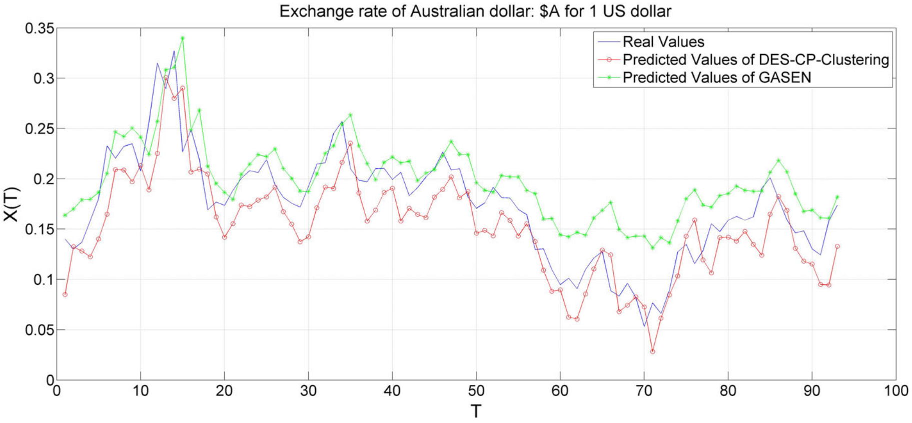


Fig. 13 Exchange rate of Australian dollar: $A for 1 US dollar (prediction values)

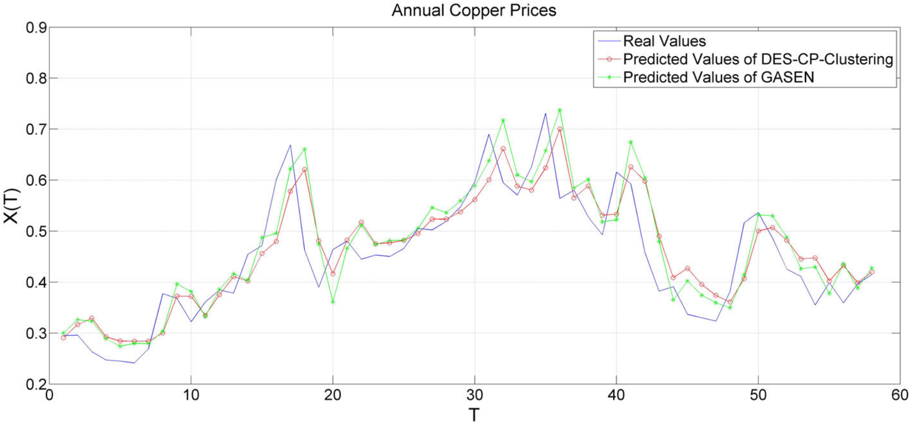


Fig. 14 Annual Copper Prices (prediction values)

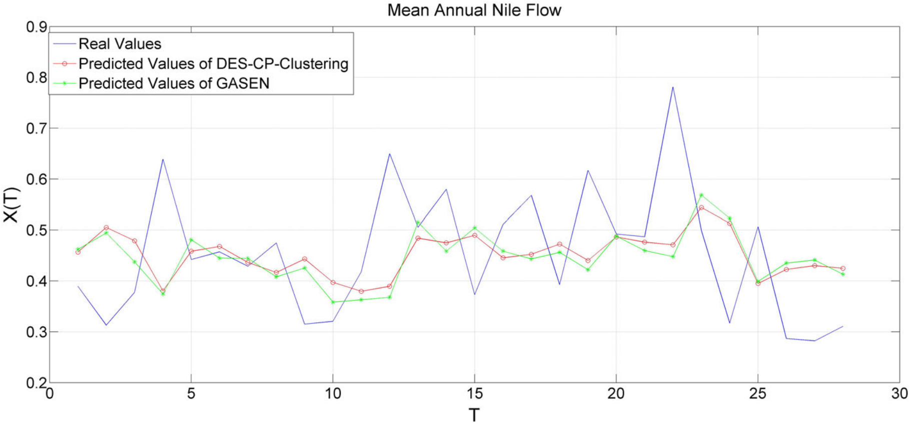


Fig. 15 Mean Annual Nile Flow (prediction values)

