

Research Article

Application of Discrete Wavelet Transform in Shapelet-Based Classification

Lijuan Yan ^{1,2}, Yanshen Liu,² and Yi Liu²

¹National Engineering Research Center for E-Learning, Central China Normal University, Wuhan 430079, China

²Hubei Research Center for Educational Informationization, Central China Normal University, Wuhan 430079, China

Correspondence should be addressed to Lijuan Yan; yanlijuan@mails.ccn.edu.cn

Received 24 April 2020; Revised 23 June 2020; Accepted 21 July 2020; Published 19 August 2020

Academic Editor: Paolo Crippa

Copyright © 2020 Lijuan Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recently, several shapelet-based methods have been proposed for time series classification, which are accomplished by identifying the most discriminating subsequence. However, for time series datasets in some application domains, pattern recognition on the original time series cannot always obtain ideal results. To address this issue, we propose an ensemble algorithm by combining time frequency analysis and shape similarity recognition of time series. Discrete wavelet transform is used to decompose the time series into different components, and the shapelet features are identified for each component. According to the different correlations between each component and the original time series, an ensemble classifier is built by weighted majority voting, and the Monte Carlo method is used to search for optimal weight vector. The comparative experiments and sensitivity analysis are conducted on 25 datasets from UCR Time Series Classification Archive, which is an important open dataset resource in time series mining. The results show the proposed method has a better performance in terms of accuracy and stability than the compared classifiers.

1. Introduction

A time series is a data sequence that represents recorded values of a phenomenon over time. Time series data constitutes a large portion of the data stored in real world databases [1]. Time series data have widely existed in many fields, such as commerce, agriculture, meteorology, bioscience, and ecology. Data such as meteorological data in weather forecast, floating currency exchange rate in foreign trade, radio wave, images captured by medical devices, and continuous signals in engineering applications can be regarded as time series [2]. Time series data are more complex to analyse than the cross-sectional data due to the way in which measurements change over time [3]. Time series classification (TSC) is one of the important tasks in time series data analysis. The TSC is applied to build a classification model based on labelled time series, and then the model is used to predict the label of unlabelled time series. Unlike traditional classification methods, the TSC requires not only numerical relationships between different attributes but also the order relationship between data.

In the past ten years, hundreds of methods have been proposed to solve the TSC problem. One of the traditional methods is the 1-nearest neighbor (1NN) classifier, which uses different distance functions. Faloutsos et al. [4] used Euclidean distance for time series matching. The Euclidean distance can only deal with time series of equal length, and it calculates time series point-to-point in the time axis but cannot match similar shapes if they are out of phase in the time axis. In order to solve these problems, Berndt et al. [5] applied dynamic time warping (DTW) technology in the speech recognition field to the pattern detection in time series. The DTW is a much more robust distance measure for time series. The DTW not only eliminates the “point-to-point” matching defect of Euclidean distance but also achieves “one-to-many” matching of time series data points through stretching or compressing the series. The traditional DTW assigns the same weight to each observation value and ignores the phase difference between the observation value and the test value. On this basis, Jeong et al. [6] proposed to use weighted DTW for time series classification. This kind of 1NN classification algorithm has high classification accuracy

and is easy to implement, but it consumes too long computing time and has poor interpretability. Many other researchers have concerned about the measurement of dissimilarity. Therefore, several dissimilarity metrics, such as normalized eigenvector correlation (NEC) [7], signal directional differences (SDDs) [8], and square eigenvector correlation (SEC) [9], are proposed recently, which measure the dissimilarity between the features extracted from the distinct path between specific features. These metrics have been verified to be effective in improving the accuracy of the feature matching technique.

Recently, many researchers have used shape similarity to solve TSC problems. The most popular method is shapelet-based classification. Shapelet is a time series subsequence which can be regarded as maximally representative of a class in some sense [10]. Classification algorithms based on shapelets were proposed at first time by Ye et al. [10, 11], and the algorithms used information gain to measure the split point of data and build decision tree by recursively searching the most discriminating shapelets. This strategy is to build a classifier at the same time as shapelets are discovered. In contrast, the other strategy is to map the time series to other spaces at first and then build a classifier. Lines et al. [12] proposed a time series classification method based on shapelet transformation (ST). This method creates new classification data before constructing the classifier, so that it keeps the explanatory power of shapelets and improves simultaneously the accuracy of classification.

Ensemble learning strategy has also been applied to time series classification, such as time series forest (TSF) proposed by Deng et al. [13], elastic ensemble (EE) method proposed by Lines et al. [14], Collection of Transformation Ensembles (COTEs) method proposed by Bagnall et al. [15], and the Hierarchical Vote Collective of Transformation-based Ensembles (HIVE-COTEs) method based on the COTE proposed by Lines et al. [16]. These methods combined multiple subclassifications, such as distance measure, shapelet identification, spectrum analysis, other time series feature representation, and transformation strategies. Compared to the method with a single classifier, the ensemble classification method has a higher accuracy, but a higher time complexity. In terms of classification accuracy, Bagnall et al. did a comparative experiment with the current popular time series classification algorithms [17, 18] and found the highest classification accuracy is in the order of HIVE-COTE, COTE, and ST. However, the ST is an important part of both COTE and COTE-HIVE algorithms. In other words, the ST is one of the effective methods to solve the time series classification.

Generally, new features extracted from time series may help to improve the performance of classification models. Techniques for feature extraction include singular value decomposition (SVD), discrete Fourier transform (DFT), discrete wavelet transform (DWT), and so on [19]. The DWT as formulated in the late 1980s has inspired extensive research into how to use this transform to study time series. The DWT is a powerful tool for a time-scale multiresolution representation on time series by using wavelets. In contrast to other techniques, the DWT is localized in time, and hence,

the wavelet variance can be readily adapted for exploring processes that are locally stationary with time varying [20] and for detecting inhomogeneities in time series [21]. Due to its ability to separate original time series into its decompositions, the DWT is a powerful tool to help researchers capture trends and patterns in data. At the same time, it is a data transformation technique that concurrently localizes both time and frequency information from the original data in its multiscale representation [22].

In this study, combining with the advantages of the DWT and shapelet approach, we propose a new ensemble method, which embeds the DWT into shapelet-discovery algorithm to get a transformed data and then implements an ensemble classifier to train and test the transformed data. By using the DWT, the original time series data are divided to one low-frequency information component and several high-frequency information components. Each decomposed information component is still in the time domain. The shapelet sets are then selected from each component, respectively. These shapelet sets reflect the corresponding classification characteristics and are used to convert the original time series into feature vector representations accordingly. These feature vectors contain more features of the original time series. Base classifier is trained with the transformed data. Finally, a weighted majority voting technique is used to integrate the prediction results of the base classifiers, and the Monte Carlo method is used to search for the local optimal weight vector. We make a comparative experiment with other popular time series classifiers and perform qualitative analysis in this study. The experiment is conducted on 25 datasets from UCR [23]. The results show the proposed method has a good performance in terms of accuracy and stability.

The paper is structured as follows: Section 2 provides related definitions on time series classification and shapelet; in Section 3, we propose a new method and describe the overall framework and the details of the method; in Section 4, we describe our experimental design and results and perform qualitative analysis for the proposed method; finally, we draw conclusions based on our analysis results in Section 5.

2. Related Definitions

Univariate time series dataset: a univariate time series is a sequence of data that are typically recorded in temporal order at fixed intervals. The number of real-valued data is the length of the time series.

A dataset $T = \{T_1, T_2, T_3, \dots, T_n\}$ has n time series. Each time series T_i has m real-valued ordered data $\langle t_{i,1}, t_{i,2}, t_{i,3}, \dots, t_{i,m} \rangle$ and a class label c_i and then $T_i = \{t_{i,1}, t_{i,2}, t_{i,3}, \dots, t_{i,m}, c_i\}$.

