Research Article

Shapelet Discovery by Lazy Time Series Classification

Wei Zhang, Zhihai Wang, Jidong Yuan, and Shilei Hao

School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Jidong Yuan; yuanjd@bjtu.edu.cn

Received 16 January 2020; Revised 15 September 2020; Accepted 21 September 2020; Published 26 October 2020

Copyright © 2020 Wei Zhang 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.

As a representation of discriminative features, the time series shapelet has recently received considerable research interest. However, most shapelet-based classification models evaluate the differential ability of the shapelet on the whole training dataset, neglecting characteristic information contained in each instance to be classified and the classwise feature frequency information. Hence, the computational complexity of feature extraction is high, and the interpretability is inadequate. To this end, the efficiency of shapelet discovery is improved through a lazy strategy fusing global and local similarities. In the prediction process, the strategy learns a specific evaluation dataset for each instance, and then the captured characteristics are directly used to progressively reduce the uncertainty of the predicted class label. Moreover, a shapelet coverage score is defined to calculate the discriminability of each time stamp for different classes. The experimental results show that the proposed method is competitive with the benchmark methods and provides insight into the discriminative features of each time series and each type in the data.

1. Introduction

In recent years, massive time series data have been generated in many fields, including weather forecasting [1], malware detection [2], voltage stability assessment [3], human identification [4], and biomedicine [5]. Hence, the study of time series has been widely applicable, among which classification is an important research field. The classification issue of time series is the same as the traditional classification problem. We hope to find a function that can map any time series to a target class label. Although a large number of time series classification algorithms have been proposed, extensive experiments show that the 1NN classifier combining different distance metrics is still a competitive model in many problem areas [6–12]. In addition to the common Euclidean distance, alternatives have been proposed to measure the similarity between time series, including dynamic time warping (DTW) [13], weighted DTW (WDTW) [14], edit distance with real penalty (ERP) [15], time warp edit (TWE) [16], and move-split-merge (MSM) [17].

The improved distance function can help promote the performance of the nearest-neighbor model, but the 1NN classifier presents obvious drawbacks. This classifier cannot indicate the common characteristics of similar instances and the dissimilarity between different classes. In other words, its interpretability is insufficient. In reality, except for the accuracy, the features of distinct instances are our concern. These features provide a deeper understanding of data and improve the interpretability of the classification model. Unfortunately, time series usually have no definite features. Hence, various feature prototypes are proposed to mine potential patterns of time series [9, 18, 19]. Among them, the most classic is the local discriminative features "shapelet" [18].

A shapelet (as shown in Figure 1(a)) is a special discriminatory subsequence of time series, which is originally applied to construct the shapelet-based decision tree (SDT) [18] through recursively searching for the best shapelet in the training set. Since the shapelet can be used to establish an interpretable classification model, it has been widely studied [9]. Shapelet-based classification models can be divided into two categories. One type of the method utilizes the top-\(k\) shapelets to create a transformed dataset, on which the traditional classification algorithms [20–24] could be applied. The other uses the shapelets to build the classification model directly [18, 25–28].

One major problem that all shapelet-based approaches generally face is the massive size of the candidate shapelet
set. To solve this problem, researchers have put forward several methods. These methods can be roughly divided into four categories: (1) training instances are selected to generate the candidate shapelets. For example, Ji et al. [29] put forward a subclass splitting method to sample the training instances for candidate shapelet generation. (2) Heuristic shapelet search method. Grabocka et al. [27] presented a heuristic gradient descent shapelet search algorithm, which created a smaller candidate shapelet set. Rakthanmanon et al. [26] proposed a fast shapelet discovery algorithm based on Symbolic Aggregate approXimation (SAX). Similarly, Fang et al. [28] introduced a novel method to search shapelets based on piecewise aggregate approximation (PAA). (3) A random selection mechanism is used to select shapelets. Renard et al. [30] first proposed a random-shapelet algorithm to build the decision trees. Karlsson et al. [31] constructed the shapelet-based random forest, in which each decision tree is built based on instances and shapelets selected randomly. Further, to omit the shapelet threshold search, Shi et al. [32] have put forward the random pairwise shapelet forest. (4) Reformulating the shapelet search problem into a numerical optimization problem. For example, Hou et al. [33] treated the shapelet search task as a numerical optimization problem, and then the shapelets were learned by numerical analysis methods. Likewise, Wang et al. [34] designed a semisupervised shapelet learning model, which transforms the feature search problem into a joint optimization problem. Ma et al. [35] proposed an end-to-end model to learn the most discriminative shapelets by the gradient descent method. Zhao et al. [36] recently proposed a regularized shapelet learning framework to improve the shapelet learning efficiency. Although the above methods improve the classification efficiency of shapelet-based models to some extent, the vast majority of shapelet-based global classification models still has the following disadvantages:

(1) The shapelets captured by most shapelet-based models cannot adequately reflect the information of feature distribution and frequency of each class in the dataset. For example, due to the existence of intraclass variation, a few instances in different classes may have low-frequency discriminative features.

(2) The whole training set is generally applied for the discriminatory evaluation of candidate shapelets. Owing to the influence of the redundant instances and the intraclass variability, the extracted shapelets are merely the best on average for instances in the training dataset and cannot accurately reflect the local characteristics of the instance to be classified. In other words, the established shapelet-based model is not suitable or efficient for each test instance. Targeted evaluation strategies are not given enough attention.

To address these problems, we have first proposed a lazy shapelet-based model to capture the local features of each instance in the literature [37] (the literature [37] is a poster in the conference ICONIP 2018. In the conference manuscript, we simply proposed a lazy model to classify the instance based on its own local features. However, in this paper, the extended version further studies the fusion of global and local similarity, the local feature distribution and frequency information discovery, etc. In addition, this paper includes details about how to discover the local features for each instance to be classified in the shapelet-based model, how to use the local characteristics of each instance to determine the classwise discriminatory information, more experiments on parameter setting, statistical analysis, model comparison, and case studies). However, the proposed model still cannot get insight into the feature distribution and frequency information of the time series data. For that,
we significantly extend the research on the data-driven, shapelet-based model (lazy shapelet classification route, LSCR) to study the feature distribution and frequency information discovery. Here, the advantages of our model are interpreted in conjunction with Figure 1. From Figure 1(b), it can be found that the heterogeneous instances are different, and that there are also differences between the homogeneous instances. Since LSCR performs targeted analysis for each instance, compared with the SDT shapelet-based model (as shown in Figure 1(a)), our model may capture characteristics that SDT cannot discover. For example, the shapelet $S^G_0$ does not appear in the model built by SDT. Further, to evaluate the classwise discriminative feature frequency, the shapelet coverage score is defined. From Figure 1(c), we find that the scores can not only indicate the local discriminant intervals for different classes but also reflect their frequency information. For instance, the low-frequency discriminative interval $[0, 7]$ reflects the location of the local features detected by our model on a few instances of Class2, which are usually caused by intraclass variation and ignored by the global shapelet-based model. The main contributions of this paper are summarized as follows:

1. In contrast with the classical kNN or 1NN model based on global similarity, our model is a fusion of global and local similarities. For the consideration of global similarity, the instance selection strategy is used in evaluating the discrimination of shapelets. The smaller evaluation dataset can eliminate the interference of intraclass variation and improve the classification performance. In addition, local similarity is applied instead of global similarity for prediction, which makes the proposed model more interpretable.

