Information Sciences 612 (2022) 835–847

Contents lists available at ScienceDirect

Information Sciences

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

Fully convolutional networks with shapelet features for time series classification

Cun Ji [a,*], Yupeng Hu [b], Shijun Liu [c,*], Li Pan [c], Bo Li [d], Xiangwei Zheng [a]

[a] School of Information Science and Engineering, Shandong Normal University, Jinan, China
[b] School of Computer Science and Technology, Shandong University, Tsingtao, China
[c] School of Software, Shandong University, Jinan, China
[d] School of Software and Electrical Engineering, Swinburne University of Technology, Melbourne, Australia

ARTICLE INFO

Article history:
Received 23 August 2021
Received in revised form 1 September 2022
Accepted 4 September 2022
Available online 12 September 2022

Keywords:
Time series classification
Feature discovery
Shapelet feature
Fully convolutional network

ABSTRACT

In recent years, time series classification methods based on shapelet features have attracted significant research interest because they are interpretable. Although researchers have studied shapelet features for decades, the time complexity of the shapelet extracting process remains high, and the accuracy rates of their methods are not ideal. This study combines a fully convolutional network with shapelet features to address these problems. First, some discriminative subsequences are effectively selected as shapelet features. The original time series is then transformed into shapelet feature vectors. Finally, a fully convolutional network classifier is trained for the transformed vectors. Experimental results on various datasets demonstrate that the proposed method can achieve high accuracy and extract shapelet features more effectively.

1. Introduction

With the development of the Internet of Things, sensors have been widely used in various fields, such as engineering, business, and medicine. Sensors usually collect data periodically, and the collected data form time series.

Time series analysis, which aims to mine something meaningful from time series, has attracted tremendous research interest in recent years. Among time series analysis tasks, time series classification (TSC) is being actively researched because of widespread applications, such as electroencephalogram classification, gesture recognition, and fault diagnosis [1,2].

The goal of TSC is to predict the class label of an unlabeled time series as accurately as possible. Many TSC methods have been proposed over the past few years to achieve this goal [3–7]. These methods are categorized into three groups [3]: 1) distance-based methods, which classify time series according to a predefined similarity measure [3], such as Euclidean distance, dynamic time warping (DTW), or edit distance [8], 2) model-based methods, which assume that time series with the same labels are generated by a model and classify a new time series as the class of the closest-matching model [3,6], and 3) feature-based methods, which extract features from time series and classify time series based on the selected features [9].

In addition to accuracy, an explanation of the classification results is required in some real-life fields [10]. For example, time series classifiers should provide doctors with why an electroencephalogram is judged as atrial fibrillation [11]. The classifiers should be able to interpret the salient features that distinguish between atrial fibrillation and normal. Accordingly, the

* Corresponding authors.
E-mail addresses: jicun@sdnu.edu.cn (C. Ji), huyupeng@sdu.edu.cn (Y. Hu), lsj@sdu.edu.cn (S. Liu), panli@sdu.edu.cn (L. Pan), boli@swin.edu.au (B. Li), xwzhengcn@163.com (X. Zheng).

https://doi.org/10.1016/j.ins.2022.09.009
0020-0255/

popular interpretable feature (i.e., shapelet)-based TSC methods are the focus of this study. A shapelet feature is one discriminative time series subsequence, and it can most accurately represent the time series in a class [12,13]. Therefore, shapelet-based methods are interpretable.

More researchers are studying the shapelet and its variants because of their interpretability [2]. Although many shapelet-based methods have been presented, they face several challenges. (1) **Accuracy**. Shapelet features have been combined with traditional classification methods, such as support vector machines (SVMs), Bayesian networks, and random forests (RaFs). However, advanced deep learning classification methods are seldom combined with shapelet features, even though some researchers adopted deep learning methods to extract shapelets [14,15]. (2) **Efficiency**. Although shapelet-based TSC methods have interpretability, the time complexity of the shapelet mining process is very high [13]. One reason is that the number of shapelet candidates is huge because the number is quadratic to the time series length. Another reason is that evaluating shapelet candidates is very time-consuming. The efficient discovery of discriminative shapelets is another fundamental challenge for TSC [16].

This study combines a Fully Convolutional Network with Shapelet Features (FCN-SF) to address these challenges. First, FCN-SF outlines a series of guidelines with which the proposed methods can efficiently discover the most discriminative subsequences as shapelet features. FCN-SF then transforms the time series into shapelet feature vectors where each value represents the distance between one time series and the corresponding shapelet feature. Finally, a Fully Convolutional Network (FCN) classifier is trained on the transformed data.

The contributions of this study are as follows:

- A series of guidelines for extracting shapelets is presented. Following these guidelines, the discriminative shapelet features can be extracted quickly.
- We combine shapelet features with FCN. Accordingly, the classification accuracy is high, maintaining high interpretability.
- We conduct experiments to demonstrate the effects of our method on various datasets from the UEA & UCR Time Series Classification Repository [17]. The experimental results demonstrate that the proposed method is feasible and effective.

This paper is structured as follows. Section 2 briefly describes related studies. Section 3 introduces the proposed method, FCN-SF. Section 4 presents the experimental results, and our conclusions are provided in Section 5.