Sets of candidate shapelet: every subsequence of series in dataset T is defined as a candidate. So the set of candidate shapelets is the union of subsequences of each series in T . The subsequence of T_i is a contiguous sequence on T_i . The length of subsequence can be 1, 2, 3, \dots , m . A subsequence of T_i can be described as $S_{i,p,l} = \{t_{i,p}, t_{i,p+1}, t_{i,p+2}, \dots, t_{i,p+l-1}\}$, where p is the starting position and l is the length. So the set

of all subsequences of length l in the time series T_i is defined as $S_{T_{i,l}} = \{S_{i,p,l}, 1 \leq p \leq m - l + 1\}$.

Similarity measures: classification of time series depends on similarity measures between data. The common time series similarity measures include Euclidean distance, dynamic time warping, Fourier coefficients, and autoregressive model. In this study, Euclidean distance [24] is used to compare the similarity between two time series with the same length. For example, consider two m -length time series, S and R , and let Euclidean distance given by equation (1) be the utilized measure of similarity:

$$\text{dist}(S, R) = \sqrt{\sum_{i=1}^m (s_i - r_i)^2}. \quad (1)$$

Before calculating the distance, the z-normalization method is used to normalize each time series [25] according to equation (2). In equation (2), the \bar{X} and σ_x are mean and standard variance of m real-valued ordered reading data $\langle t_{i1}, t_{i2}, t_{i3}, \dots, t_{im} \rangle$ in each time series T_i , respectively:

$$x_{\text{norm}} = \frac{x - \bar{X}}{\sigma_x}. \quad (2)$$

The similarity between each candidate shapelet and each series is measured, and this sequence of distances with associated class membership is used to assess shapelet quality. The candidate shapelet is short, and the time series is relatively long. When calculating the distance between two time series with different lengths, the short series slides on the long series until getting the minimum distance between them. The distance between a time series T_i and a candidate shapelet S with length l is defined by equation (3). The distances between S and all subsequences of length l in T_i are calculated, and the minimum distance is taken as the distance between S and T_i :

$$\text{subdist}(S, T_i) = \min_{R \in S_{T_{i,l}}} (\text{dist}(S, R)). \quad (3)$$

Information gain and shapelet: in probability theory and information theory, the information gain (IG) is asymmetric to measure the difference between the two probability distributions. The IG is usually used to determine the quality of a shapelet [10, 11, 26]. After calculating all the distances between a candidate shapelet S and all time series in T , it will get a set D_S with n distance values. The D_S is sorted, and the IG at each possible split point sp is then assessed for S . Here, a valid split point is defined as the mean value between any two consecutive distances in D_S . For each possible split point sp , as shown in Figure 1, the IG is calculated by partitioning all elements of $D_S < sp$ into A_S , and all elements of $D_S > sp$ are grouped as B_S , respectively. The IG at sp is calculated according to the following equation:

$$\text{IG}(D_S, sp) = H(D_S) - \left(\frac{|A_S|}{|D_S|} H(A_S) + \frac{|B_S|}{|D_S|} H(B_S) \right), \quad (4)$$

where $|D_S|$ is the cardinality of the set D_S and $H(D_S)$ is the entropy of D_S . The $H(D_S)$ is defined as follows:

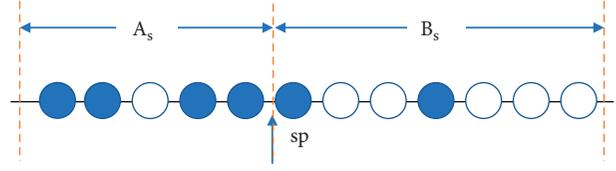


FIGURE 1: Split point.

$$H(D_S) = - \sum_{v \in V} p_v \log_2 p_v, \quad (5)$$

where V is the set of class label and p_v is the probability of each label.

The IG of shapelet S , IG_S , is calculated as

$$\text{IG}_S = \max_{sp \in D_S} \text{IG}(D_S, sp). \quad (6)$$

In general, shapelets are extracted with maximum information gain by comparing all the candidate shapelets.

3. The Proposed Method

3.1. Method Structure. The proposed method in this study consists of three major parts: decomposition, feature extraction, and classification. The whole process of the proposed method is outlined in Figure 2.

The three major parts of the proposed method are described briefly as listed below:

- (1) Using decimated DWT, time series is decomposed into different components in form of one approximation component (A_R) and several detail components (D_1, D_2, \dots, D_R).
- (2) Shapelet transform is used on each component to extract shapelets and transform the data to a set of new feature vector.
- (3) The transformed data are fed into a base classifier (L) to predict class label. Based on the predictive result of the base classifier, a weighted majority voting is implemented to build an ensemble classifier according to the correlation between components and original data. The weights are optimized by the Monte Carlo method, and then, the final classification result can be obtained.

3.2. Discrete Wavelet Transform. The DWT is a technique of a mathematical origin and is very appropriate for a time-scale multiresolution analysis on time series [22]. The DWT provides an effective way to isolate nonstationary signals into signals at various scales. This kind of signal processing is called signal decompositions. Various aspects of nonstationary signals such as trends, discontinuities, and repeated patterns are clearly revealed in the signal decompositions. Some time series data have multiscale signal components that are more meaningful in parts than in sum, such as audio signals and patients' ECG heart rates. For those reasons, the DWT is a suitable technique to combine with classification approaches in order to categorize an unknown signal into a

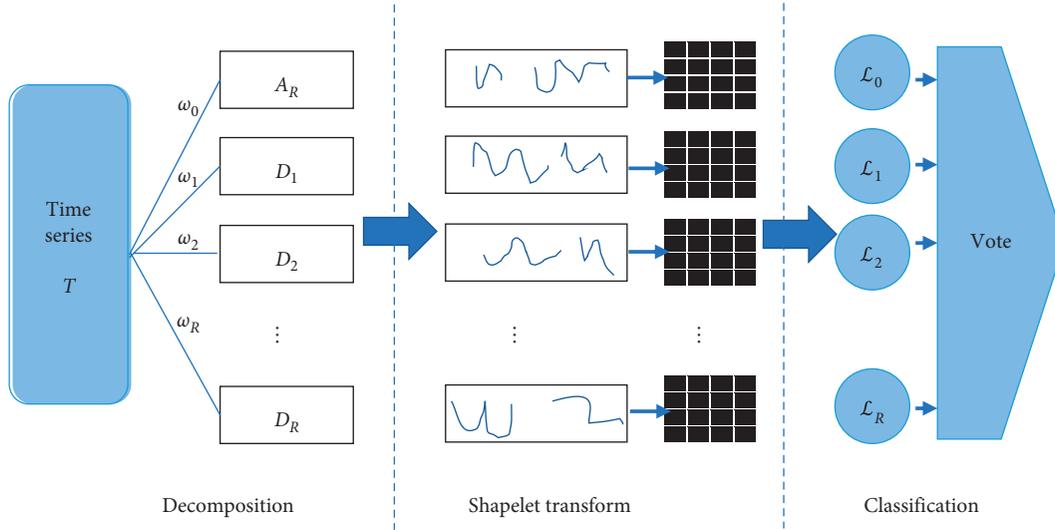


FIGURE 2: The overall framework of the proposed method.

predefined type of signals [22]. This section explains how the DWT assists in the classification process.

The effective way to implement DWT is to use a filter, which was proposed by Mallat in 1988 and is well-known as Mallat algorithm. This algorithm uses filter banks to implement the DWT which can decompose the signal into several different frequency components, and Figure 3 illustrates an example of a two-level wavelet decomposition and reconstruction processes of the decimated DWT.