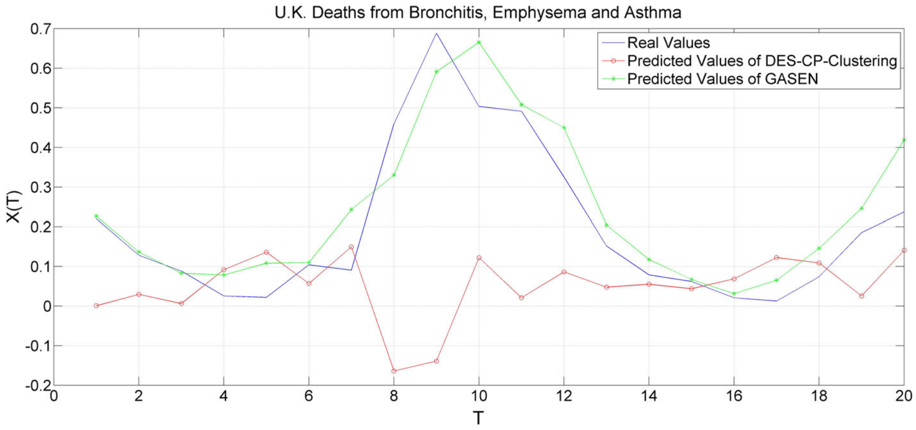


Fig. 16 U.K. Deaths from Bronchitis, Emphysem and Asthma (prediction values)

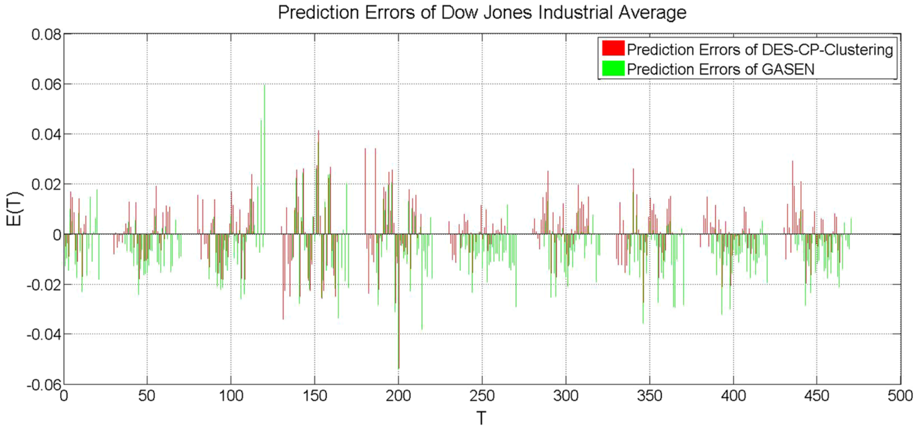


Fig. 17 Dow Jones Industrial Average (prediction errors)