2. Related work

2.1. Categories of TSC methods

Many TSC methods have been proposed in recent decades. They can be divided into three categories [3]:

(1) **Distance-based TSC methods**. Distance-based TSC methods use a similarity measure between time series [6]. These similarity measures are usually exploited within one nearest neighbor (1NN) classifier. A 1NN classifier with DTW is usually selected as a strong benchmark because it is effective and difficult to outperform [18]. TSC methods in this category are simple, accurate, and robust. However, their major shortcoming is that similarity measures cannot provide an explanatory ability for classification results [19].
(2) **Model-based TSC methods**. Model-based TSC methods assume that the same underlying model generates time series in the same class. Auto-regressive models, hidden Markov models, and deep learning models [4] are commonly used in these methods. In recent years, researchers have tried to design some explaining models for TSC. Mercier et al. [20] introduced an explaining deep model, PatchX, that interpreted the TSC results based on local patterns. Ma et al. [21] built a deep learning model with built-in interpretability. This model learned sequences with the same length as interpretable prototypes. In these explaining deep models, the explanatory ability of these models is provided by features or prototypes, so the interpretation ability is not intuitive.
(3) **Feature-based TSC methods**. Feature-based TSC methods classify time series according to the extracted features. Compared with global features, some local features (such as shapelets or symbol words) have a more precise explanation for the classification results. There are two essential branches of interpretable TSC methods: shapelet-based [22–24] and dictionary-based [25,26]. Shapelet-based methods used discriminative subsequences to interpret classification results. In contrast, dictionary-based methods distinguish different classes by the frequency of symbols words [6]. The interpretability of shapelet-based methods is more intuitive in these two branches.

Consequently, this study focuses on shapelet-based TSC methods because of their intuitive interpretation ability.

2.2. Shapelet-based TSC methods

Since Ye and Keogh's introduction [12], shapelets have become an active focus of TSC research.

2.2.1. Acceleration strategies for extracting shapelet features

Although shapelet-based TSC methods are interpretable and accurate, these methods face one major problem: the shapelet feature extracting process is time-consuming. Researchers have proposed four types of acceleration strategies to solve this problem.

First, the shapelet selection process can be accelerated by reducing measurement complexity. The time complexity of shapelet evaluation can be reduced by subsequence distance early abandoning, admissible entropy pruning, distance calculation sharing, and parallelism [13,27]. Compared with information gain, Kruskal–Wallis and Mood's Median statistics can also accelerate the shapelet extraction process [28]. These acceleration strategies are generally used in combing with other types of strategies.

Second, shapelet extraction can be accelerated by random sampling. Renard et al. [29] introduced a shapelet random selection (RS) process into the decision tree construction. Gordon et al. [30] fast built the classification trees using SALSA-R (ShApLet SAmpling with Random order). Karlsson et al. [31] proposed a generalized random shapelet forest (gRSF) that randomly selects shapelets to construct an RaF. However, sampling has strong randomness, and the results of each training may vary considerably.

Third, some researchers have attempted to accelerate the shapelet extracting process by pruning shapelet candidates. Hills et al. [32] estimated the length of a shapelet before shapelet selection, and only subsequences meeting the length requirements are considered shapelet candidates. Grabocka et al. [33] introduced Scalable shapelet Discovery (SD) that pruned similar candidates. Some researchers have pruned shapelet candidates with special points [13]. Rakthanmanon et al. [34], Fang et al. [35], and Li et al. [16] pruned candidates based on symbolic aggregate approximation. However, pruning candidates without limitations may decrease classification accuracy.

Finally, the shapelet extracting process can be accelerated by learning. The shapelet features can be obtained by learning by converting the shapelet extraction process into an optimization problem. Grabocka et al. [23] first adopted this idea and learned shapelet features based on objective function optimization. Hou et al. [36] learned the position of a shapelet by adopting the generalized eigenvector method. Wang et al. [37] learned shapelet features by combining regularized least-square, shapelet regularization, spectral analysis, and pseudo-labels. Zhao et al. [38] proposed a new shapelet learning framework that combined a fused lasso regularization and least-squares loss function. Although shapelet learning methods are efficient, they require adjusting parameters for various datasets.

2.2.2. Classification strategies with shapelet features

When originally introduced, shapelet features are embedded within a binary decision tree [12]. Karlsson et al. [31] constructed an RaF with several random shapelet trees to obtain higher classification accuracy. In each internal node of decision trees, Shi et al. [39] used a pair of shapelet features from different classes instead of a single shapelet feature.

Lines et al. [32,22] proposed Shapelet Transformation (ST), which separated the shapelet extraction and the classification process, to improve classification accuracy. Accordingly, shapelet features can be combined with various advanced classifiers.

Based on ST, Ji et al. [40] adopt an XG-Boost classifier to classify the transformed shapelet feature vectors to improve TSC accuracy. Simultaneously, Yan et al. [41] introduced an ensemble method that obtained the final results through every shapelet-based classifier.

Although some researchers have obtained shapelet features using deep learning [14,15], few have adopted deep learning methods to classify the shapelet-transformed datasets.

3. The proposed method

FCN-SF is proposed in this study to extract discriminative time series subsequences as shapelet features effectively and achieve high TSC accuracy. As depicted in Fig. 1, there are three steps in FCN-SF:

Step 1. Shapelet feature extraction. This step effectively extracts discriminative time series subsequences as shapelet features.

Step 2. Shapelet transformation. This step produces transformed shapelet feature vectors, which use the distances between a time series and one shapelet feature as the corresponding feature values. After transformation, we can adopt more advanced classifiers.

Step 3. Classifier training. This step constructs an FCN as a classifier to achieve high accuracy.

In the following subsection, we describe each FCN-SF step in detail.

3.1. Shapelet feature extraction

This step effectively extracts discriminative subsequences as shapelet features. Accordingly, FCN-SF discoveries shapelet features according to the following three guidelines.