2. To reduce the massive number of redundant candidate shapelets generated by the brute-force algorithm, a novel strategy is proposed for extracting candidate shapelets from the instance to be classified. This strategy can guarantee that the extracted shapelets accurately reflect the local characteristics of each test instance.

3. The shapelet coverage score of each sampling point is calculated to analyze the local characteristics of different classes in the dataset. Since the proposed model can efficiently analyze the local features of each instance, more accurate local characteristic information can be obtained. In particular, the classwise discriminative feature frequency and distribution information can be presented, which can help us to understand the data more comprehensively.

The remainder of the paper is organized as follows. Section 2 introduces related concepts and basic theories. Section 3 describes the proposed model and algorithm design in detail. Section 4 presents the experimental analysis. Section 5 offers the conclusion of this paper.
often need to judge the similarity between unequal time series. For example, in our work, we need to determine whether a time series contains a specific local feature through the distance between a subsequence and a whole time series.

**Definition 5** (similarity of unequal-length time series). Let\(\text{distul}(T, S)\) be a similarity function, in which time series \(T\) and \(S\) have different lengths. The function returns a non-negative optimal matching distance between two sequences as the degree of similarity. The distance between time series \(T\) and sequence \(S\) is
\[
\text{distul}(T, S) = \min(\text{dist}(S_i, S_j), S_i \in W_{[S]})
\]
where the symbol \(W_{[S]}\) represents the subsequence set with length \(|S|\) of time series \(T\) and \(|T|\) denote the length of sequences \(S\) and \(T\), respectively, and \(|S| \leq |T|\).

In our model, the distance between subsequences with equal length is calculated by the Euclidean distance, while the distance between complete time series is measured by the specified distance function.

In reality, different subsequences may have disparate discriminability. In our work, information gain is used to measure the discriminability of shapelets. To reduce distance computation and improve the shapelet discriminant property, the concept of the evaluation dataset is put forward.

**Definition 6** (shapelet evaluation dataset). The shapelet evaluation dataset is a specific subset of the training dataset, which is designed to evaluate the discriminability of local features of each test case.

**Definition 7** (entropy of dataset). The entropy of a given dataset \(D\) is calculated by the following formula:
\[
H(D) = - \sum_{i=0}^{c_i-1} p(D_{c_i}) \log(p(D_{c_i})).
\]
where \(c_i\) is an element in the class value set \(C\) of \(D\), \(D_{c_i}\) is the subset of instances with class \(c_i\) in \(D\), and the proportion \(p(D_{c_i})\) is calculated by
\[
p(D_{c_i}) = \frac{|D_{c_i}|}{|D|}.
\]

**Definition 8** (shapelet information gain). Given a shapelet \(S\) and a dataset \(D\) containing instances with different classes, the information gain of \(S\) is calculated as follows:
\[
\text{IG}(D, S) = \arg \max_{i} \left( H(D) - \frac{|D_{\text{left}}|}{|D|} H(D_{\text{left}}) \right)
\]
\[
- \frac{|D_{\text{left}}|}{|D|} H(D_{\text{right}}),
\]
where \(S\) is a split distance that can be applied to divide the dataset into two subsets: \(D_{\text{left}} = \{T_i | \text{distul}(T_i, S) \leq s\}\) and \(D_{\text{right}} = \{T_i | \text{distul}(T_i, S) > s\}\).

Finally, the maximum value of information gain is normally treated as the shapelet discrimination, and the corresponding distance \(s\) is taken as threshold \(\delta\). The split distance \(s\) usually takes the middle distance between two distance points. The detailed calculation process can be found in the literature [18]. In our model, the shapelet information gain calculated by equation (7) on a specific evaluation dataset reflects the reduction in uncertainty in the predicted class label of the test instance. Here, the mathematical description of our model is introduced.

Given a specific test instance \(T\), its uncertainty of predicted class in our model is
\[
U(T) = H(D_{\delta}(T)),
\]
where \(D_{\delta}(T)\) is the initial evaluation dataset of \(T\).

In the prediction process, the proposed model LSCR tries to progressively reduce the uncertainty by its own characteristics. The model can be formulated as
\[
\min_{s} \left( \text{IG}(T, D_{\delta}(T)) \right)
\]
\[
- \sum_{i=0}^{c_i-1} \text{IG}(D(T_{\delta}^i), S_i),
\]
where \(S_i\) denotes the \(i\)th element in the learned shapelet set \(S\) of \(T\).

Additionally, \(D(T_{\delta}^i)\) is the corresponding evaluation dataset of \(S_i\), which is determined by
\[
D(T_{\delta}^i) = \begin{cases} \delta \subset D(T_{\delta-1}), & i = 0, \\ \{T' | \text{distul}(T', S_{i-1}) \leq S_{i-1}, \delta, T' \in D(T_{\delta-1}), & i > 0. \end{cases}
\]

Finally, the main class property of the dataset \(D(T_{\delta}^i)\) would be taken as the predicted class value of \(T\). Generally, there will be only one type of instance left in the final evaluation dataset.

3. **Targeted Shapelet Extraction Technique**

3.1. **Overview.** To provide a brief introduction to the model LSCR, a schematic diagram is first presented in Figure 2. As determined from the figure, the evaluation dataset \(D_{\delta}\) for the test instance \(T\) is first generated; second, the candidate shapelets from \(T\) are evaluated on \(D_{\delta}\), and then the best shapelet is employed to exclude the instances that do not contain the local feature represented by \(S_i^j\); third, the optimal shapelet on the dataset \(D_{\delta}\) is continually searched until termination. Generally, there will be only one class of instances left in the final dataset, and the class will be taken as the predicted value.