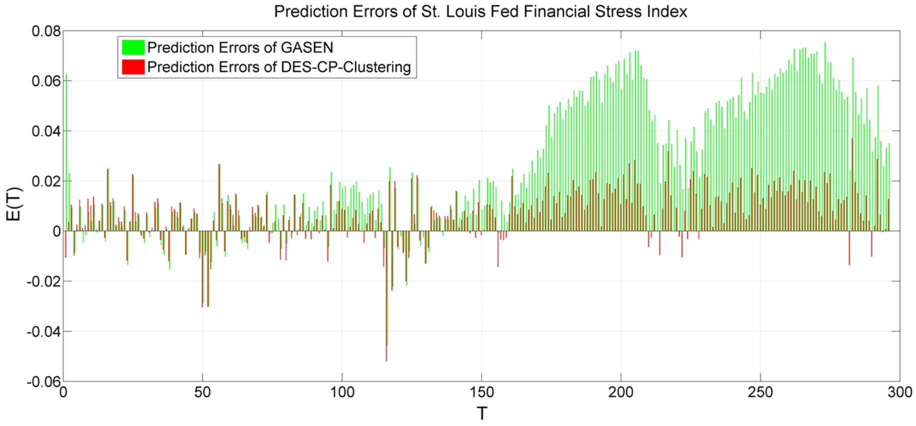


Fig. 18 St. Louis Fed Financial Stress Index (prediction errors)

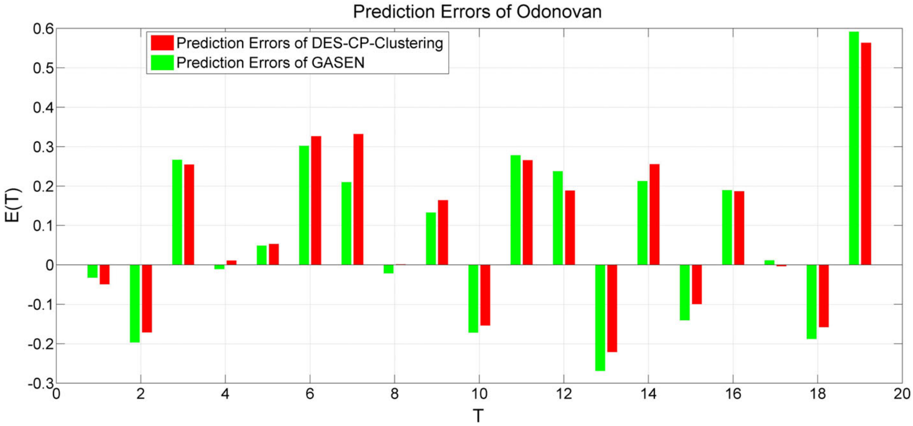


Fig. 19 Odonovan (prediction errors)

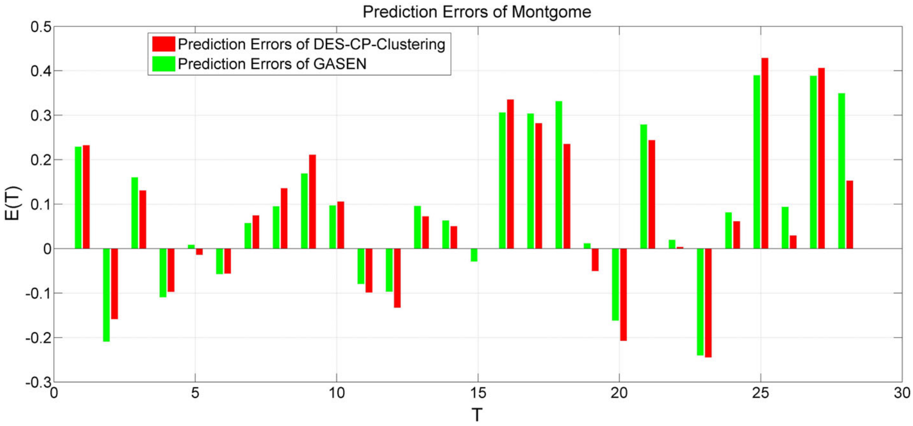


Fig. 20 Montgome (prediction errors)

obtained by DES-CP-Clustering and GASEN are displayed in Figs. 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 and 28.

From the above comparisons, it can be concluded that DES-CP-Clustering, one representative of the proposed DES algorithms, has better generalization performance than GASEN on the twelve benchmark time series prediction problems. In addition, a conclusion can be reached that the more training samples provided to the proposed models, the better performance can the models obtain. Therefore, in order to achieve better performance, adequate training samples are indispensable.