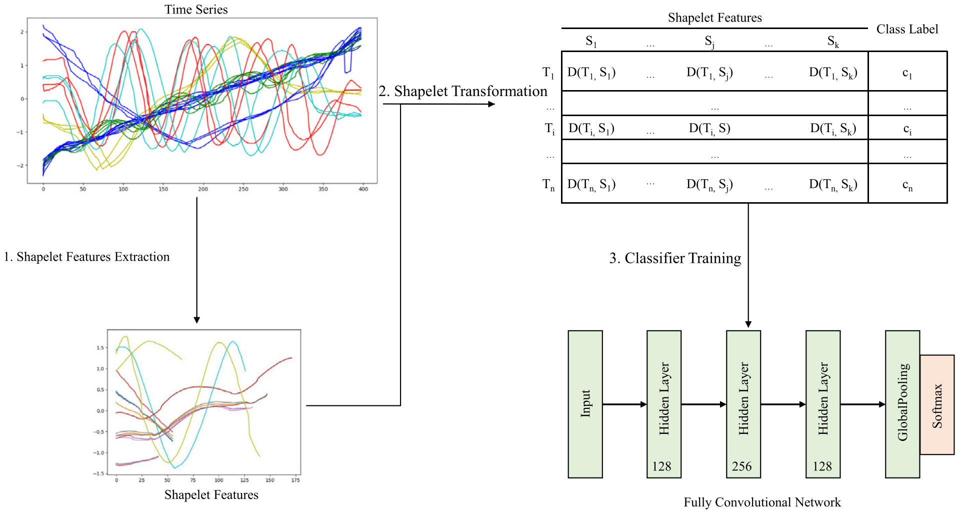

Fig. 1. Steps of FCN-SF: 1) shapelet feature extraction, 2) ST, and 3) classifier training.

Guideline 1. Extracting shapelet features from subsequences contains special points.

As discriminative subsequences, shapelets should be highly recognizable. Accordingly, FCN-SF filters out subsequences without obvious discrimination. For example, subsequence $S2$ in Fig. 2 is approximately a line segment. Therefore, it should be filtered out. In contrast, subsequence $S1$, which contains a special point, has higher discrimination and will be selected as a shapelet candidate.

Guideline 2. Extracting shapelet features from subsequences that use special points as endpoints.

As depicted in Fig. 3, $S1$ and $S3$ are two subsequences that satisfy Guideline 1. Furthermore, they are very similar. FCN-SF extracts shapelet features from subsequences that use special points as endpoints to reduce similar candidates.

Guideline 3. Extracting shapelet features from the representative time series.

In the training set, some time series instances are similar. FCN-SF selected the representative time series to represent a group of similar time series. As depicted in Fig. 4, two procedures are adopted to obtain the representative time series. First, FCN-SF divides the time series into similar groups using subclass splitting [42]. Second, the time series with the minimum sum Euclidean distance with the other time series is selected as the representative time series.

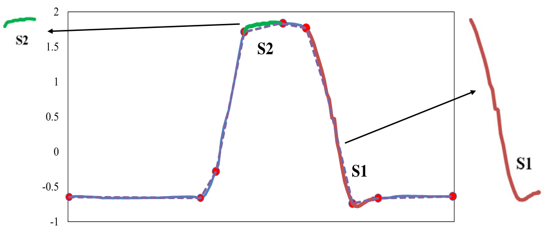

Fig. 2. An example of shapelet candidates selection.

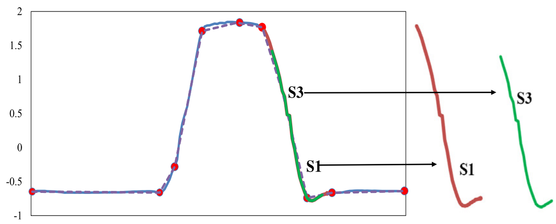

Fig. 3. An example of similar subsequences.

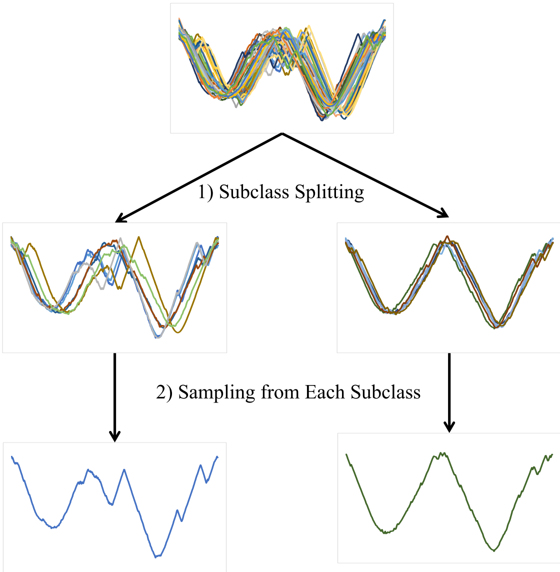

Fig. 4. Time series sampling.

With the guidelines described above, only a tiny number of discriminative subsequences are used as candidates in FCN-SF. The FCN then evaluates the candidates with information gain. The candidates with larger information gain are selected as shapelet features.

3.2. Shapelet transformation

FCN-SF fully exploits the interpretability of shapelet features by transforming time series instances into shapelet feature vectors, as defined by Eq. (1).

$$T' = \{dist(T, S_1), \cdots, dist(T, S_i), \cdots, dist(T, S_k)\} \tag{1}$$

In Eq. (1), $S_1, \cdots, S_i, \cdots, S_k$ are the k shapelet features selected by the previous step, T is the original time series, and the feature value $dist(T, S_i)$ is the distance between T and S_i.

Because the time series length is far longer than a shapelet, we calculated the distance $dist(T, S_i)$ between them using a sliding window, consistent with [12]. Fig. 5 illustrates the distance calculation process. In this process, a sliding window is used to obtain subsequences of T, which has the same length as shapelet S_i. Then, the distances between these subsequences and shapelet S_i are calculated. Finally, the minimum distances are the distance between T and S_i. This feature value $dist(T, S_i)$ calculating process can be formally expressed as Eq. (2). The sliding window is used to obtain the closest matching subsequence. Because Keogh and Lin [43] found that a sliding window can cause the meaningless problem, we will use the closest matching motif to replace the closest matching subsequence in the future.

$$dist(T, S_i) = min\{dist(T_1^{|S_i|}, S_i), \cdots, dist(T_j^{|S_i|}, S_i), \cdots, dist(T_{m-|S_i|+1}^{|S_i|}, S_i)\} \tag{2}$$

In Eq. (2) and Fig. 5, $|S_i|$ is the length of shapelet feature S_i. $T_j^{|S_i|}$ is a contiguous subsequence, the start position of $T_j^{|S_i|}$ is j, and the length of $T_j^{|S_i|}$ is $|S_i|$. In Eq. (2), $dist(T_j^{|S_i|}, S_i)$ is the Euclidean distance between $T_i^{|S_i|}$ and S_i.