The predicted result may be different from the class value of the nearest neighbor or the majority class in the initial dataset. This is the greatest difference between our model
3.2. Building Shapelet Evaluation Dataset. From the perspective of information theory, the purpose of extracting the best shapelet is to minimize the uncertainty of the class label of the test instance. The uncertainty is reflected in the class distribution of each subdataset generated based on whether the instance contains the specific feature. Hence, the more unbalanced the distribution of the subset of instances selected based on the shapelet, the more discriminative the feature. For example, for a binary classification problem, it is ideal to use the shapelet to divide the dataset containing instances with different class values into two subsets, each of which contains only one type of instance.

For the large-scale dataset, the running time of the searching shapelet is unbearable, so we attempt to sample the training instances for shapelet evaluation. In this paper, instances are selected based on the neighbor distance and the class value of the closest neighbor. Moreover, if the nearest-neighbor instances in the neighborhoods corresponding to the initial node of the classification path belong to the same class, then the route degrades into a single node. In particular, when the size of neighborhoods is set to 1, the model degenerates into 1NN, and the discriminant feature cannot be extracted effectively. In view of these problems, we propose to build a small targeted subset that contains instances with different classes for the instance to be classified. The subset ensures that the distinguishing nature of the local features can be evaluated. In addition, the data sampling strategy can eliminate the impact of intraclass variation on the local feature discriminant evaluation.

As shown in Algorithm 1, according to the class value of the nearest-neighbor instance, we select \( k \) identical and different instances for the test instance to construct a shapelet evaluation dataset (lines 2–6).

3.3. Finding the Optimal Shapelet. To reduce the computational complexity of extracting the best shapelet and make the extracted shapelets better reflect the characteristics of the test instance, a data-driven shapelet search algorithm is further proposed to find the best shapelet. We only search the best shapelets from the subsequence space of the instance to be classified so that the extracted shapelet accurately reflects the local features of each test instance. The process of generating the candidate shapelets collection for the test instance is given in Algorithm 2. In the algorithm, each subsequence of \( T \) with starting point \( i \) and length \( j \) constitutes the candidate shapelet set (line 4).

The candidate set corresponding to each node in the model SDT contains \( O(nm^2) \) candidate shapelets, where \( n \) is the number of time series and \( m \) is the length of each time series. In our model, we only consider the subsequences of test instance \( T \), so there are only \( O(m^2) \) candidate shapelets to be evaluated for each node. Our work reduces the size of candidate shapelet collection of each node by one order of magnitude.

The purpose of extracting shapelets from time series is to classify the time series using the discriminator. The discriminability of shapelets provides us with a way to explain the classification results. Algorithm 3 introduces the method for finding the best shapelet for the instance to be classified on the evaluation dataset. First, the candidate shapelet set is generated for \( T \) (line 3). Then, each candidate is evaluated by equation (7) to search for the best shapelet (lines 4–8).

The time complexity of our model to find the optimal feature is \( O(km^3) \), while that of the brute-force search algorithm in SDT is \( O(nm^4) \). Considering that \( k \) is smaller than \( n \), the calculation in the progress of searching for the best shapelet is significantly less. In addition, two ways to improve the efficiency of shapelet search are used [18, 25]: the early abandonment mechanism and the shapelet entropy pruning strategy.

3.4. Lazy Shapelet Classification Algorithm. A classification route based on shapelets for each instance to be classified is built through Algorithm 4.

Algorithm 4 mainly consists of five steps. First, the targeted shapelet evaluation dataset is established for \( T \) (line 1). Second, the candidate shapelet set is generated for \( T \) (line 2). Third, the.
model searches for the best shapelet on the evaluation dataset (line 3) and judges whether the termination condition is satisfied (line 4). It does not meet the termination condition at the initial node; that is, it will not degrade to the single node route. Generally, a best shapelet $S$ is found. Fourth, the extracted shapelet $S$ is applied to update the evaluation dataset for the child node (lines 7–8). Only the training instances whose distances are not greater than the split threshold are selected to form the subdataset. Fifth, the model repeats steps 2–8 until the termination condition is satisfied (line 10). Last, the shapelet-based classification route for $T$ is returned (line 11).

### 3.5. Computing Shapelet Coverage Score

**Definition 9 (shapelet coverage).** Shapelet coverage refers to the corresponding time interval of a given shapelet $S$. If a time stamp $t$ falls within the shapelet coverage of $S$, then we state that $t$ is covered by $S$.

In our work, the discriminatory score is calculated for each time stamp based on the coverage intervals of all obtained shapelets. First, an indicator function is presented to determine whether a time stamp $t$ is covered by a given shapelet $S$:
Input: the dataset of correctly predicted instances: $D_{\text{correct}}$, the set of class values: $C$, the length of time series: $m$.
Output: the shapelet coverage scores: $\text{Scores}$. 
(1) Initialization of a $|C| \times m$ matrix $\text{Scores}$ 
(2) for each $i = 1$ to $|C|$ do 
(3) Select the instances with class $c_i$ from $D_{\text{correct}}$ to form the dataset $D_{\text{correct}}^i$ 
(4) for each $T$ in $D_{\text{correct}}^i$ do 
(5) Get the shapelet set $S_T$ of $T$ on its classification route learned by Algorithm 4 
(6) end for 
(7) Calculate the shapelet coverage score $\text{Scores}$ based on equation (12) 
(8) end for 
(9) return $\text{Scores}$

Algorithm 5: ComputeScore ($D_{\text{correct}}, C, m$).

Here, the algorithm of computing the shapelet coverage scores for different classes in the dataset will be introduced. In Algorithm 5, to compute the shapelet coverage score for each class, the dataset $D_{\text{correct}}^i$ of the correctly predicted instances with the specific class is first obtained (lines 2-3). Then, all shapelets captured by LSCR for every instance in $D_{\text{correct}}^i$ are collected (lines 4-6). Finally, the scores reflecting the discriminability of each time stamp for every class are calculated based on equation (12) (line 7).

4. Experiments

Experimental analyses are conducted on 20 datasets from the UCR time series repository [38], most of which are frequently used for evaluation of shapelet-based models [9, 21, 22, 24, 27]. The experimental data are divided into training and test parts. The former part is used to build the model, while the latter part is applied to calculate the classification accuracy. The information of datasets is listed in Table 1, including train (size of the training set), test (size of the test set), max $\_\_k$ (the minimum number of instances of a class in the training set; that is, the maximum value that parameter $k$ can take.), length (the length of time series), and classes (the number of classes).
4.1. Parameter k Analysis. To study the effect of the size of the evaluation dataset on the discriminative evaluation of shapelets, the accuracy trends of LSCR DTW within the specified range over 10 datasets are first analyzed as a representative. Then, the average accuracy curves of 5 fusion models (LSCR DTW, LSCR ERP, LSCR ED, LSCR TWE, and LSCR MSM) are presented for parameter setting. Figure 3 shows the sensitivity of prediction results of different datasets to parameter $k$. From Figure 3(a), it can be seen that most accuracy rates on 5 binary-class datasets reach the maximum values when parameter $k$ is 5, and then all of the curves show a significant downward trend. From Figure 3(b), except for the MiddlePhalanxOutlineAgeGroup, the accuracy rates on 4 multiclass datasets exhibit a growth trend as $k$ increases in the previous stage. When $k$ is greater than 6, the accuracy on each dataset tends to be stable or decrease. The above experimental results suggest that the accuracy of LSCR generally varies regularly with $k$. Therefore, we propose to set $k$ based on the trend of average accuracy.