Generally, a filter bank approach is adopted because of its efficiency. As shown in Figure 3, the $S(n)$ is a real signal, $h(n)$ is the high-pass filters which filter out the low-frequency part of the signal, and $g(n)$ is the low-pass filters which can filter out the high-frequency part. The half-band filters downsample the signal by a factor of 2 at each level of decomposition. At the first level decomposition, the input signal is firstly passed through the wavelet filters and followed by a decimation factor of two. Then, the output of the low-pass filter is used as the new input signal, and the same filtering and decimation process will be reiterated. This is carried out until the desired level of wavelet decomposition is reached, or the allowed maximum level is reached. The combination of the filtering and the decimation processes enables the same filters to be used throughout the entire wavelet decomposition procedure [27]. The outputs of the decomposition process are the approximation coefficients (cA_i) and detail coefficient (cD_i), where i denotes the level of filter. In practical application, the appropriate decomposition level is generally selected according to the characteristics of the signal or the appropriate standard.

For the reconstruction process, the original signal can be reconstructed from the approximate and detail coefficients at every level by upsampling by two, passing through high- and low-pass synthesis filters, and adding them. The original signal can be reconstructed from the approximation coefficients of the last level and detail coefficients of each level.

Similarly, the approximate component (A) and the detail component (D) of the signal can be reconstructed from the

approximate coefficient and the detail coefficient by omitting the other sets of coefficients, separately. This can be done best by setting the corresponding coefficients to zero of matching the same shape. In this way, the reconstructed component is the same length as the original signal. Approximation component can capture rough features that can be used to estimate the original data, while detail components can capture detail features that can be used to describe frequent movements of the data. The amount of information contained in different frequency levels is different. With the deepening of the decomposition level, the curve information carried by components is gradually reduced.

The DWT has been used to break down an original time series. An original time series data can be decomposed into two types of component: approximation component and detail component via the above method. Each component may carry meaningful signals of the original time series. For example, if the selected level of decomposition is 4, the original time series data are decomposed into one approximation component and 4-dimensional detail components, as shown in Figure 4. These reconstructed components are still in the time domain. Consequently, the DWT is considered as a time-scale transformation [28]. The approximate component A reflects the overall trend of time series, while the detail component D reflects the characteristics of time series under the interference of different factors.

For example, considering a dataset containing n time series and class labels, each time series has m data points. After choosing the mother wavelet, if the maximum level allowed is R , we can get approximation component matrix $A_{n,m+1}$ and R detail component matrixes $D_{n,m+1}$.

The DWT decomposes a single signal into multiscale signals using wavelet functions. The filter coefficients are determined by the mother wavelet. The characteristics of the transformation are also impacted by the choice of the mother wavelet. The commonly used mother wavelets

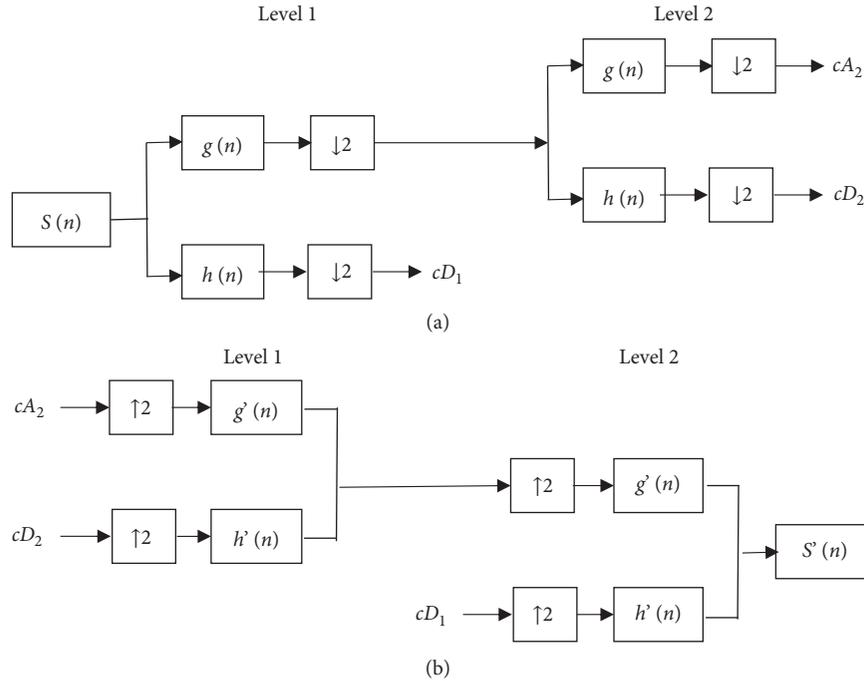


FIGURE 3: A two-level wavelet decomposition (a) and reconstruction (b) processes for the decimated DWT.

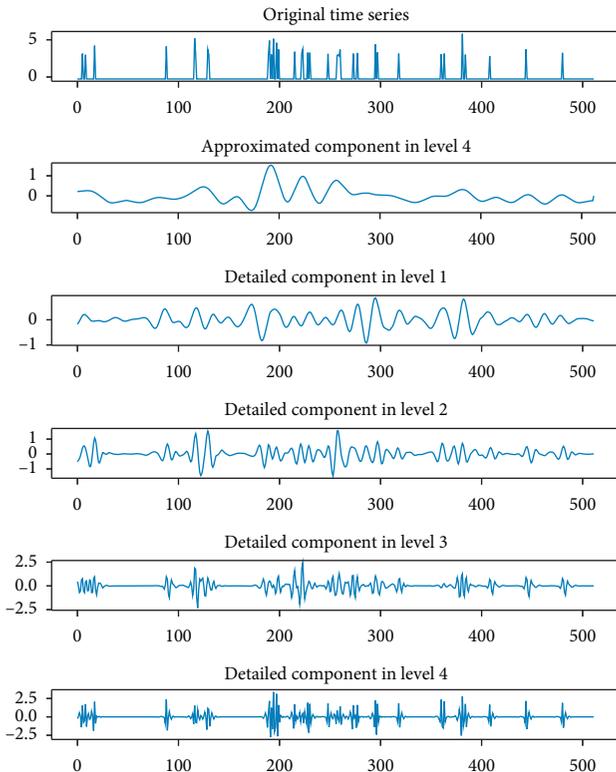


FIGURE 4: A time series in earthquakes dataset decomposed with level 4 and sym8 wavelet is used.

include Haar, Daubechies, biorthogonal, Coiflets, and symlets. The influence of different mother wavelets on classification performance will be tested in the following experiments.

3.3. *Feature Extraction.* We extract features of on each component through the shapelet transformation, which has been proposed by Lines et al. [12]. The main contribution of shapelet transformation is to separate shapelets discovery and classifier construction. The transformed data can be used in different classifiers. The corresponding algorithm includes two major steps:

- Step 1: the algorithm performs a single scan of the data to extract the best k shapelets.
- Step 2: by calculating the distance between k shapelets and every time series, an instance with k attributes is obtained; then, a new transformed dataset is created.

Algorithm 1 describes the process of extracting k best shapelets from the dataset. The min and max parameters limit the length of the candidate shapelets. Each time a candidate shapelet is obtained, and the distance between the candidate shapelet and every time series is calculated. The results are sorted to calculate the split point that can be used to get the maximum information gain. After all the candidate shapelets are accessed, they are sorted according to the information gain and self-similar shapelets are removed. Finally, the top k shapelets are retained in the set of nonself-similar shapelets.