The function of ST is presented in Tables 1 and 2. Table 1 is the original time series dataset, and Table 2 is the transformed dataset.

3.3. Classifier training

The last step of FCN-SF is training an FCN classifier for the transformed vectors to achieve high accuracy.

The structure of the FCN classifier is depicted in Fig. 6. The FCN classifier contains five layers:

- One input layer incorporates the transformed shapelet feature vectors (in the form depicted in Eq. (1))) into the classifier.
- Three hidden layers are designed to handle the input shapelet feature vectors. As depicted in Fig. 6, each hidden layer contain a convolutional layer that uses a batch normalization layer and *ReLU* activation function. Furthermore, a global average pool is used in each hidden layer to generate channel-wise statistics. As described in Fig. 6, the filter numbers for these three hidden layers are 128, 256, and 128. The corresponding kernel sizes are 8, 5, and 3.

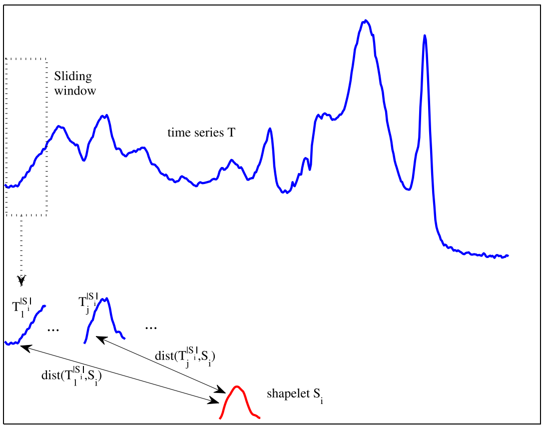

Fig. 5. Distance calculation diagram.

Table 1
Original time series dataset.

time series	class label
T_1	c_1
$\vdots$	$\vdots$
T_i	c_i
$\vdots$	$\vdots$
T_n	c_n

Table 2
The transformed dataset.

	shapelet features					class label
	S_1	$\cdots$	S_j	$\cdots$	S_k	
T'_1	$dist(T_1, S_1)$	$\cdot$	$dist(T_1, S_j)$	$\cdots$	$dist(T_1, S_k)$	c_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
T'_i	$dist(T_i, S_1)$	$\cdot$	$dist(T_i, S_j)$	$\cdots$	$dist(T_i, S_k)$	c_i
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
T'_n	$dist(T_n, S_1)$	$\cdot$	$dist(T_n, S_j)$	$\cdots$	$dist(T_n, S_k)$	c_n

- One output layer is responsible for predicting the final class label. As depicted in Fig. 6, the output layer uses the *softmax* function to obtain the final classification results.

4. Experiments and evaluation

4.1. Experimental setup

4.1.1. Datasets

In our experiments, we selected the 12 datasets from the UEA & UCR Time Series Classification Repository[1] [17]. The datasets were adopted by [11,13] and are commonly used datasets of the baseline methods.

Table 3 lists information about these datasets. The UEA & UCR Time Series Classification Repository splits each dataset into training and testing data. Like in most TSC research, the training data are used to construct TSC methods in our experiments, and the testing data are used to increase classification accuracy. The accuracy of dataset D can be calculated as Eq. 3, where N_{test} is the total number of testing data points, and N_c is the correctly classified number on the corresponding testing data point.

[1] UEA & UCR Time Series Classification Repository:http://www.timeseriesclassification.com

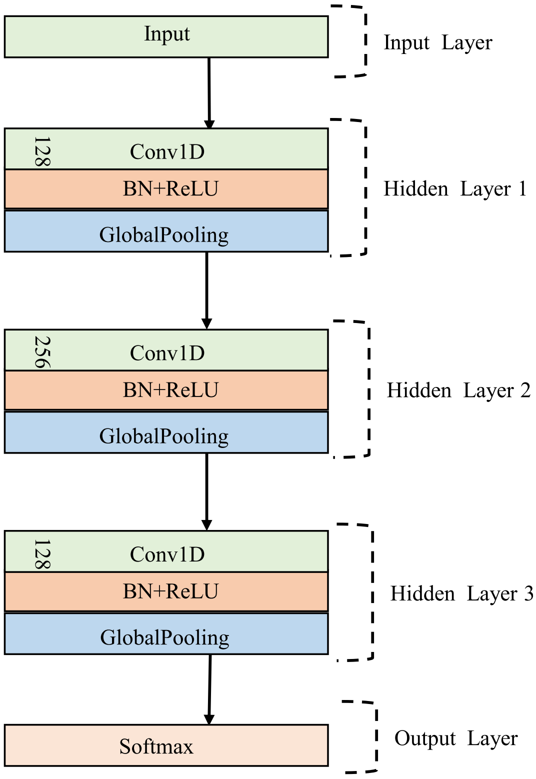

Fig. 6. The network structure of the FCN classifier.

Table 3
Datasets.

Datasets	Train Size	Test Size	Length	No. of Classes
Adiac	390	391	37	176
Beef	30	30	5	470
ChlorineConcentration	467	3840	3	166
Coffee	28	28	2	286
DiatomSizeReduction	16	306	4	345
ItalyPowerDemand	67	1029	2	24
Lightning7	70	73	7	319
MedicalImages	381	760	10	99
MoteStrain	20	1252	2	84
Symbols	25	995	6	398
Trace	100	100	4	275
TwoLeadECG	23	1139	2	82

$$Accuracy = \frac{N_c}{N_{test}} \quad (3)$$

4.1.2. Groups of experiments

We designed three groups of experiments to demonstrate that FCN-SF is effective and efficient:

- Comparison among classifiers on the transformed shapelet feature vectors. This experiment demonstrates that an FCN is more suitable for the transformed shapelet feature vectors.
- Comparison between FCN-SF and other shapelet-based methods. This experiment demonstrates that FCN-SF has higher accuracy than other shapelet-based methods, and FCN-SF extracts shapelet features the fastest among the methods with high accuracy.
- Comparison between FCN-SF and TSC baselines. This experiment demonstrates that FCN-SF has higher accuracy than the baselines.