In Figure 4(a), it can be seen that, on binary-class datasets, the average accuracy variations of 5 fusion models show two significantly different trends. One shows a trend of increasing and gradually becoming stable, while the other presents a significant decline in accuracy after passing the inflection point. In Figure 4(b), on multiclass datasets, the average accuracies of all fusion models first increase and then decrease slowly after reaching the peak. Finally, in our work, the value $k$ corresponding to the highest average accuracy is set as the final parameter of LSCR dist on binary-class and multiclass datasets, respectively. See Table 2 for specific settings of the 5 fusion models.

In addition, the effect of instance selection on model performance is interpreted in Figure 5. In the scatter diagram, each point stands for a dataset. The more points there are in the figure that fall below the diagonal line, the better the performance of LSCR DTW. Since the targeted evaluation dataset is very helpful to improve the feature quality, the proposed shapelet evaluation strategy can significantly improve the performance of our model. As shown in Figure 5, LSCR DTW outperforms LSCR DTW (whole) on almost all datasets. The accuracy rates of the above two models are listed in Table 3.

4.2. Fusion Strategy Analysis. In this section, the effectiveness of the fusion strategy of global and local similarities is analyzed. Table 3 presents the accuracies of 5 1NN models combined with different distance functions (DTW, ERP, ED, TWE, and MSM) and their corresponding five fusion models. The accuracies of the 1NN models are taken from the website [38].

Figure 6 shows a critical difference diagram studied in literature [39, 40] for the 10 classification models on 20 datasets. This diagram is used for the overall test of significance of average ranks and can group models without significant differences into cliques. From the figure, we find that there are no significant differences in the performance of the 10 models on the 20 datasets, but the rankings of all fusion models are better than those of the corresponding 1NN models. This result suggests that the fusion model can effectively improve the classification performance of the 1NN model to some extent. Since LSCR DTW ranks first, it will be used for further analysis in the following.

4.3. Performance Analysis. In this section, the proposed model is compared with shapelet-based classifiers and 1NN classifiers combining with many commonly used distance functions and deep learning models.
4.3.1. Comparison with Shapelet-Based Classifiers. Since the classification process of the model proposed is similar to the decision tree, for the sake of fairness, all shapelet-based comparison models choose the decision tree to make predictions. Our model is first compared with SDT, fast shapelet tree (FS) [26], and C4.5 model [41] based on two respective shapelet transform algorithms. The symbol SSC4.5 denotes the C4.5 model based on the transform algorithm proposed by Yuan et al. [22], and STC4.5 represents the C4.5 model based on the other one proposed by Lines et al. [20]. The shapelet length ranges from 3 to the full length of time series and increases by 1 each time. All experimental results are:

![Accuracy plots](a)  
![Accuracy plots](b)  
![Time plots](c)  

Figure 3: The accuracy and time variations of the model LSCR_{DTW} on 5 binary-class and 5 multiclass datasets with the value $k$ increasing.

![Average accuracy plots](a)  
![Average accuracy plots](b)  

Figure 4: The average accuracy variations of the model LSCR_{dist} combined with different distance functions on (a) 5 binary-class and (b) 5 multiclass datasets with the parameter $k$ increasing.

4.3.1. Comparison with Shapelet-Based Classifiers. Since the classification process of the model proposed is similar to the decision tree, for the sake of fairness, all shapelet-based comparison models choose the decision tree to make predictions. Our model is first compared with SDT, fast shapelet tree (FS) [26], and C4.5 model [41] based on two respective shapelet transform algorithms. The symbol SSC4.5 denotes the C4.5 model based on the transform algorithm proposed by Yuan et al. [22], and STC4.5 represents the C4.5 model based on the other one proposed by Lines et al. [20]. The shapelet length ranges from 3 to the full length of time series and increases by 1 each time. All experimental results are:
listed in Table 4, and the accuracies of FS are obtained from the website [38]. In addition, the last line in Table 4 provides the average accuracy of each model over 20 datasets.

As observed from Figure 7, it is clear that although there is no significant difference between LSCRDTW and the other 4 shapelet-based models, the average rank of LSCRDTW is the best.

Figure 8 presents the scatter plot of accuracy comparison between LSCRDTW and the 4 classical shapelet-based classifiers. Figure 8(a) shows that LSCRDTW is better than SDT (14 of 20) over the 20 datasets. From Table 4, we find that compared with that of SDT built on the entire training set, the accuracies of LSCRDTW on datasets MoteStrain, FacesUCR, SonyAIBORobotSurface2, etc., are significantly improved. In particular, on the dataset FacesUCR, the accuracy of LSCRDTW is 20% greater than that of SDT. In Figures 8(b)–8(d), it can be concluded that LSCRDTW is also better than STC4.5 (13 of 16), SSC4.5 (13 of 20), and FS (14 of