5 Conclusions and Future Works

Among the two types of selective ensemble paradigms, i.e., static and dynamic ensemble selection, the latter one has shown to be a very effective scheme for TSP problems. In this

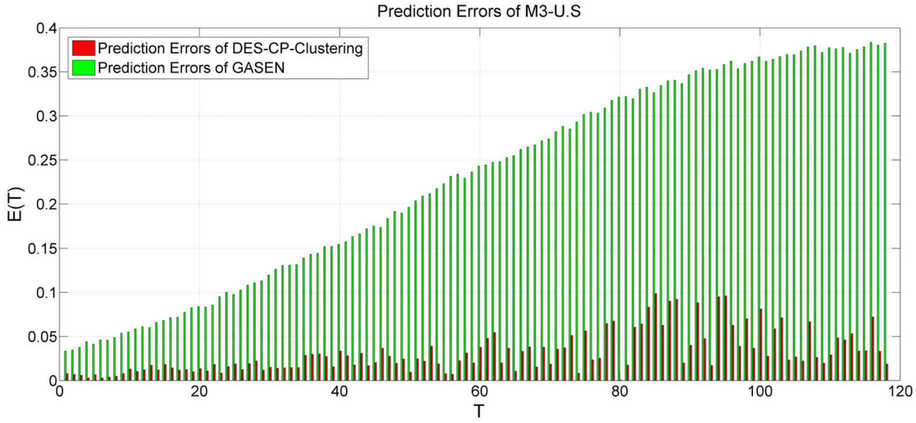


Fig. 21 M3-U.S (prediction errors)

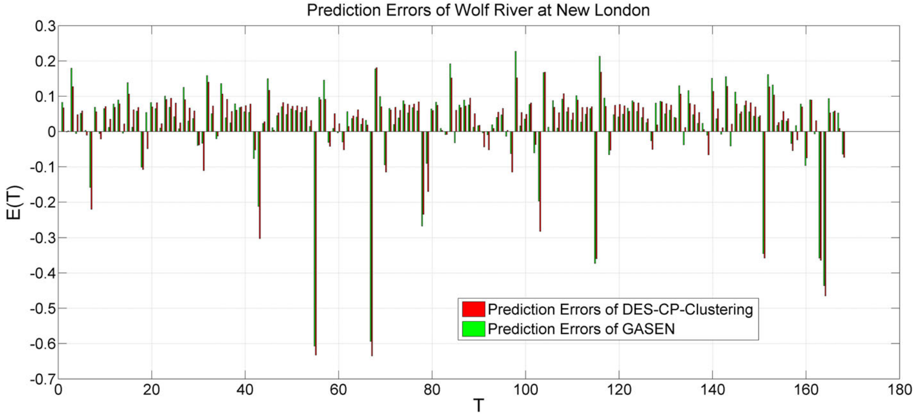


Fig. 22 Wolf River at New London (prediction errors)

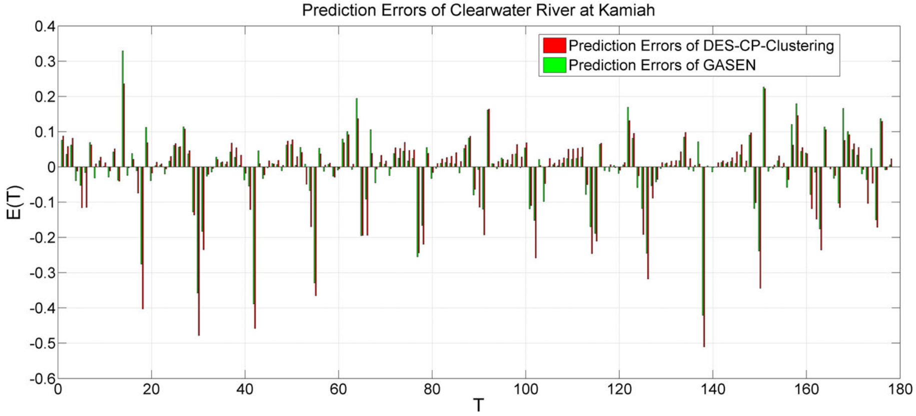


Fig. 23 Clearwater River at Kamiah (prediction errors)