4.1.3. Reproducibility

For reproducibility, we released our code and parameters on Github: 1) feature transform [2] based on [17], and 2) classification [3] based on [4]. Anyone can independently replicate the results.

4.2. Comparison among classifiers on the transformed shapelet feature vectors

This set of experiments adopts other classifiers to replace the FCN in the training step described in Section 3.3 to demonstrate that an FCN is most suitable for the transformed shapelet feature vectors. The alternative classifiers are decision tree (C4.5), 1NN, Naive Bayes(NB), Bayesian Networks (BN), Rotation Forest (RoF), RaF, SVM, and Weight Ensemble (Ensemble, [44]).

Table 4 summarizes the classification accuracy rates of all the classifiers. For each dataset, the highest accuracy rate is indicated in bold. The last row lists the average accuracy for each classifier.

Fig. 7 illustrates the critical difference diagram derived from Table 4 to distinguish the FCN from alternative classifiers.

Table 4 and Fig. 7 illustrate that an FCN is more accurate than other methods because it has the highest average accuracy and the lowest average rank; it is the superior method with most datasets.

We also compare the FCN with each alternative classifier directly. The comparison results are presented in Table 5. An FCN is more accurate for most datasets than the alternative classifiers.

4.3. Comparison between FCN-SF and other shapelet-based methods

4.3.1. Accuracy comparison

We contrast FCN-SF with the nine shapelet-based methods to demonstrate its higher accuracy: ST [22,32], LS [23], FS [34], SD [33], RS [29], SALSA-R [30], gRSF [31], FSS [13], and SSHVG [11]. The accuracy rates of these shapelet-based methods are from [11].

The classification accuracy rates of these methods are listed in Table 6. For each dataset, the highest accuracy rate is indicated in bold. The last row lists the average accuracy for each method.

Fig. 8 illustrates the critical difference diagram derived from Table 6 to distinguish FCN-SF from other shapelet-based methods.

As presented in Table 6 and Fig. 8, FCN-SF is more accurate than other shapelet-based methods because it has the highest average accuracy and the smallest average rank; it is the superior method with the most datasets.

We also compare FCN-SF with other shapelet-based methods directly. The comparison results are presented in Table 7. FCN-SF outperforms other shapelet-based methods on most datasets.

4.3.2. Shapelet extracting time comparison

The speedup rate (r) is adopted to evaluate the acceleration effect and obtained using Eq. (4). The speedup rate (r) is the ratio between the ST's shapelet extracting time (t_{ST}) and the corresponding acceleration method's shapelet extracting time (t_{cor}). A larger value of r indicates a greater acceleration effect.

$$r = \frac{t_{ST}}{t_{cor}} \tag{4}$$

Table 6 and Fig. 8 reveal that FCN-SF, SSHVG, ST, FSS, and LS have the highest accuracy. The speedup rates of these methods are presented in Table 8. FCN-SF and FSS are in the same column because they adopt the same acceleration strategies.

Table 8 demonstrates that the FCN-SF extracted shapelet features are fastest on most datasets.

4.4. Comparison between FCN-SF and TSC baselines

In this group of experiments, we compare our methods FCN-SF with the following TSC methods:

- 1NN-DTW: the common TSC baseline [6].
- MLP [45]: the deep learning baseline for TSC.
- BoP [25] and SAX-VSM [19]: two transformations methods based on Symbolic Aggregate approXimation (SAX) [46,47].
- BOSS [26], one transformation method based on symbolic Fourier approximation.
- ACF, one transformation method based on autocorrelation function [48].
- PS, one transformation method based on power spectrum [48].

The classification accuracy rates of these methods are listed in Table 9. For each dataset, the highest accuracy rate is indicated in bold. The last row lists the average accuracy of each method.

[2] https://github.com/sdujicun/FeatureTransform
[3] https://github.com/sdujicun/FCN_SF

Table 4
Classification accuracy rates among classifiers on the transformed shapelet feature vectors.

Datasets	**C4.5**	**1NN**	**NB**	**BN**	**RaF**	**RoF**	**SVM**	**Ensemble**	**FCN**
Adiac	0.637	0.476	0.701	0.678	0.754	0.783	0.414	**0.795**	0.706
Beef	0.533	0.767	0.800	0.700	0.767	0.800	0.667	**0.833**	**0.833**
ChlorineConcentration	0.563	0.573	0.359	0.584	0.649	0.680	0.573	0.643	**0.699**
Coffee	0.893	**1.000**	**1.000**	0.964	**1.000**	0.964	**1.000**	**1.000**	**1.000**
DiatomSizeReduction	0.676	0.918	0.794	0.948	0.899	0.866	0.709	0.856	**0.997**
ItalyPowerDemand	0.930	0.941	0.934	0.922	0.941	0.954	0.914	0.940	**0.970**
Lightning7	0.548	0.726	0.685	0.753	0.753	0.726	0.658	0.740	**0.781**
MedicalImages	0.571	0.703	0.516	0.530	0.695	0.703	0.678	0.712	**0.754**
MoteStrain	0.838	0.868	0.863	0.893	0.881	0.884	0.897	0.895	**0.909**
Symbols	0.734	0.932	0.841	0.917	0.889	**0.951**	**0.951**	**0.951**	0.945
Trace	0.990	0.950	**1.000**	**1.000**	**1.000**	**1.000**	0.990	**1.000**	**1.000**
TwoLeadECG	0.886	0.883	0.977	0.976	0.954	0.985	0.982	**0.986**	0.960
average	0.733	0.811	0.789	0.822	0.849	0.858	0.786	0.863	**0.880**