| Dataset             | LSCRDTW | DTW | LSCRERP | LSCRDTW | LSCRDTW | LSCRDTW | LSCRDTW | LSCRDTW | LSCRDTW | LSCRDTW |
|---------------------|---------|-----|---------|---------|---------|---------|---------|---------|---------|---------|
| ArrowHead           | 0.749   | 0.800 | 0.749   | 0.800   | 0.794   | 0.800   | 0.743   | 0.794   | 0.726   | 0.806   |
| Beef                | 0.733   | 0.667 | 0.667   | 0.667   | 0.700   | 0.667   | 0.733   | 0.660   | 0.700   | 0.467   |
| BeetleFly           | 0.800   | 0.650 | 0.700   | 0.750   | 0.750   | 0.800   | 0.700   | 0.700   | 0.600   | 0.500   |
| CBF                 | 0.991   | 0.994 | 0.993   | 0.998   | 0.984   | 0.852   | 0.989   | 0.991   | 0.997   | 0.969   |
| ECGFiveDays         | 0.863   | 0.797 | 0.905   | 0.806   | 0.880   | 0.797   | 0.879   | 0.829   | 0.900   | 0.891   |
| FaceFour            | 0.909   | 0.898 | 0.864   | 0.864   | 0.909   | 0.784   | 0.886   | 0.852   | 0.818   | 0.943   |
| FacesUCR            | 0.924   | 0.908 | 0.926   | 0.921   | 0.811   | 0.769   | 0.854   | 0.920   | 0.926   | 0.971   |
| GunPoint            | 0.880   | 0.913 | 0.893   | 0.947   | 0.887   | 0.913   | 0.873   | 0.953   | 0.927   | 0.973   |
| ItalyPowerDemand    | 0.948   | 0.955 | 0.935   | 0.955   | 0.948   | 0.955   | 0.948   | 0.948   | 0.932   | 0.944   |
| Lightning7          | 0.589   | 0.712 | 0.616   | 0.740   | 0.548   | 0.575   | 0.521   | 0.753   | 0.658   | 0.753   |
| MedicalImages       | 0.699   | 0.747 | 0.682   | 0.676   | 0.653   | 0.684   | 0.628   | 0.711   | 0.696   | 0.741   |
| MiddlePhalanxOutlineAgeGroup | 0.539 | 0.519 | 0.519 | 0.481 | 0.519 | 0.519 | 0.545 | 0.519 | 0.494 | 0.494 |
| MoteStrain          | 0.903   | 0.866 | 0.858   | 0.872   | 0.859   | 0.879   | 0.908   | 0.797   | 0.881   | 0.855   |
| Plane               | 1.000   | 1.000 | 1.000   | 1.000   | 1.000   | 1.000   | 1.000   | 1.000   | 1.000   | 1.000   |
| ProximalPhalanxOutlineAgeGroup | 0.810 | 0.785 | 0.815 | 0.795 | 0.800 | 0.785 | 0.805 | 0.805 | 0.727 | 0.790 |
| SonyAIBORobotSurface1 | 0.750 | 0.696 | 0.815 | 0.689 | 0.775 | 0.696 | 0.734 | 0.681 | 0.785 | 0.730 |
| SonyAIBORobotSurface2 | 0.843 | 0.859 | 0.864 | 0.823 | 0.877 | 0.859 | 0.846 | 0.853 | 0.885 | 0.871 |
| ToeSegmentation1     | 0.816   | 0.750 | 0.851   | 0.833   | 0.662   | 0.680   | 0.811   | 0.820   | 0.789   | 0.816   |
| ToeSegmentation2     | 0.900   | 0.908 | 0.915   | 0.923   | 0.823   | 0.808   | 0.938   | 0.785   | 0.892   | 0.754   |
| Wine                 | 0.611   | 0.611 | 0.611   | 0.611   | 0.574   | 0.611   | 0.593   | 0.574   | 0.648   | 0.593   |
| Average              | 0.813   | 0.802 | 0.809   | 0.807   | 0.788   | 0.767   | 0.802   | 0.794   | 0.804   | 0.798   |

Figure 5: Accuracy comparison between LSCRDTW combining instance selection for shapelet evaluation and LSCRDTW (whole) using the whole training set as the evaluation set on 20 datasets.

Figure 6: Accuracy comparison between LSCRDTW combining instance selection for shapelet evaluation and LSCRDTW (whole) using the whole training set as the evaluation set on 20 datasets.
Hence, Figure 8 demonstrates that LSCRDTW outperforms the existing shapelet-based decision tree model on the 20 datasets.

To compare the time complexity of LSCRDTW with other shapelet-based models, the changes in running times of 6 shapelet-based models with increasing number of instances and instance length are shown in Figure 9, respectively. LSCRDTW(single) represents the training time of a single model LSCRDTW for a specific test instance, while LSCRDTW represents the model running on the entire test set. In the experiment, a binary-class dataset with uniform distribution is designed to run the analysis models, and the size of the training set is always the same as that of the test dataset. For the first experiment, the sizes of training and test datasets
increase by 10 at a time. The length of all instances remains 100. For another, the instance length increases by 10 at a
time. The sizes of training and test sets are 100.

From Figures 9(a) and 9(b), we can determine that the
training time of LSCR_{DTW} (single) is not sensitive to the size
of the training set and is polynomial with respect to the
length of the time series. In particular, it is faster than the
current fastest shapelet-based model FS. Further, as ob-
served from Figures 9(c) and 9(d), with increasing training
set size and instance length, the time consumption gap
between SDT, STC4.5, and LSCR_{DTW} widens, while the gap
between LSCR_{DTW} and SSC4.5 is not obvious.

In conclusion, LSCR_{DTW} is an accurate and rapid
shapelet-based classification model, which is the basis of
learning feature distribution and frequency information.

4.3.2. Comparison with Various 1NN Classifiers. In this
section, except for the five distance functions listed above, we
further compare our model with several DTW variants,
including WDTW, complexity-invariant DTW (CID) [42],
and derivative distance_{DTW} (DD_{DTW}) [43]. The accuracies
for the above 1NN models are provided by the website [38]
(see Table 4).

Figure 10 demonstrates that our model exhibits no
significant difference from DTW and its variants, but the
average ranks of LSCR_{DTW} are the best. In Figure 11, it is
clear that LSCR_{DTW} is better than ED (16 of 20), TWE (14 of
20), ERP (13 of 20), MSM (12 of 20), DTW (11 of 20),
WDTW (14 of 20), CID (11 of 20), and DD_{DTW} (13 of 20).

In summary, LSCR_{DTW} is competitive with the 1NN
model that depends on global similarity. The comparison
results suggest that continuously narrowing the search space of class attributes based on local similarity is an effective way to make prediction.

4.3.3. Comparison with Deep Learning Models. Now, various deep learning models have been widely studied in the field of time series classification. Karim et al. [44] attempted to improve the univariate time series classification performance of fully convolutional neural networks (FCNs) by using the long short-term memory recurrent neural network (LSTM-RNNs) sub-modules and attention mechanism and proposed the excellent models LSTM-FCN and ALSTM-FCN. Furthermore, the authors applied the above two models to the multivariate time series classification problem [45] and studied the reasons why the two models have superior performance [46]. Fawaz et al. [47] have reviewed some deep learning models of time series classification. Here, our model is compared with LSTM-FCN and 9 deep learning models (ResNet, FCN, Encoder, MLP, Time-CNN, TWIESN, MCDCNN, MCNN, and t-LeNet) analyzed in the literature [47]. All the experimental results were obtained from the corresponding literature (see Table 5).

Figure 12 shows that, except for ResNet and FCN, LSTM-FCN is significantly better than other models, and that our model is significantly better than MCNN and t-LeNet. Among the 10 deep learning models, the average rank of our model is better than 7 of them. In Figure 13, it can be seen that LSCRDTW is better than Encoder (12 of 20), Time-CNN (12 of 20), MLP (12 of 20), TWIESN (13 of 20), MCDCNN (17 of 20), t-LeNet (19 of 20), and MCNN (20 of 20).