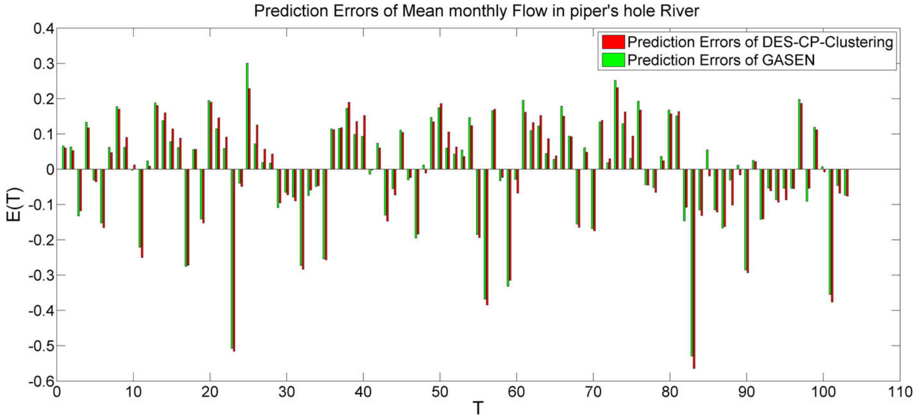


Fig. 24 Mean monthly Flow in piper's hole River (prediction errors)

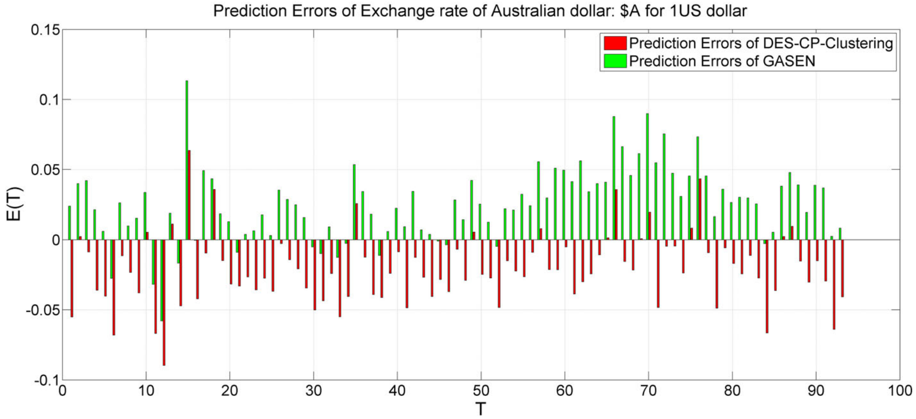


Fig. 25 Exchange rate of Australian dollar: $A for 1 US dollar (prediction errors)

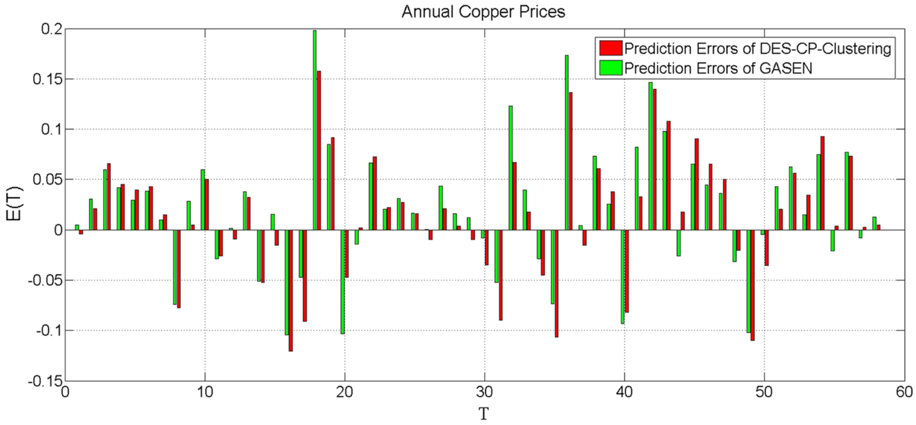


Fig. 26 Annual Copper Prices (prediction errors)