Once the best k shapelets have been found, the transform is performed with Algorithm 2. For each instance of data T_i , the subsequence distance is computed between T_i and SK_j , where $j = 1, 2, \dots, k$. The calculated k distances are used to form a new instance of transformed data, where each attribute corresponds to the distance between a shapelet and the original time series. The subsequence distance calculation has been described in equation (3).

```

Input: a list of time series  $T$ , min, and max length shapelet to search for and  $k$  the maximum number of shapelets to find
Output: the best  $k$  shapelets
1:  $k$  shapelets  $\leftarrow \Phi$ 
2: for all  $T_i$  in  $T$  do
3: shapelets  $\leftarrow \Phi$ 
4: for  $l \leftarrow \min$  to  $\max$  do
5: for  $p \leftarrow 1$  to  $m-l+1$  do
6:  $S_{T_{i,l}} \leftarrow \text{generateCandidate}(T_i, l)$ 
7: for all candidate shapelet  $S$  in  $S_{T_{i,l}}$  do
8:  $D_S \leftarrow \text{subdist}(S, T)$ 
9: quality  $\leftarrow \text{assessCandidate}(S, D_S)$ 
10: Shapelets.add( $S$ , quality)
11: removeSelfSimilar(shapelets)
12: sortByQuality(nonselt – similar shapelets)
13: return kShapelets

```

ALGORITHM 1: ShapeletSelection(T , min, max, k).

```

Input: SK, a set of the best  $k$  shapelets which is generated from the training data and  $T$ , dataset containing time series and class labels
Output: a new transformed dataset
1:  $T' \leftarrow \Phi$ 
2: for all  $T_i$  in  $T$  do
3:  $T'_i \leftarrow \langle \rangle$ 
4: for all shapelets  $S$  in SK do
5: dist  $\leftarrow \text{subdist}(S, T_i)$ 
6:  $T'_i \leftarrow \text{append}(T'_i, \text{dist})$ 
7:  $T'_i \leftarrow \text{append}(T'_i, T.c)$ 
8:  $T' \leftarrow T' \cup T'_i$ 
9: return  $T'$ 

```

ALGORITHM 2: ShapeletTransform(SK, T).

With shapelet transformation technology, the selection process of shapelets is optimized, and different classification strategies can be flexibly applied. On this basis, several other shapelet approaches have been proposed, such as logical shapelets [26], fast shapelets [29], binary shapelets [30], and learnt shapelets [31].

The extracted low-frequency and high-frequency information components in the time domain are used as separate new time series to generate candidate matrix. Then, the corresponding shapelets are extracted from the candidate matrix. The distance between the shapelets set extracted from each component will be calculated to form a set of new feature vector. In this step, we can get $R+1$ transformed matrix $T'_{k,m+1}$.

3.4. Ensemble Classification. In this paper, we build a combined classifier finally. We train the base classifier on the $R+1$ transformation matrix and use weighted majority voting to integrate the prediction results of the base classifiers, and then use the Monte Carlo method to optimize the weight vector. The above process is described by Algorithm 3.

In order to evaluate the strength and direction of relationship between each component and original time series,

Pearson correlation coefficient is calculated. The obtained correlation coefficient matrix is normalized to meet the equation 7. The mean value of each type of component is taken as the initial value of weight ω_j , where j can be 0, 1, 2, 3, ..., R . The weights meet the condition shown as follows:

$$\sum_{j=0}^R \omega_j = 1. \quad (7)$$

For the component with high correlation with the original data, its classifier is assigned a larger weight, so as to improve the performance of the ensemble classifier.

We discuss a multiple classification task with class labels $i \in \{1, 2, \dots, c\}$ and predict the class label \hat{y} based on the predicted probabilities p for each base classifier L_j , where j can be 0, 2, 3, ..., R . The label \hat{y} is calculated as follows:

$$\hat{y} = \arg \max_i \sum_{j=0}^R \omega_j p_{ij}, \quad (8)$$

where ω_j is the weight of the j th base classifier L_j and p_{ij} is the class probability for j th classifier L .

The key part to build the ensemble classifier is the selection of weights. In the proposed method, the Monte Carlo

```

Input:  $R+1$  transformed matrix  $T'$ 
the original time series dataset  $T$ 
base classifier  $L$ 
simulation times  $N$ 
Output: the optimal weights and the maximum accuracy
1: get the initial weight  $\omega < \omega_1, \omega_2, \dots, \omega_{R+1} >$  and step length  $d_j$ 
2: for all  $T'_i$  in  $T'$  do
3:  $L^*(T'_i, \omega) = \text{voting}(L_1(T'_i), L_2(T'_i), \dots, L_{R+1}(T'_i))$ 
4:  $\text{acc} \leftarrow \text{computeAccuracy}(L^*(T'_i, \omega))$ 
5:  $\text{accList} \leftarrow < >$ 
6:  $\text{accList} \leftarrow \text{append}(\text{acc})$ 
7: while True
8:  $\text{acc}' \leftarrow \text{MonteCarlo}(N d_j)$ 
9:  $\text{accList} \leftarrow \text{append}(\text{acc}')$ 
10: update simulation repeat time from  $N$  to  $2N$ 
11: update step length from  $d_j$  to  $2d_j$ 
12: if  $\Delta \text{accList} < 0$ 
13: break
14: return  $\omega'$  and  $\max(\text{accList})$ 

```

ALGORITHM 3: EnsembleClassifier(T', T, L, N).

```

Input: simulation times  $N$  step length  $d_j$ 
Output: the maximum accuracy of  $N$  simulations
1: List  $\leftarrow < >$ 
2: for  $i \leftarrow 1$  to  $N$  do
3: for  $j \leftarrow 1$  to  $R+1$  do
4: generate random weight  $\omega'_j \in [\omega_j - d_j, \omega_j + d_j]$ 
5: List  $\leftarrow \text{append}(\text{computeAccuracy}(L^*(T'_i, \omega')))$ 
6:  $\text{acc}' \leftarrow \max(\text{List})$ 
7: return  $\omega'$  and  $\text{acc}'$ 

```

ALGORITHM 4: Monte Carlo ($N d_j$).

method is used to find the optimal weight parameters, as described in Algorithm 4. It includes following major steps:

Step 1: Pearson correlation coefficient of each component and the original time series is calculated and normalized. The mean value of each type of component is taken as the initial value of the weight ω_j .

Step 2: the initial weight ω_j is multiplied by the predicted class probability of the base classifier corresponding to each component, and the maximum probability is taken to determine the final class and to obtain the accuracy of the ensemble classifier.

Step 3: the extreme value of each component's Pearson correlation coefficient can be calculated in Step 1, and it is recorded as d_j , where j can be 0, 2, 3, ..., R . The new weight combination is generated by the Monte Carlo method. In each Monte Carlo event, we generate $R+1$ uniformly distributed random number in range of $[\omega_j - d_j, \omega_j + d_j]$. After N simulations, N groups of weight combination will be produced.

Step 4: the N groups of weight combination will be substituted into Step 2 to calculate the accuracy,

respectively. The maximum accuracy is the result of this step.

Each iteration contains N times Monte Carlo simulation. If the accuracy does not improve compared to the accuracy in last iteration, we will update d_j to $2d_j$ to broaden the domain of generated random numbers and increase the Monte Carlo statistics from N to $2N$.

Monte Carlo simulation is a computerized mathematical technique to generate random sample data based on given distribution for numerical experiments. We use Monte Carlo to generate a large set of random weight vector, and the range of weight is constrained by d_j so that the prediction result of components with strong correlation will be given a higher weight. Different weight vectors are calculated with the above method to get different accuracies, and the optimal weight vector and accuracy are obtained after several Monte Carlo iterations.