Generally, to improve performance, deep learning model tuning requires an enormous computational cost. To pursue the optimal accuracy rate, Fawaz et al. [48] even proposed the neural network ensemble model with 60 deep learning models, but it is still not better than the traditional ensemble model HIVE-COTE [49]. It is unfair to compare our model with the deep learning model based on accuracy alone. In addition to improving accuracy, we believe that the model interpretability and data comprehensibility require more attention. However, the existing feature extraction methods for time series usually cannot simultaneously obtain the feature distribution and frequency information. For the deep learning model, it is difficult to train a targeted model for the specific instance on a small dataset. Accordingly, in our work, a highly interpretable classification model based on the lazy learning strategy is built for each target instance, which can be applied to gain insight into the local feature
4.4. Interpretability. To demonstrate the stronger interpretability of our model, this section separately analyses LSCRDTW on a binary-class dataset, MoteStrain, and the CBF multiclass dataset.

4.4.1. MoteStrain Dataset. Sensing data in MoteStrain are originally collected to detect potential variables online in the sensor network [50]. The classification task on this dataset is to distinguish whether the sensor is used for humidity measurement or temperature measurement. The classification performance of our model is significantly better than that of the model used for comparison and is close to the current best classification result provided by Bagnall et al. [38] on this dataset.

To further investigate the proposed model, the shapelet decision tree built by SDT is shown in Figure 14, where the symbol \( S^4_{(0,0)} \) represents the shapelet extracted from the fourth training instance Train4 in the root node of the shapelet decision tree. As seen from Figure 14, there is only one shapelet in the decision tree, where \( d \) denotes the distance between the test instance and the shapelet and \( \delta \) represents the split threshold of the shapelet. Based on the shapelet decision tree, when the distance between the test instance and the shapelet corresponding to the root node is not greater than the split threshold, the class prediction value of the test instance is Class1. Otherwise, it is Class2.

Figure 15 shows six instances and their shapelets extracted through LSCR DTW. Our model can correctly

\[
\begin{array}{cccccccccc}
\text{Dataset} & \text{LSTM-FCN} & \text{MLP} & \text{FCN} & \text{ResNet} & \text{Encoder} & \text{MCNN} & \text{t-LeNet} & \text{MCDCNN} & \text{Time-CNN} & \text{TWIESN} \\
\hline
\text{ArrowHead} & 0.909 & 0.778 & 0.843 & 0.845 & 0.804 & 0.339 & 0.303 & 0.685 & 0.723 & 0.659 \\
\text{Beef} & 0.900 & 0.720 & 0.697 & 0.753 & 0.643 & 0.200 & 0.200 & 0.563 & 0.763 & 0.537 \\
\text{BeetleFly} & 0.950 & 0.870 & 0.860 & 0.850 & 0.745 & 0.500 & 0.500 & 0.580 & 0.890 & 0.730 \\
\text{CBF} & 0.998 & 0.872 & 0.994 & 0.995 & 0.947 & 0.332 & 0.332 & 0.820 & 0.957 & 0.890 \\
\text{ECGFiveDays} & 0.992 & 0.970 & 0.987 & 0.975 & 0.982 & 0.499 & 0.497 & 0.762 & 0.882 & 0.698 \\
\text{FaceFour} & 0.943 & 0.840 & 0.928 & 0.955 & 0.815 & 0.268 & 0.295 & 0.712 & 0.906 & 0.855 \\
\text{FacesUCR} & 0.929 & 0.833 & 0.946 & 0.955 & 0.874 & 0.153 & 0.143 & 0.756 & 0.869 & 0.644 \\
\text{GunPoint} & 1.000 & 0.927 & 1.000 & 0.991 & 0.936 & 0.513 & 0.493 & 0.867 & 0.932 & 0.961 \\
\text{ItalyPowerDemand} & 0.963 & 0.954 & 0.961 & 0.963 & 0.965 & 0.500 & 0.499 & 0.955 & 0.955 & 0.880 \\
\text{Lightning7} & 0.836 & 0.630 & 0.827 & 0.845 & 0.625 & 0.310 & 0.260 & 0.534 & 0.651 & 0.664 \\
\text{MedicalImages} & 0.801 & 0.721 & 0.779 & 0.770 & 0.734 & 0.514 & 0.514 & 0.640 & 0.676 & 0.649 \\
\text{MiddlePhalanxOutlineAgeGroup} & 0.813 & 0.531 & 0.553 & 0.569 & 0.579 & 0.188 & 0.571 & 0.585 & 0.566 & 0.581 \\
\text{MoteStrain} & 0.939 & 0.858 & 0.937 & 0.928 & 0.840 & 0.508 & 0.539 & 0.765 & 0.882 & 0.785 \\
\text{Plane} & 1.000 & 0.978 & 1.000 & 1.000 & 0.976 & 0.130 & 0.134 & 0.965 & 0.965 & 1.000 \\
\text{ProximalPhalanxOutlineAgeGroup} & 0.893 & 0.856 & 0.831 & 0.853 & 0.844 & 0.488 & 0.488 & 0.838 & 0.828 & 0.844 \\
\text{SonyAIBORobotSurface1} & 0.982 & 0.672 & 0.960 & 0.958 & 0.743 & 0.443 & 0.422 & 0.653 & 0.687 & 0.638 \\
\text{SonyAIBORobotSurface2} & 0.978 & 0.834 & 0.979 & 0.978 & 0.839 & 0.594 & 0.617 & 0.774 & 0.841 & 0.697 \\
\text{ToeSegmentation1} & 0.983 & 0.583 & 0.961 & 0.963 & 0.659 & 0.505 & 0.526 & 0.490 & 0.595 & 0.865 \\
\text{ToeSegmentation2} & 0.931 & 0.745 & 0.880 & 0.906 & 0.795 & 0.632 & 0.815 & 0.443 & 0.738 & 0.842 \\
\text{Wine} & 0.870 & 0.565 & 0.587 & 0.744 & 0.500 & 0.500 & 0.500 & 0.500 & 0.517 & 0.759 \\
\text{Average} & 0.930 & 0.787 & 0.876 & 0.890 & 0.792 & 0.406 & 0.433 & 0.694 & 0.791 & 0.759 \\
\end{array}
\]

Table 5: The classification accuracies of 10 deep learning models on 20 datasets.

Figure 12: The critical difference diagram for LSCRDTW and 10 deep learning models on 20 datasets.

distribution and frequency information. The following is a detailed introduction.
Figure 15: Six test instances from the MoteStrain dataset and their shapelets found by LSCR DTW. The three instances (a–c) are from Class1, and the other three (d–f) are from Class2. (a) Test11, (b) Test129, (c) Test189, (d) Test3, (e) Test41, and (f) Test55.