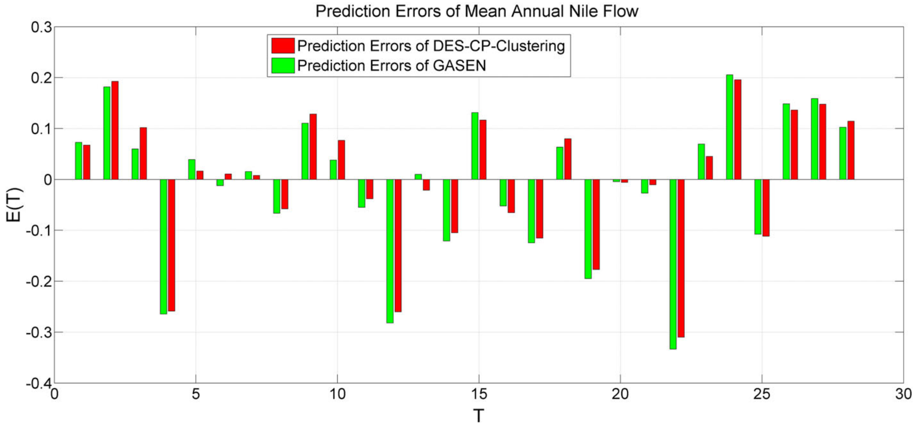


Fig. 27 Mean Annual Nile Flow (prediction errors)

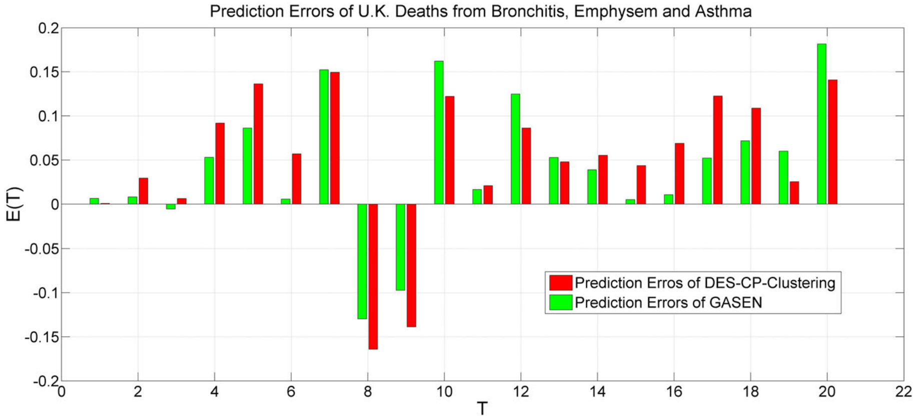


Fig. 28 U.K. Deaths from Bronchitis, Emphysem and Asthma (prediction errors)

paper, several new DES algorithms are designed for enhancing the ensemble generalization performance.

With the proposed DES-PALR algorithm, the predictors performing better on the local region could also perform better on the test sample, while different test sample locates different region of competence, thereby successfully realizing dynamic ensemble selection.

The strength of the proposed group of DVS-PvOv, DVS-NsAs and DVS-OpOp algorithms mainly lies in that, they are not limited by the quality of the local region of competence solely defined in the feature space, which greatly boost their predictive performances.

The major advantage of DES-CP is that, it does not need any information from the region of competence, and therefore, it is not restricted by the algorithms which define the region of competence.