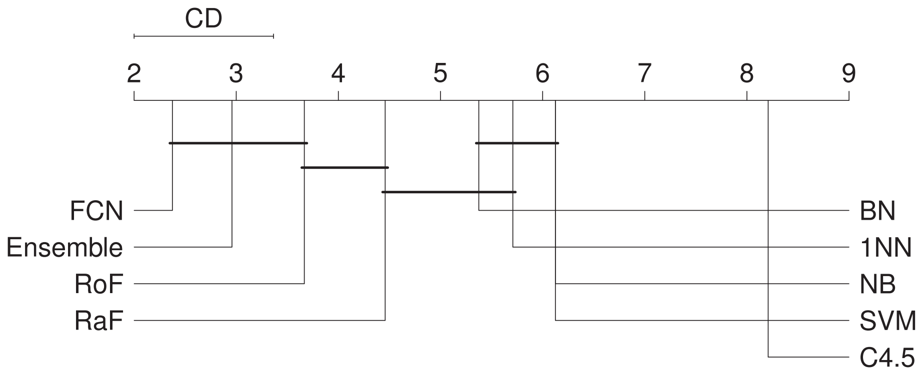

Fig. 7. Critical difference diagram for classifiers on transformed datasets.

Table 5
Comparison results between FCN and each alternative classifier.

	FCN Better	**Tier**	**FCN Worse**
C4.5 VS FCN	**12**	0	0
1NN VS FCN	**11**	1	0
NB VS FCN	**9**	2	1
BN VS FCN	**10**	1	1
RaF VS FCN	**9**	2	1
RoF VS FCN	**8**	1	3
SVM VS FCN	**9**	1	2
Ensemble VS FCN	**3**	3	3

Table 6
Classification accuracy among shapelet-based methods.

Datasets	**FCN-SF**	**ST**	**FSS**	**SSHVG**	**LS**	**gRSF**	**SALSA-R**	**FS**	**SD**	**RS**
Adiac	0.706	0.783	0.785	**0.790**	0.522	0.732	0.726	0.593	0.583	0.516
Beef	0.833	**0.900**	0.833	0.633	0.867	0.633	0.609	0.567	0.507	0.324
ChlorineConcentration	0.699	0.700	0.643	**0.710**	0.592	0.658	0.671	0.546	0.553	0.572
Coffee	**1.000**	0.964	**1.000**	0.964	**1.000**	0.964	0.960	0.929	0.961	0.769
DiatomSizeReduction	0.997	0.925	0.856	0.915	**0.980**	0.779	0.769	0.866	0.896	0.774
ItalyPowerDemand	**0.970**	0.948	0.940	0.951	0.960	0.944	0.951	0.917	0.920	0.924
Lightning7	0.781	0.726	0.740	0.740	**0.795**	0.726	0.695	0.644	0.652	0.635
MedicalImages	**0.754**	0.670	0.712	0.736	0.664	0.697	0.686	0.624	0.676	0.529
MoteStrain	0.909	0.897	0.895	**0.939**	0.883	0.952	0.854	0.777	0.783	0.815
Symbols	0.945	0.882	**0.951**	0.923	0.932	0.755	0.864	0.934	0.865	0.795
Trace	**1.000**	**1.000**	**1.000**	0.980	**1.000**	**1.000**	**1.000**	**1.000**	0.965	0.934
TwoLeadECG	0.960	**0.997**	0.986	0.974	0.996	0.991	0.958	0.924	0.867	0.914
average	**0.880**	0.866	0.862	0.855	0.849	0.819	0.812	0.777	0.769	0.708

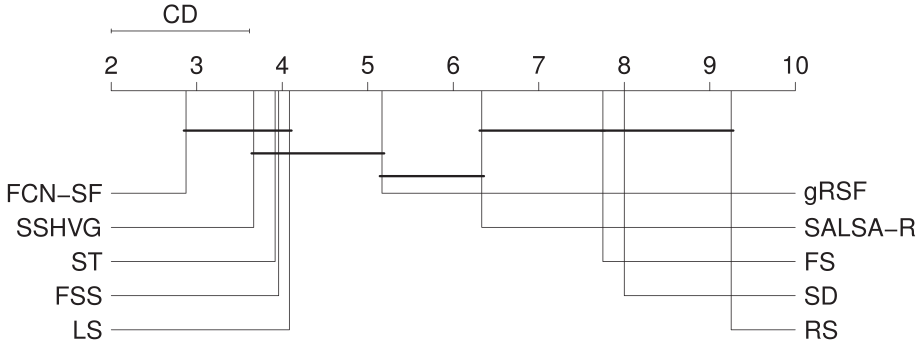

Fig. 8. Critical difference diagram for FCN-SF and other shapelet-based methods.

Table 7
Comparison results between FCN-ST and other shapelet-based methods.

	FCN-SF Better	Tier	FCN-SF Worse
ST VS FCN-SF	**7**	1	4
FSS VS FCN-SF	**6**	3	3
SSHVG VS FCN-SF	**8**	0	4
LS VS FCN-SF	**7**	2	3
gRSF VS FCN-SF	**8**	1	3
SALSA-R VS FCN-SF	**10**	1	1
FS VS FCN-SF	**11**	1	0
SD VS FCN-SF	**12**	0	0
RS VS FCN-SF	**12**	0	0

Table 8
Speedup rates among methods with the highest accuracy.