Figure 14: The shapelet decision tree and the shapelets found by SDT.

Figure 13: Accuracy comparison between LSCR DTW and 10 deep learning models.
predict the class properties of these six instances, while SDT fails. Meanwhile, it is obvious that there are not only significant differences between instances of different types but also intraclass variations among similar instances. For example, the differences among the three instances with the same class label, Test11, Test129, and Test189, are noticeable, while there are no obvious common features. However, in the shapelet decision tree built by SDT, only a shapelet is found, which is not sufficient to distinguish the two classes. In contrast to the illustration of the shapelet given in Figure 14, it is not difficult to find that the optimal shapelet obtained in the shapelet decision tree is not the most discriminatory feature for the test instances. This result verifies that the shapelets extracted from the entire dataset are the most discriminatory for each training instance in the average sense; this is also the reason for the poor performance of SDT on the MoteStrain dataset.

Since the characteristics of each test instance have been considered in our model, in light of this situation, we can achieve better classification results. In addition, based on the shapelet obtained by LSCR\textsubscript{DTW}, the prediction process of each instance can be explained. For example, as shown in Figure 15, the reason why the 11th test instance belongs to Class1 is that the local feature $S_{11}^{11}$ lies in its initial stage, while the local feature $S_{41}^{41}$ of the 41st test instance in the middle part determines its predicted class label.
Figure 16 displays the scores of shapelets coverage on the MoteStrain training and test datasets. It is obvious that the high score intervals on training and test sets are very similar. Since there are only 20 instances in the training set, the scores of shapelet coverage (as shown in Figure 16(a)) may not accurately reflect the local characteristics. However, we propose to directly evaluate the local characteristics of test cases, which can help us utilize large amounts of test data. In Figure 16(b), it can be seen that local features with different coverage frequencies of various classes have been detected from the test dataset, which cannot be captured by other shapelet-based models. For example, the intervals [9, 16] (low frequency), [44, 54] (medium frequency), and [59, 76] (high frequency) (as shown in Figure 16(b)) are three significant discriminative intervals of Class2, while the interval [21, 50] (high frequency) is the most discriminative part of Class1. Based on the proposed model, using more instances with accurate labels results in obtaining more accurate local feature information. The statistical information helps us acquire more comprehensive local characteristic information of time series data, such as feature distribution and frequency.

4.4.2. CBF Dataset. This section studies the multiclass dataset CBF, which contains three types: Cylinder, Bell, and Funnel. Figure 17 shows the coverage scores of the CBF test dataset. It can be determined that the interval [31, 75] covers the most discriminative intervals for all three classes. In addition, unlike Cylinder and Funnel, the local interval [0, 15] with relatively low coverage frequency is discriminative for Bell.

Next, four specific test instances are presented. As shown in Figure 18, for the multiclass dataset, our model can not only detect high-frequency shapelets (as shown in Figures 18(a)–18(c)) but also effectively capture low-frequency shapelets (as shown in Figure 18(d)). The proposed model LSCR_dist is helpful to make a targeted analysis of each category of data and each test instance.

5. Conclusions

Aiming at the problems of global shapelet-based models built on the whole training set, a data-driven model fusing global and local similarities is proposed. In the model, the shapelet discriminability is evaluated through a specific subdataset. A smaller evaluation dataset reduces the computational time and improves the quality of shapelets. Moreover, target learning for each instance helps us understand the prediction process clearly. For example, the shapelets extracted by our LSCR model can be directly used to explain what characteristics determine the class value of the test instance. Furthermore, the proposed shapelet coverage score is applied to accurately analyze the local feature information of each class, which provides comprehensive insight into data characteristics. In the future, the application of the model in specific fields will be further studied, including ECG detection and image contour feature discovery.

Data Availability

The time series data used to support the findings of this study have been deposited in the UEA and UCR Time Series Classification Repository (http://www.timeseriesclassification.com).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 61672086, 61702030, and 61771058) and Beijing Natural Science Foundation (no. 4182052).

References

[1] A. McGovern, D. H. Rosendahl, R. A. Brown, and K. K. Droegemeier, “Identifying predictive multi-dimensional time series motifs: an application to severe weather prediction,” Data Mining and Knowledge Discovery, vol. 22, no. 1-2, pp. 232–258, 2011.
[2] O. Patri, M. Wojnowicz, and M. Wolff, “Discovering malware with time series shapelets,” in Proceedings of the 50th Hawaii International Conference on System Sciences, pp. 6079–6088, Maui, HI, USA, January 2017.
[3] L. Zhu, C. Lu, and Y. Sun, “Time series shapelet classification based online short-term voltage stability assessment,” IEEE
[4] H. Zou, Y. Zhou, J. Yang, W. Gu, L. Xie, and C. J. Spanos, "Wi-Fi-based human identification via convex tensor shapelet learning," in *Proceedings of the 32th AAAI Conference on Artificial Intelligence*, pp. 1711–1718, New Orleans, Louisiana, USA, February 2018.

[5] C. Bock, T. Gumbesb, M. Moor, B. Rieck, D. Roquero, and K. Borgwardt, "Association mapping in biomedical time series via statistically significant shapelet mining," *Bioinformatics*, vol. 34, no. 13, pp. 438–446, 2018.

[6] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. Keogh, "Querying and mining of time series data," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1542–1552, 2008.

[7] A. Abanda, U. Mori, and J. A. Lozano, "A review on distance based time series classification," *Data Mining and Knowledge Discovery*, vol. 33, no. 2, pp. 378–412, 2019.

[8] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 565–592, 2015.

[9] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. 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, 2017.

[10] Y. Ye, J. Jiang, B. Ge, Y. Dou, and K. Yang, "Similarity measures for time series data classification using grid representation and matrix distance," *Knowledge and Information Systems*, vol. 60, no. 2, pp. 1105–1134, 2019.

[11] J. Yuan, A. Douzal-Chouakria, S. Varasteh Yazdi, and Z. Wang, "A large margin time series nearest neighbour classification under locally weighted time warps," *Knowledge and Information Systems*, vol. 59, no. 1, pp. 117–135, 2019.

[12] J. Yuan, Q. Lin, W. Zhang, and Z. Wang, "Locally slope-based dynamic time warping for time series classification," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1713–1722, Beijing, China, November 2019.

[13] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in *Proceedings of the 5th SIAM International Conference on Data Mining*, pp. 506–510, Newport Beach, CA, USA, April 2005.

[14] 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.