The innovation of this work manifests in that, to our best knowledge, it is the first time that all these new DES algorithms are developed for the research of TSP.

Experimental results on twelve benchmark time series datasets verify that the proposed six DES algorithms achieve significantly higher predictive performance than the comparative state-of-the-art methods, including GASEN, ELM, H-ELM, BP and DrELM.

Future works on this topic will involve: (a) finding the ensemble of predictors that are complementary and diverse for solving TSP problems; (b) trying some different performance measurements so as to better assess prediction performance of models.

Acknowledgements This work is supported by the National Natural Science Foundation of China under the Grant No. 61473150.

References

1. Hamilton JD (1994) Time series analysis. Princeton University Press, Princeton
2. Brockwell PJ, Davis RA (2009) Introduction to time series and forecasting. Springer, Berlin
3. Pulido ME, Melin P (2012) Optimization of type-2 fuzzy integration in ensemble neural networks for predicting the Dow Jones time series. In: Fuzzy information processing society, pp 1–6
4. Palivonaite R, Ragulskis M (2016) Short-term time series algebraic forecasting with internal smoothing. Neurocomputing 171:854–865
5. Ma Z, Dai Q (2016) Selected an stacking ELMs for time series prediction. Neural Process Lett 44:1–26
6. Balkin SD, Ord JK (2000) Automatic neural network modeling for univariate time series. Int J Forecast 16:509–515
7. Giordano F, La Rocca M, Perna C (2007) Forecasting nonlinear time series with neural network sieve bootstrap. Comput Stat Data Anal 51:3871–3884
8. Jain A, Kumar AM (2007) Hybrid neural network models for hydrologic time series forecasting. Appl Soft Comput 7:585–592
9. Lapedes AS, Farber RF (1987) Nonlinear signal processing using neural networks: prediction and system modeling. In: 1. IEEE international conference on neural networks
10. Chakraborty K, Mehrotra K, Mohan CK, Ranka S (1992) Original contribution: forecasting the behavior of multivariate time series using neural networks. Neural Netw 5:961–970
11. Chatfield C, Weigend AS (1994) Time series prediction: forecasting the future and understanding the past: Neil A. Gershenfeld and Andreas S. Weigend, 1994, 'The future of time series', in: A.S. Weigend and N.A. Gershenfeld, eds., (Addison-Wesley, Reading, MA), 1–70. Int J Forecast 10:161–163
12. Adhikari R (2015) A neural network based linear ensemble framework for time series forecasting. Neurocomputing 157:231–242
13. Pelikan E, Groot CD, Wurtz D (1992) Power consumption in West-Bohemia: improved forecasts with decorrelating connectionist networks. Neural Netw World 2:701–712
14. Britto AS, Sabourin R, Oliveira LES (2014) Dynamic selection of classifiers—a comprehensive review. Pattern Recognit 47:3665–3680
15. Kittler J, Hatef M, Duin RPW, Matas J (1998) On combining classifiers. IEEE Trans Pattern Anal Mach Intell 20:226–239
16. Adhikari R, Verma G, Khandelwal I (2014) A model ranking based selective ensemble approach for time series forecasting. In: International conference on intelligent computing, communication and convergence, pp 14–21
17. Cruz RMO, Sabourin R, Cavalcanti GDC, Ren TI (2015) META-DES: a dynamic ensemble selection framework using meta-learning. Pattern Recognit 48:1925–1935
18. Gheyas IA, Smith LS (2011) A novel neural network ensemble architecture for time series forecasting. Neurocomputing 74:3855–3864
19. Kourentzes N, Barrow DK, Crone SF (2014) Neural network ensemble operators for time series forecasting. Expert Syst Appl Int J 41:4235–4244
20. Donate JP, Cortez P, Sánchez GG, Miguel ASD (2013) Time series forecasting using a weighted cross-validation evolutionary artificial neural network ensemble. Neurocomputing 109:27–32
21. Krikunov AV, Kovalchuk SV (2015) Dynamic selection of ensemble members in multi-model hydrometeorological ensemble forecasting. Procedia Comput Sci 66:220–227
22. Adhikari R, Verma G (2016) Time series forecasting through a dynamic weighted ensemble approach. Springer, New Delhi
23. Kolter JZ, Maloof MA (2007) Dynamic weighted majority: an ensemble method for drifting concepts. J Mach Learn Res 8:2755–2790
24. Woods K, Kegelmeyer WP, Bowyer K (1997) Combination of multiple classifiers using local accuracy estimates. IEEE Trans Pattern Anal Mach Intell 19:405–410