In Figure 5, the blue dot line indicates the termination position of the iterations. The condition of iteration termination is that the accuracy obtained is no longer increasing. Obviously, this method cannot obtain the global optimum, but the weight obtained is closest to the initial

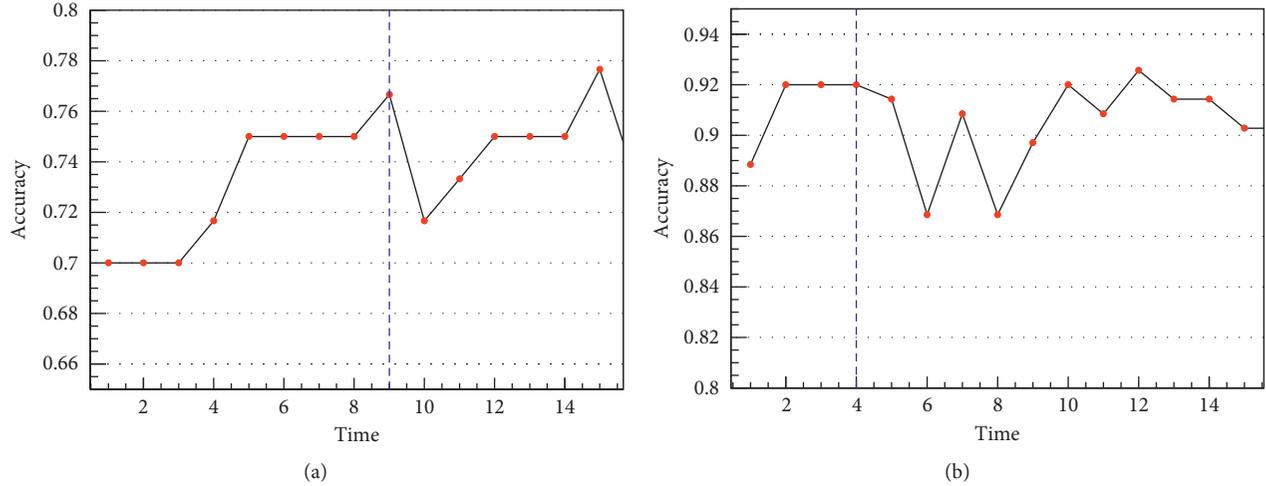


FIGURE 5: The learning curves of datasets the Car (a) and the Fish (b).

weight. It is in line with our assumption that the more relevant components play a more important role in classification.

4. Experiment

4.1. Experimental Settings

4.1.1. Experimental Dataset. In this paper, we use 25 datasets from UCR repository [23]. These have been commonly adopted by TSC researchers. The basic information of the datasets is shown in Table 1.

The classification labels of multiclassification datasets are represented by Arabic numerals. For example, for 4 classification datasets, the classification labels are 1, 2, 3, and 4, respectively. As shown in Table 1, the types of datasets used are diverse and come from three fields, including sensor data, image contour information, human ECG, and action data. The length is also different, the shortest is 24, and the longest is 512. Therefore, the performance of the algorithm can be comprehensively tested. In order to facilitate the performance comparison, the default training set and test set partition are adopted in this paper, k value is set to $m/2$, min value is selected to 3, the max value is m , and m is the length of time series. The initial value of N is 1000 in our experiments.

4.1.2. Experiment Design. Our first objective is to choose a base classifier which has best performance on transformed data. For this purpose, we test the performance of five traditional classifiers on the transformed data constructed by the ST method. These classifiers are Naïve Bayes [32], C4.5 decision tree [33], support vector machines [34] with polykernels, random forest [35, 36] (with 100 trees), and Bayesian networks [37]. These algorithms are commonly used in machine learning.

The characteristics of the transformation are impacted by the choice of the mother wavelet and the number of detail

levels, and thus, the mother wavelet type and the number of detail levels should be taken into consideration in the experiment. We try different mother wavelets and number of levels to test the influence of these two parameters on the results.

Finally, we implement a comparative experiment to compare the performance between our method (DSE) and other six time series classifiers, including 1-nearest neighbor classifiers using Euclidean distance (1NN-ED) based on raw data, 1-nearest neighbor classifiers using dynamic time warping (1NN-DTW) based on raw data, 1-nearest neighbor classifiers using dynamic time warping with window size set through cross validation (1NN-DTWCV) based on raw data, a random forest classifier based on raw data binary shapelet transform (BinaryST) [30], time series forest (TSF) [13], and elastic ensemble (EE) [14].

4.1.3. Evaluating Indicator. To the classification problem, classification accuracy is the most important criterion to evaluate algorithm performance. In addition to accuracy, Friedman test and Nemenyi test are widely used in machine learning to evaluate the performance of algorithms over multiple datasets. After getting the accuracy of the K algorithms on the N dataset, Friedman test ranks algorithms for each dataset separately. The algorithm with the highest classification accuracy is marked as 1, and the second-highest label is marked as 2, and so forth. The algorithms with the same accuracy value will be marked as average ranks between them. In this way, we can get a rank matrix of $N \times K$. r_{ij} is the rank mark of the i^{th} dataset on the j^{th} algorithm, and the average ranges R_j are calculated as follows:

$$R_j = \frac{1}{N} \sum_{i=1}^N r_{ij}, \quad (9)$$

Under the null hypothesis, all algorithms are equivalent, so their R_j should be equal. The Friedman statistics is defined by

TABLE 1: Datasets used in the experiments.

#	Dataset	Type	Train	Test	Length	Class
1	ECG200	ECG	100	100	96	2
2	ECGFiveDays	ECG	100	861	136	2
3	TwoLeadECG	ECG	100	1139	82	2
4	ECG5000	ECG	100	4500	140	5
5	BeetleFly	Image	20	20	512	2
6	BirdChicken	Image	20	20	512	2
7	DistalPhalanxOutlineCorrect	Image	20	276	80	2
8	Herring	Image	20	64	512	2
9	MiddlePhalanxOutlineCorrect	Image	20	291	80	2
10	PhalangesOutlinesCorrect	Image	276	858	80	2
11	ProximalPhalanxOutlineCorrect	Image	64	291	80	2
12	Yoga	Image	291	3000	426	2
13	FaceFour	Image	858	88	350	4
14	Fish	Image	291	175	463	7
15	ArrowHead	Image	3000	175	251	3
16	Earthquakes	Sensor	139	139	512	2
17	FordA	Sensor	139	1320	500	2
18	FordB	Sensor	139	810	500	2
19	ItalyPowerDemand	Sensor	139	1029	24	2
20	MoteStrain	Sensor	139	1252	84	2
21	SonyAIBORobotSurface1	Sensor	139	601	70	2
22	DodgerLoopGame	Sensor	139	138	288	2
23	DodgerLoopWeekend	Sensor	139	138	288	2
24	Plane	Sensor	139	105	144	7
25	Car	Sensor	139	60	577	4

$$\chi_F^2 = \frac{12N}{K(K+1)} \left[\sum_{j=1}^K R_j^2 - \frac{K(K+1)^2}{4} \right], \quad (10)$$

which is according to χ_F^2 with $K - 1$ degrees of freedom.

The research of Demiša et al. [38] shows that Friedman's statistics are too conservative and proposed a better statistical formula as follows:

$$F_F = \frac{(N-1)\chi_F^2}{N(K-1) - \chi_F^2}, \quad (11)$$

which is according to the F-distribution with $K - 1$ and $(K - 1)(N - 1)$ degrees of freedom.

If the null hypothesis is rejected, indicating significant differences between these algorithms, the difference between the algorithms can be tested by the Nemenyi test to compare all the algorithms to each other. At a significance level of α , the critical difference (CD) value is defined by the following equation:

$$CD = q_\alpha \sqrt{\frac{K(K+1)}{6N}}. \quad (12)$$

All algorithms were divided into different groups by the CD value so that there was no significant difference in the performance of the algorithms in the group. In this way, performance differences between different algorithms can be represented by the critical difference diagram.

4.2. Experiment Results. The experimental platform used in this paper is Python 3.7, hardware configuration: Pentium Dual Core CPU (2.5 GHz), 8G memory.

4.2.1. Base Classifier Selection. Table 2 lists the accuracy results from five classifiers on the transformed data. Random forest has a good performance, with an average rank of 2.2200 and the best performance in 13 out of 25 problems. The results show that random forest provides a reliable predictive performance on different datasets.

Random forest [35] refers to an ensemble learning method of training, classifying, and predicting sample data by using multiple decision trees whose outputs are aggregated by majority voting. To classify a new instance, each decision tree provides a classification for input data; random forest collects the classifications and chooses the most voted prediction as the result. The input of each tree is sampled data from the original dataset. In addition, a subset of features is randomly selected from the optional features to grow the tree at each node. Each tree is grown without pruning. Essentially, random forest enables many weak or weakly correlated classifiers to form a strong classifier [36]. It does not need to assume data distribution; it can handle thousands of input variables without variable deletion. It is relatively fast, simple, robust to outliers and noise, and easily parallelized; avoids overfitting; and performs well in many classification problems.