Datasets	FCN-SF/FSS	SSHVG	gRSF	LS	ST
Adiac	622.88	**2155.53**	69.62	0.02	1
Beef	388.82	**448.96**	29.35	0.30	1
ChlorineConcentration	**7285.34**	391.00	119.41	10.96	1
Coffee	**833.66**	759.85	25.94	0.58	1
DiatomSizeReduction	941.79	**2618.47**	15.15	0.20	1
ItalyPowerDemand	**419.19**	327.49	77.43	0.11	1
Lightning7	**1391.17**	140.67	66.51	0.46	1
MedicalImages	**5001.63**	3587.05	81.08	0.21	1
MoteStrain	**511.66**	293.48	1.85	0.13	1
Symbols	1025.51	**13086.86**	76.34	0.50	1
Trace	**846.44**	72.23	124.36	1.62	1
TwoLeadECG	**428.69**	408.74	2.08	0.13	1
fastest number	8	4	-	-	-

Table 9
Classification accuracy between FCN-SF and TSC baselines.

Datasets	1NN-DTW	MLP	BoP	SAX-VSM	BOSS	ACF	PS	FCN-SF
Adiac	0.604	**0.737**	0.588	0.453	0.765	0.665	0.752	0.706
Beef	0.633	0.600	0.600	0.433	0.800	0.733	0.733	**0.833**
ChlorineConcentration	0.648	**0.861**	0.641	0.654	0.661	0.662	0.722	0.699
Coffee	**1.000**	0.964	0.929	0.929	**1.000**	**1.000**	**1.000**	**1.000**
DiatomSizeReduction	0.967	0.964	0.922	0.882	0.931	0.889	0.873	**0.997**
ItalyPowerDemand	0.950	0.946	0.86	0.816	0.909	0.79	0.897	**0.970**
Lightning7	0.726	0.644	0.575	0.575	0.685	0.575	0.603	**0.781**
MedicalImages	0.737	0.705	0.493	0.508	0.718	0.678	0.662	**0.754**
MoteStrain	0.835	0.846	0.849	0.794	0.879	0.743	0.688	**0.909**
Symbols	0.950	0.757	**0.951**	0.622	0.967	0.897	0.799	0.945
Trace	**1.000**	0.840	0.970	**1.000**	**1.000**	**1.000**	0.950	**1.000**
TwoLeadECG	0.905	0.951	0.975	0.897	**0.981**	0.804	0.906	0.960
average	0.830	0.818	0.779	0.714	0.858	0.786	0.803	**0.880**

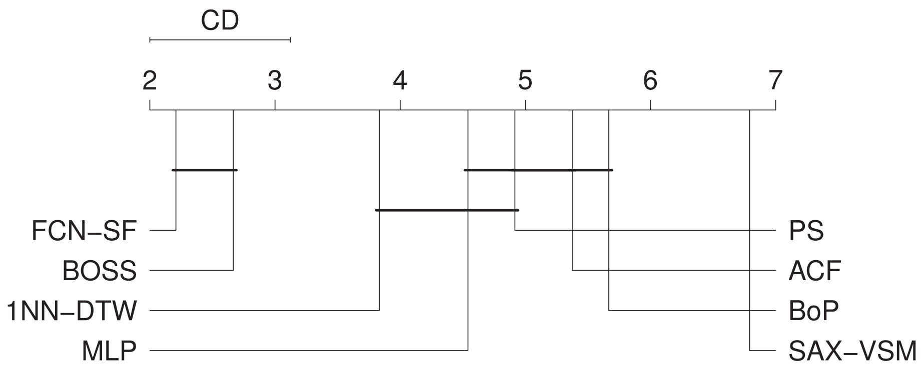

Fig. 9. Critical difference diagram for FCN-SF and baseline methods.

Table 10
Comparison results between FCN-ST and TSC baselines.

	FCN-SF Better	Tier	FCN-SF Worse
1NN-DTW VS FCN-SF	**9**	2	1
MLP VS FCN-SF	**10**	0	2
BoP VS FCN-SF	**10**	0	2
SAX-VSM VS FCN-SF	**11**	1	0
BOSS VS FCN-SF	**7**	2	3
ACF VS FCN-SF	**10**	2	0
PS VS FCN-SF	**8**	2	2

Fig. 9 illustrates the critical difference diagram derived from Table 9 to distinguish FCN-SF from the baselines.

We also compare FCN-SF with each baseline method directly. The comparison results are presented in Table 10. FCN-SF outperforms baseline methods on most datasets.

5. Conclusion

Shapelet-based TSC methods are being actively researched because of their interpretability. However, the time complexity of the shapelet extracting process is high, and the accuracy rates are not ideal. FCN-SF is introduced in this paper to address these challenges. FCN-SF extracts shapelet features with high accuracy using three acceleration guidelines, transforms the original time series into shapelet feature vectors, and trains a deep learning classifier for the transformed vectors. Accordingly, FCN-SF can fully exploit shapelet features and deep learning models. Experimental results on various datasets from the UEA & UCR Time Series Classification Repository demonstrate that the proposed method is more accurately and can effectively extract shapelet features. In the future, we will adjust the parameters of the FCN with some MLOps tools to obtain even higher accuracy. Furthermore, we will combine more advanced deep learning models with shapelet features.

CRediT authorship contribution statement

Cun Ji: Conceptualization, Methodology, Validation, Writing - original draft, Writing - review & editing. **Yupeng Hu:** Methodology, Validation. **Shijun Liu:** Supervision, Validation. **Li Pan:** Methodology, Validation. **Bo Li:** Methodology, Validation. **Xiangwei Zheng:** Supervision, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Special Project on Innovative Methods [Grant No. 2020IM020100], the Shandong Provincial Natural Science Foundation [Grant No. ZR2020QF112], and the project of Qingdao Postdoctoral Applied Research [Grant No. QDPostD20190901].

The authors would like to thank the anonymous reviewers and the editors for their insightful comments and suggestions, which are greatly helpful in improving the quality of this paper. The authors also thank Philip Pape, Ph.D., from Liwen Bianji (Edanz) (www.liwenbianji.cn) for editing the English text of a draft of this manuscript.