25. Smits PC (2002) Multiple classifier systems for supervised remote sensing image classification based on dynamic classifier selection. IEEE Trans Geosci Remote Sens 40(4):801–813
26. Kuncheva LI (2000) Clustering-and-selection model for classifier combination. In: International conference on knowledge-based intelligent engineering systems and allied technologies. Proceedings, vol 1, pp 185–188
27. Kuncheva LI, Bezdek JC, Duin RPW (2001) Decision templates for multiple classifier fusion: an experimental comparison. Pattern Recognit 34:299–314
28. Zhou ZH, Wu JX, Jiang Y, Chen SF (2001) Genetic algorithm based selective neural network ensemble. In: International joint conference on artificial intelligence, pp 797–802
29. Santos EMD, Sabourin R, Maupin P (2008) A dynamic overproduce-and-choose strategy for the selection of classifier ensembles. Pattern Recognit 41:2993–3009
30. Huang GB, Zhu QY, Siew CK (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
31. Zhao G, Shen Z, Miao C, Gay R (2008) Enhanced Extreme learning machine with stacked generalization. In: International joint conference on neural networks, pp 1191–1198
32. Huang GB (2003) Learning capability and storage capacity of two-hidden-layer feedforward networks. IEEE Trans Neural Netw 14:274–281
33. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: IEEE international joint conference on neural networks, vol 2, pp 985–990
34. Zhou Z-H, Wu J, Tang W (2002) Ensembling neural networks: many could be better than all. Artif Intell 137:239–263
35. Ko AHR, Sabourin R, Britto AS (2008) From dynamic classifier selection to dynamic ensemble selection. Pattern Recognit 41:1718–1731
36. Kuncheva LI (2002) Switching between selection and fusion in combining classifiers: an experiment. IEEE Trans Syst Man Cybern B Cybern 32:146–156
37. Hartigan JA, Wong MA (1979) Algorithm AS 136: a k-means clustering algorithm. Appl Stat 28:100–108
38. Wang S, Qi L, Yu P, Peng X (2011) CLS-SVM: a local modeling method for time series forecasting. Chin J Sci Instrum 32:1824–1829
39. Paterlini S, Minerva T (2003) Evolutionary approaches for cluster analysis. Soft Computing Applications. Physica-Verlag, HD
40. Bezdek JC, Pal NR (1998) Some new indexes of cluster validity. IEEE Trans Syst Man Cybern B Cybern 28:301–315
41. Dos Santos EM, Sabourin R, Maupin P (2006) Single and multi-objective genetic algorithms for the selection of ensemble of classifiers. In: International joint conference on neural networks, IJCNN, pp 3070–3077
42. Partalas I, Tsoumakas G, Vlahavas I (2008) Focused ensemble selection: a diversity-based method for greedy ensemble selection. In: European Conference on Artificial Intelligence (ECAI). IOS Press
43. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison-Wesley Pub. Co, Boston
44. Hyndman R (ed) Time Series Data Library. https://datamarket.com/data/list/
45. Root-mean-square deviation. https://en.wikipedia.org/wiki/Root-mean-square_deviation
46. Mean absolute error. https://en.wikipedia.org/wiki/Mean_absolute_error

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