In the following experiments, we chose random forest as the base classifier.

4.2.2. Sensitivity of Parameters. Performing the DWT on the original time series instance can get the reconstruction wavelet coefficient. Before employing DWT, two parameters need to be specified, and they are mother wavelet type and the number of detail levels required. In this paper, we

TABLE 2: Testing accuracy and ranks of base classifiers.

#	Naïve Bayes	C4.5	SVM	Random forest	Bayesian networks
1	0.8100	0.7900	0.8700	0.8800	0.8000
2	0.9895	0.9768	0.9849	0.9978	0.9977
3	0.9719	0.8788	0.9991	0.9745	0.9930
4	0.8836	0.8862	0.8733	0.8933	0.8942
5	0.7500	0.6500	0.9000	0.8500	0.9000
6	0.7000	0.7000	0.8000	0.7500	0.8000
7	0.6594	0.6884	0.5870	0.6051	0.6884
8	0.5000	0.6250	0.5938	0.6563	0.6250
9	0.5704	0.6598	0.5704	0.6598	0.6564
10	0.6492	0.6678	0.6480	0.6853	0.6538
11	0.6357	0.7801	0.8007	0.8797	0.6667
12	0.6493	0.7647	0.7050	0.8383	0.6753
13	0.9886	0.7273	1.0000	1.0000	1.0000
14	0.9200	0.8171	0.9771	0.9314	0.9486
15	0.5543	0.6114	0.7257	0.7200	0.6286
16	0.9065	0.9712	0.8777	1.0000	0.8129
17	0.6977	0.7038	0.7098	0.6864	0.7060
18	0.6247	0.6370	0.6407	0.6074	0.6370
19	0.9407	0.8552	0.9446	0.9417	0.9193
20	0.9273	0.8043	0.9081	0.9401	0.9401
21	0.9584	0.9750	0.9384	0.9401	0.9401
22	0.7067	0.5512	0.6929	0.6772	0.5512
23	0.9683	0.9603	0.9841	0.9841	0.9683
24	1.0000	0.9048	0.9048	1.0000	1.0000
25	0.6500	0.6833	0.7833	0.7833	0.7000
Average rank	3.7000	3.6200	2.7200	2.2200	2.7400
Win	2	3	11	13	7

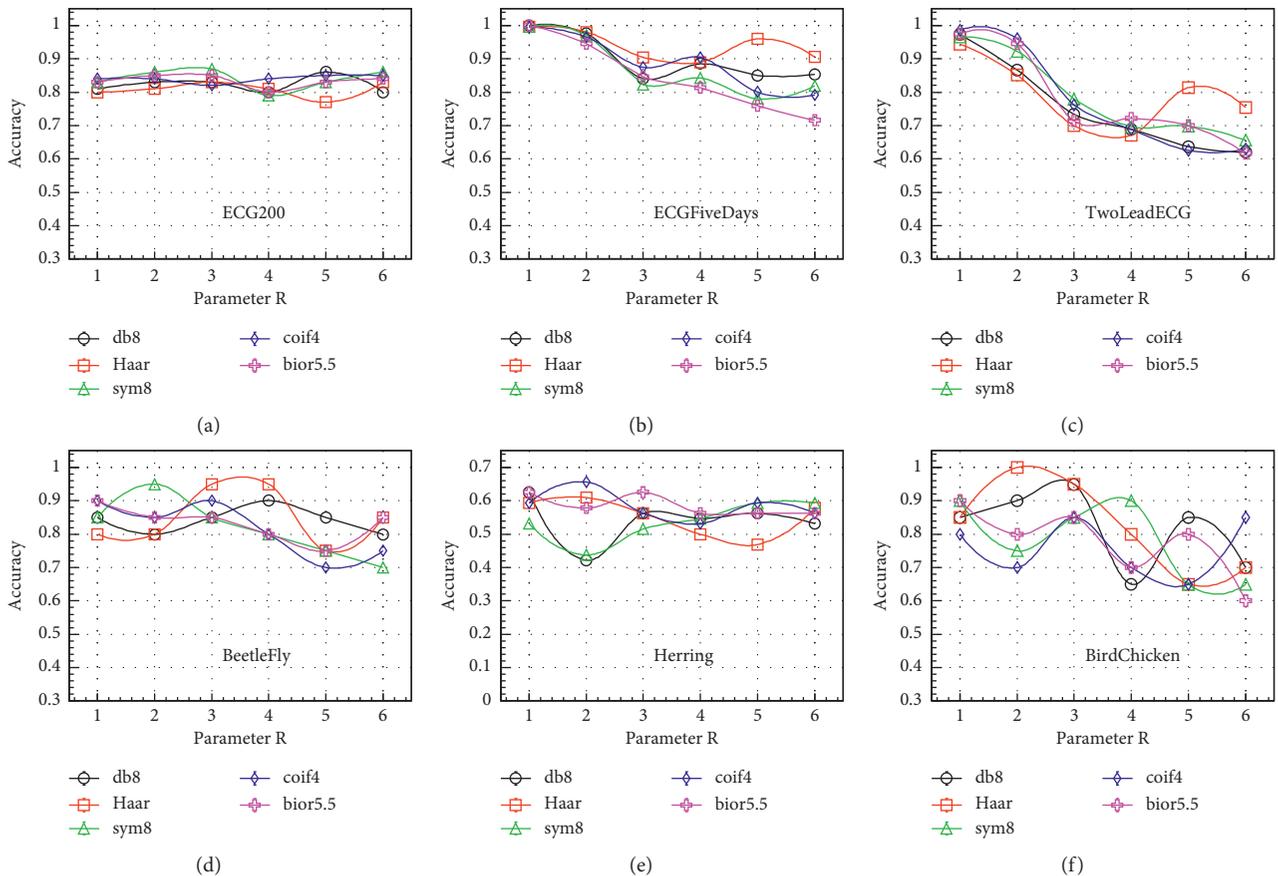


FIGURE 6: Continued.

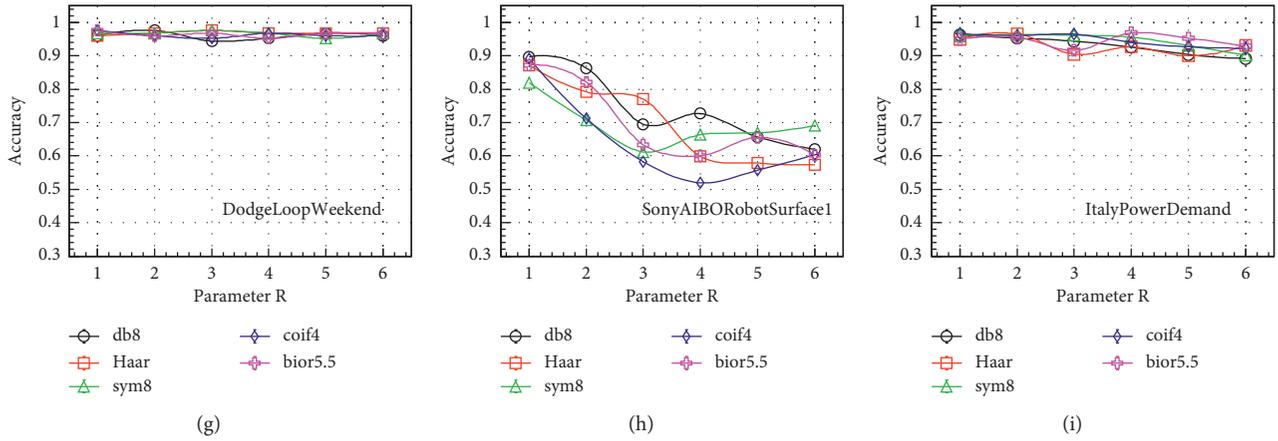


FIGURE 6: Sensitivity of mother wavelet and level.