[15] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings of the 30th international Conference on Very Large Data Bases*, pp. 792–803, Toronto, Canada, August 2004.

[16] P. F. Marteau, "Time warp edit distance with stiffness adjustment for time series matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 306–318, 2008.

[17] A. Stefan, V. Athitsos, and G. Das, "The move-split-merge metric for time series," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1425–1438, 2013.

[18] L. Ye and E. Keogh, "Time series shapelets: a new primitive for data mining," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 947–956, Paris, France, June 2009.

[19] B. K. Iwana and S. Uchida, "Time series classification using local distance-based features in multi-modal fusion networks," *Pattern Recognition*, vol. 97, Article ID 107024, 2020.

[20] J. Lines, L. M. Davis, J. Hills, and A. Bagnall, "A shapelet transform for time series classification," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 289–297, Beijing, China, August 2012.

[21] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, "Classification of time series by shapelet transformation," *Data Mining and Knowledge Discovery*, vol. 28, no. 4, pp. 851–881, 2014.

[22] J. D. Yuan, Z. H. Wang, and M. Han, "A discriminative shapelets transformation for time series classification," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 28, no. 6, pp. 1–28, 2014.

[23] F. J. Baldán and J. M. Benítez, "Distributed FastShapelet transform: a big data time series classification algorithm," *Information Sciences*, vol. 496, pp. 451–463, 2019.

[24] W. Zalewski, F. Silva, A. G. Maletzke, and C. A. Ferrero, "Exploring shapelet transformation for time series classification in decision trees," *Knowledge-Based Systems*, vol. 112, pp. 80–91, 2016.

[25] A. Mueen, E. Keogh, and N. Young, "Logical-shapelets: an expressive primitive for time series classification," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1154–1162, San Diego, CA, USA, August 2011.

[26] T. Rakthanmanon and E. Keogh, "Fast shapelets: a scalable algorithm for discovering time series shapelets," in *Proceedings of the 13th SIAM International Conference on Data Mining*, pp. 668–676, Austin, TX, USA, May 2013.

[27] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning time-series shapelets," in *Proceedings of the 20th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, pp. 392–401, New York, NY, USA, August 2014.

[28] Z. Fang, P. Wang, and W. Wang, "Efficient learning interpretable shapelets for accurate time series classification," in *Proceedings of the 34th International Conference on Data Engineering*, pp. 497–508, Paris, France, April 2018.

[29] C. Ji, C. Zhao, S. Liu et al., "A fast shapelet selection algorithm for time series classification," *Computer Networks*, vol. 148, pp. 231–240, 2019.

[30] X. Renard, M. Rifqi, W. Erray, and M. Dety niecki, "Random-shapelet: an algorithm for fast shapelet discovery," in *Proceedings of 2015 IEEE International Conference on Data Science and Advanced Analytics*, pp. 1–10, Paris, France, October 2015.

[31] J. Karlsson, P. Papapetrou, and H. Boström, "Generalized random shapelet forests," *Data Mining and Knowledge Discovery*, vol. 36, no. 5, pp. 1053–1085, 2016.

[32] M. H. Shi, Z. H. Wang, J. D. Yuan, and H. Y. Liu, "Random pairwise shapelets forest," in *Proceedings of the 22th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 68–80, Melbourne, VIC, Australia, June 2018.

[33] L. Hou, J. T. Kwok, and J. M. Zurada, "Efficient learning of timeseries shapelets," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 1209–1215, Phoenix, AR, USA, February 2016.

[34] H. Wang, Q. Zhang, J. Wu, S. Pan, and Y. Chen, "Time series feature learning with labeled and unlabeled data," *Pattern Recognition*, vol. 89, pp. 55–66, 2019.

[35] Q. Ma, W. Zhuang, and G. Cottrell, "Triple-shapelet networks for time series classification," in *Proceedings of 2019 IEEE International Conference on Data Mining (ICDM 2019)*, pp. 1246–1251, Beijing, China, November 2019.
[36] H. Zhao, Z. Pan, and W. Tao, “Regularized shapelet learning for scalable time series classification,” Computer Networks, vol. 173, Article ID 107171, 2019.

[37] W. Zhang, Z. H. Wang, J. D. Yuan, and S. L. Hao, “An effective lazy shapelet discovery algorithm for time series classification,” in Proceedings of the 25th International Conference on Neural Information Processing (ICONIP 2018), pp. 46–55, Siem Reap, Cambodia, December 2018.

[38] A. Bagnall, J. Lines, W. Vickers, and E. Keogh, The UEA & UCR Time Series Classification Repository, 2019, http://www.timeseriesclassification.com.

[39] J. Demsar, “Statistical comparisons of classifiers over multiple data sets,” Journal of Machine Learning Research, vol. 7, no. 1, pp. 1–30, 2006.

[40] A. Benavoli, G. Corani, and F. Mangili, “Should we really use post-hoc tests based on mean-ranks?” The Journal of Machine Learning Research, vol. 17, no. 1, pp. 152–161, 2016.

[41] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, USA, 1993.

[42] G. E. A. P. A. Batista, E. J. Keogh, O. M. Tataw, and V. M. A. de Souza, “CID: An efficient complexity-invariant distance for time series,” Data Mining and Knowledge Discovery, vol. 28, no. 3, pp. 634–669, 2014.

[43] T. Górecki and M. Łuczak, “Using derivatives in time series classification,” Data Mining and Knowledge Discovery, vol. 26, no. 2, pp. 310–331, 2013.

[44] F. Karim, S. Majumdar, H. Darabi, and S. Chen, “LSTM fully convolutional networks for time series classification,” IEEE Access, vol. 6, pp. 1662–1669, 2018.

[45] F. Karim, S. Majumdar, H. Darabi, and S. Harford, “Multivariate LSTM-FCNS for time series classification,” Neural Networks, vol. 116, pp. 237–245, 2019.

[46] F. Karim, S. Majumdar, and H. Darabi, “Insights into LSTM fully convolutional networks for time series classification,” IEEE Access, vol. 7, pp. 67718–67725, 2019.

[47] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, “Deep learning for time series classification: a review,” Data Mining and Knowledge Discovery, vol. 33, pp. 917–963, 2019.

[48] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P. A. Muller, “Deep neural network ensembles for time series classification,” in Proceedings of the 2019 International Joint Conference on Neural Networks, pp. 14–19, Budapest, Hungary, July 2019.

[49] J. Lines, S. Taylor, and A. Bagnall, “Time series classification with HIVE-COTE: the hierarchical vote collective of transformation-based ensembles,” ACM Transactions on Knowledge Discovery from Data, vol. 12, no. 5, p. 52, 2018.

[50] 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/.