TABLE 3: Testing accuracy and ranks of different classifiers on the datasets.

#	INN-ED	INN-DTW	INN-DTWCV	BinaryST	TSF	EE	DSE
1	0.8800 (2)	0.7700 (7)	0.8800 (2)	0.8300 (5)	0.8200 (6)	0.8800 (2)	0.8700 (4)
2	0.7967 (5.5)	0.7677 (7)	0.7967 (5.5)	1.0000 (1.5)	0.9872 (3)	0.8409 (4)	1.0000 (1.5)
3	0.7471 (7)	0.9043 (4)	0.8683 (5)	0.9894 (1)	0.8306 (6)	0.9333 (3)	0.9860 (2)
4	0.9249 (4)	0.9244 (5)	0.9251 (3)	0.8438 (7)	0.9013 (6)	0.9300 (1)	0.9282 (2)
5	0.7500 (3)	0.7000 (5.5)	0.7000 (5.5)	0.7500 (3)	0.7500 (3)	0.6000 (7)	0.9500 (1)
6	0.5500 (7)	0.7500 (4.5)	0.7000 (6)	0.9500 (2)	0.9000 (3)	0.7500 (4.5)	1.0000 (1)
7	0.7174 (5.5)	0.7174 (5.5)	0.7246 (4)	0.7899(1)	0.7138 (7)	0.7681 (2)	0.7304 (3)
8	0.5156 (6)	0.5312 (4.5)	0.5312 (4.5)	0.4219 (7)	0.5625 (3)	0.7031 (1)	0.6562 (2)
9	0.7663 (2.5)	0.6976 (7)	0.7663 (2.5)	0.7251 (6)	0.7354 (4.5)	0.7904 (1)	0.7354 (4.5)
10	0.7611 (2.5)	0.7284 (5)	0.7611 (2.5)	0.6911 (6)	0.7319 (4)	0.7832 (1)	0.6131 (7)
11	0.8076 (4)	0.7835 (6)	0.7904 (5)	0.8832 (1)	0.7423 (7)	0.8282 (3)	0.8660 (2)
12	0.8303 (4)	0.8363 (3)	0.8440 (2)	0.7573 (6)	0.7463 (7)	0.8793 (1)	0.8207 (5)
13	0.7841 (6)	0.8295 (5)	0.8864 (4)	0.9886 (2)	0.7614 (7)	0.9091 (3)	1.0000 (1)
14	0.7829 (6)	0.8229 (5)	0.8457 (4)	0.8571 (3)	0.5429 (7)	0.9657 (1)	0.9200 (2)
15	0.8000 (3.5)	0.7029 (6)	0.8000 (3.5)	0.7485 (5)	0.4800 (7)	0.8400 (1)	0.8057 (2)
16	0.7122 (6.5)	0.7194 (5)	0.7266 (3)	0.7122 (6.5)	1.0000 (1.5)	0.7194 (4)	1.0000 (1.5)
17	0.6652 (4)	0.5545 (7)	0.6909 (3)	0.6621 (5)	0.6106 (6)	0.8182 (1)	0.7288 (2)
18	0.6062 (5)	0.6198 (3)	0.6074 (4)	0.4963 (7)	0.5556 (6)	0.7346 (1)	0.6988 (2)
19	0.9553 (4.5)	0.9504 (6)	0.9553 (4.5)	0.9495 (7)	0.9602 (3)	0.9611 (2)	0.9689 (1)
20	0.8786 (4)	0.8347 (7)	0.8658 (5)	0.9173 (2)	0.8411 (6)	0.8858 (3)	0.9449 (1)
21	0.6955 (6.5)	0.7255 (4)	0.6955 (6.5)	0.9168 (1)	0.7654 (3)	0.7072 (5)	0.8968 (2)
22	0.8841 (3)	0.8768 (4)	0.9275 (1)	0.7953 (7)	0.8031 (6)	0.8898 (2)	0.8346 (5)
23	0.9855 (1)	0.9493 (7)	0.9783 (4)	0.9606 (6)	0.9841 (2.5)	0.9841 (2.5)	0.9762 (5)
24	0.9619 (5.5)	1.0000 (2.5)	1.0000 (2.5)	0.9619 (5.5)	0.9429 (7)	1.0000 (2.5)	1.0000 (2.5)
25	0.7333 (5.5)	0.7333 (5.5)	0.7667 (3)	0.7833 (2)	0.5333 (7)	0.8333 (1)	0.7500 (4)
Average rank	4.5600	5.2400	3.8200	4.2200	5.1400	2.3800	2.6400
Win	2	1	3	4	1	12	8

Note. The results highlighted in bold denote that the method gets the highest accuracy for this dataset.

measured the impact of mother wavelet and level of decomposition. We tested five different mother wavelets (db8, Haar, sym8, coif4, and bior5.5), and the value of detail level is set from 1 to 6. We compared the experimental results with different combinations of parameters.

As shown in Figure 6, in terms of ECG dataset (ECG200, ECGFiveDays, and TwoLeadECG) and sensor dataset (DodgerLoopWeekend, SonyAIBORobotSurface1, and ItalyPowerDemand), the choice of parameters has little effect on

the results. Generally, the best prediction accuracy can be achieved after one level decomposition. Increasing the value of level leads to increasing the amount of calculation and may also cause a significant decrease in accuracy. In terms of image dataset (BeetleFly, Herring, and BirdChicken), the choice of parameters has significant influence on the results. For example, the highest accuracy is 0.9500 with Haar wavelet in level 2 on the BeetleFly dataset, the highest accuracy is 0.9500 with Haar wavelet in level 2 on the BeetleFly dataset, the

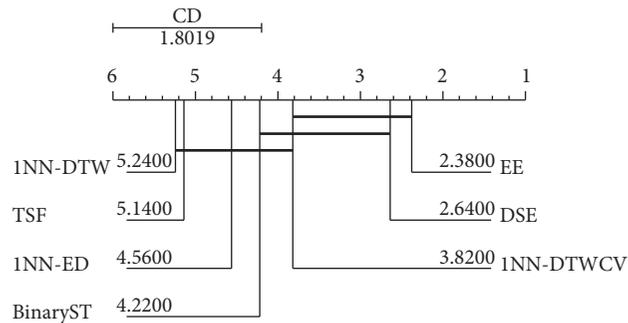


FIGURE 7: Critical difference diagram for seven classifiers derived from results in Table 3.

highest accuracy is 0.6562 with *coif4* wavelet in level 2 on the Herring dataset, and the highest accuracy is 1.0000 with Haar wavelet in level 3 on the BirdChicken dataset.

4.2.3. Comparison Result. Table 3 lists the classification accuracies of seven classifiers for 25 datasets. The last two lines of Table 3 represent the average rank of each classifier on different datasets and best performing times, respectively.

According to the results shown in Table 3, the EE is the best classifier, with an average rank of 2.38, and the best performance in 12 out of 25 problems. The performance of DSE proposed in this paper is slightly lower than the performance of EE. It wins on 8 out of 25 datasets and has the close average rank of 2.64 to the EE. The EE integrates a variety of distance measurement methods, and the DSE only uses Euclidean distance, which could lead to the little difference of performance between them. However, the DSE is still significantly more accurate than all the other alternatives, including BinaryST. This underlines the utility of decomposition on original time series data. The DWT is effective to improve the accuracy of shapelet transformation method.

When the significance level is 0.05 and the degree of freedom is $(6, 144)$, $F_F = 2.2781 > F_{0.05}(6, 144) = 2.162$. Therefore, given the significant level of 0.05, the original hypothesis is rejected, and the seven classifiers are significantly different. The critical difference diagram is shown in Figure 7. The critical difference for $\alpha = 0.05$ is 1.8019. Figure 7 depicts the superiority of the proposed method, and the EE and DSE have significantly a higher accuracy than the BinaryST, the TSF, the 1NN-DTW, and 1NN-ED on these datasets. The difference between the performance of DSE and the EE is not significant, relatively.

Based on the above analysis, the results show that the performance of the DSE method proposed in this paper is very close to the EE method and has higher accuracy and better stability than the other five compared classifiers.

5. Conclusions

In this study, an ensemble method by combining time frequency analysis and shape similarity recognition of time series is proposed to solve TSC problems. The proposed method embeds DWT into the shapelet-discovery algorithm

to produce a transformed data and then trains and tests base classifier on the transformed data; finally, the method implements a weighted majority voting on the results of base classifiers according to the correlation between components and original data. The experiment results indicate that the proposed method outperforms other methods in terms of accuracy. We also pay attention to the influence of parameter selection for the results and carry out study, which gives suggestions on the selection of mother wavelet and number of levels for different time series data types. According to the results in our experimental comparative studies, the proposed method is not only robust and efficient but can also be generalized for use in different application domains. However, the proposed method is still time-consuming. How to improve its efficiency will be considered in the next work.

Data Availability

The dataset used to support this study is the open dataset “UCR Time Series Classification Archive,” which is available at https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The research for this paper was supported by the Hubei Research Center for Educational Informationization (Central China Normal University).

References

- [1] R. Agrawal, C. Faloutsos, and A. N. Swami, “Efficient similarity search in sequence databases,” *Foundations of Data Organization and Algorithms*, vol. 730, no. 4, pp. 69–84, 1993.
- [2] E. Keogh and S. Kasetty, “On the need for time series data mining benchmarks: a survey and empirical demonstration,” *Data Mining and Knowledge Discovery*, vol. 7, no. 4, pp. 349–371, 2003.
- [3] F. Andrew, Siegel, *Practical Business Statistics*, pp. 431–466, Academic Press, Cambridge, MS, USA, 7th edition, 2016.

- [4] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, "Fast subsequence matching in time-series databases," *ACM SIGMOD Record*, vol. 23, no. 2, pp. 419–429, 2000.
- [5] D. J. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series," pp. 359–370, KDD, San Diego, CA, USA, 1994.
- [6] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognition*, vol. 44, no. 9, pp. 2231–2240, 2011.
- [7] S. M. M. Kahaki, M. Jan Nordin, A. H. Ashtari, and S. J. Zahra, "Deformation invariant image matching based on dissimilarity of spatial features," *Neurocomputing*, vol. 175, pp. 1009–1018, 2016.
- [8] S. M. M. Kahaki, M. J. Nordin, A. H. Ashtari et al., "Invariant feature matching for image registration application based on new dissimilarity of spatial features," *PLoS ONE*, vol. 11, no. 3, Article ID e0149710, 2016.
- [9] S. M. M. Kahaki, A. Haslina, N. M. Jan et al., "Geometric feature descriptor and dissimilarity-based registration of remotely sensed imagery," *Plos One*, vol. 13, no. 7, Article ID e0200676, 2018.
- [10] L. Ye and E. J. Keogh, "Time series shapelets: a new primitive for data mining," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Paris, France, June 2009.
- [11] L. Ye and E. Keogh, "Time series shapelets: a novel technique that allows accurate, interpretable and fast classification," *Data Mining and Knowledge Discovery*, vol. 22, no. 1-2, pp. 149–182, 2011.
- [12] J. Lines, L. M. Davis, J. Hills et al., "A shapelet transform for time series classification," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining ACM*, Beijing, China, August 2012.
- [13] H. Deng, E. G. Runger, and M. Vladimir, "A time series forest for classification and feature extraction," *Information Sciences*, vol. 239, no. 4, pp. 142–153, 2013.
- [14] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," vol. 29, pp. 565–592, Kluwer Academic Publishers, Dordrecht, Netherland, 2015.
- [15] A. Bagnall, J. J. Lines, and A. Bostrom, "Time-series classification with COTE: the collective of transformation-based ensembles," *IEEE Transactions On Knowledge And Data Engineering*, vol. 27, no. 9, pp. 2522–2535, 2015.
- [16] J. Lines, S. Taylor, and A. Bagnall, "HIVE-COTE: the hierarchical Vote collective of transformation-based ensembles for time series classification," in *Proceedings of the 2016 IEEE 16th International Conference on Data Mining*, (ICDM) IEEE, Barcelona, Spain, December 2016.
- [17] A. Bagnall, A. J. Lines, and E. J. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2016.
- [18] A. Bagnall, A. Bostrom, J. Large et al., "The Great Time Series Classification Bake off: An Experimental Evaluation of Recently Proposed Algorithms Extended Version," arXiv e-prints, 2016.
- [19] Y. Zhao, "R and Data Mining," pp. 75–87, Academic Press, Cambridge, MS, USA, 2013.
- [20] G. P. Nason, R. von Sachs, and G. Kroisandt, "Wavelet processes and adaptive estimation of the evolutionary wavelet spectrum," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 62, no. 2, pp. 271–292, 2000.
- [21] B. J. Whitcher, S. D. Byers, P. Guttorp, and D. B. Percival, "Testing for homogeneity of variance in time series: long memory, wavelets and the Nile River," *Water Resources Research*, vol. 38, no. 5, pp. 1054–1070, 2002.
- [22] Pimwadee, Chaovalit, Aryya et al., "Discrete wavelet transform-based time series analysis and mining," *Acm Computing Surveys*, vol. 43, no. 2, 2011.
- [23] H. A. Dau, E. Keogh, K. Kamgar et al., "The UCR time series classification archive," 2018, https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.
- [24] E. A. Maharaj, "Comparison and classification of stationary multivariate time series," *Pattern Recognition*, vol. 32, no. 7, pp. 1129–1138, 1999.
- [25] T. Rakthanmanon, B. Campana, A. Mueen et al., "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, London, UK, 2012.
- [26] A. Mueen, E. Keogh, and N. E. Young, "An expressive primitive for time series classification," in *Proceedings of ACM SIGKDD: International Conference on Knowledge Discovery and Data Mining*, pp. 1154–1162, San Diego, CA, USA, 2011.
- [27] M. H. Foo, J. J. Soraghan, and W. H. Siew, "Application of non-decimated discrete wavelet transform for partial discharge analysis," in *Proceedings of the International Conference & Exhibition on Electricity Distribution. IET*, Turin, Italy, 2005.
- [28] M. Misiti, Y. Misiti, G. Oppenheim et al., "Matlab Wavelet Toolbox User's Guide Version 3," The MathWorks, Natick, MS, USA, 2004.
- [29] T. Rakthanmanon and E. Keogh, "Fast shapelets: a scalable algorithm for discovering time series shapelets," in *Proceedings of the 2013 SIAM International Conference on Data Mining*, pp. 668–676, Austin, TA, USA, May 2013.
- [30] A. Bostrom and A. Bagnall, "Binary shapelet transform for multiclass time series classification," *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXXII*, vol. 32, pp. 24–46, 2017.
- [31] J. Grabocka, N. Schilling, M. Wistuba et al., "Learning time-series shapelets," in *Proceedings of the The 20th ACM SIGKDD conference on knowledge discovery and data mining*, New York, USA, 2014.
- [32] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, Hobokon, NY, USA, 2nd edition, 2001.
- [33] J. R. Quinlan, "C4. 5: programs for machine learning," *Morgan Kaufmann*, vol. 1, 1993.
- [34] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [35] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [36] M. Shatnawi, "Review of recent protein-protein interaction techniques," *Emerging Trends in Computational Biology, Bioinformatics, and Systems Biology*, vol. 12, no. 5, pp. 99–121, 2015.
- [37] D. Heckerman and D. Geiger, "The combination of knowledge and statistical data Mach Learn," *Chickering Learning Bayesian Networks*, vol. 20, pp. 197–243, 1995.
- [38] J. Demišar and D. Schuurmans, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, no. 1, pp. 1–30, 2006.