References

[1] R. Chen, X. Yan, S. Wang, G. Xiao, Da-net: Dual-attention network for multivariate time series classification, Information Sciences.
[2] C. Ji, M. Du, Y. Hu, S. Liu, L. Pan, X. Zheng, Time series classification based on temporal features, Applied Soft Computing 109494 (2022).
[3] A. Abanda, U. Mori, J.A. Lozano, A review on distance based time series classification, Data Mining and Knowledge Discovery 33 (2) (2019) 378–412.
[4] H.I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, Deep learning for time series classification: a review, Data Mining and Knowledge Discovery 33 (4) (2019) 917–963.
[5] Z. Zheng, Z. Zhang, L. Wang, X. Luo, Denoising temporal convolutional recurrent autoencoders for time series classification, Information Sciences 588 (2022) 159–173.
[6] A. Bagnall, J. Lines, A. Bostrom, J. Large, E. Keogh, The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances, Data Mining and Knowledge Discovery 31 (3) (2017) 606–660.
[7] L. Chen, D. Chen, F. Yang, J. Sun, A deep multi-task representation learning method for time series classification and retrieval, Information Sciences 555 (2021) 17–32.
[8] Z. Geler, V. Kurbalija, M. Ivanović, M. Radovanović, Weighted knn and constrained elastic distances for time-series classification, Expert Systems with Applications 162 (2020) 113829.
[9] Z. Xiao, X. Xu, H. Xing, S. Luo, P. Dai, D. Zhan, Rtfn: A robust temporal feature network for time series classification, Information Sciences 571 (2021) 65–86.
[10] K. Fauvel, E. Fromont, V. Masson, P. Faverdin, A. Termier, Xem: An explainable-by-design ensemble method for multivariate time series classification, Data Mining and Knowledge Discovery 36 (3) (2022) 917–957.
[11] C. Ji, Y. Hu, K. Wang, P. Zhan, X. Li, X. Zheng, Identifiable temporal feature selection via horizontal visibility graph towards smart medical applications, Interdisciplinary Sciences: Computational Life Sciences 13 (4) (2021) 717–730.
[12] L. Ye, E. Keogh, Time series shapelets: a new primitive for data mining, in, in: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2009, pp. 947–956.
[13] C. Ji, C. Zhao, S. Liu, C. Yang, L. Pan, L. Wu, X. Meng, A fast shapelet selection algorithm for time series classification, Computer Networks 148 (2019) 231–240.
[14] Q. Ma, W. Zhuang, S. Li, D. Huang, G.W. Cottrell, Adversarial dynamic shapelet networks, in: Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI Press, 2020, pp. 5069–5076.
[15] Y. Wang, R. Emonet, E. Fromont, S. Malinowski, R. Tavenard, Adversarial regularization for explainable-by-design time series classification, in: 2020 IEEE 32nd International Conference on Tools with Artificial Intelligence, IEEE, 2020, pp. 1079–1087.
[16] G. Li, B.K.K.C. Choi, J. Xu, S.S. Bhowmick, K.-P. Chun, L. Grace Wong, Efficient shapelet discovery for time series classification, IEEE Transactions on Knowledge and Data Engineering (2020), 1–1.
[17] A. Bagnall, J. Lines, W. Vickers, E. Keogh, The uea & ucr time series classification repository, URL www.timeseriesclassification.com.
[18] H. Li, J. Liu, Z. Yang, R.W. Liu, K. Wu, Y. Wan, Adaptively constrained dynamic time warping for time series classification and clustering, Information Sciences 534 (2020) 97–116.
[19] P. Senin, S. Malinchik, Sax-vsm: Interpretable time series classification using sax and vector space model, 2013 IEEE 13th international conference on data mining, IEEE, 2013, pp. 1175–1180.
[20] D. Mercier, A. Dengel, S. Ahmed, Patchx: Explaining deep models by intelligible pattern patches for time-series classification, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.
[21] D. Ma, Z. Wang, J. Xie, B. Guo, Z. Yu, Interpretable multivariate time series classification based on prototype learning, International Conference on Green, Pervasive, and Cloud Computing, Springer, 2020, pp. 205–216.
[22] J. Lines, L.M. Davis, J. Hills, A. Bagnall, A shapelet transform for time series classification, in: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2012, pp. 289–297.
[23] J. Grabocka, N. Schilling, M. Wistuba, L. Schmidt-Thieme, Learning time-series shapelets, in: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2014, pp. 392–401.
[24] Z. Liang, H. Wang, Efficient class-specific shapelets learning for interpretable time series classification, Information Sciences 570 (2021) 428–450.
[25] J. Lin, R. Khade, Y. Li, Rotation-invariant similarity in time series using bag-of-patterns representation, Journal of Intelligent Information Systems 39 (2) (2012) 287–315.
[26] P. Schäfer, The boss is concerned with time series classification in the presence of noise, Data Mining and Knowledge Discovery 29 (6) (2015) 1505–1530.
[27] F.J. Baldán, J.M. Benítez, Distributed fastshapelet transform: a big data time series classification algorithm, Information Sciences 496 (2019) 451–463.
[28] J. Lines, A. Bagnall, Alternative quality measures for time series shapelets, in: International Conference on Intelligent Data Engineering and Automated Learning, Springer, 2012, pp. 475–483.
[29] X. Renard, M. Rifqi, W. Erray, M. Detyniecki, Random-shapelet: an algorithm for fast shapelet discovery, in: 2015 IEEE International Conference on Data Science and Advanced Analytics, IEEE, 2015, pp. 1–10.
[30] D. Gordon, D. Hendler, L. Rokach, Fast and space-efficient shapelets-based time-series classification, Intelligent Data Analysis 19 (5) (2015) 953–981.
[31] I. Karlsson, P. Papapetrou, H. Boström, Generalized random shapelet forests, Data mining and knowledge discovery 30 (5) (2016) 1053–1085.
[32] J. Hills, J. Lines, E. Baranauskas, J. Mapp, A. Bagnall, Classification of time series by shapelet transformation, Data Mining and Knowledge Discovery 28 (4) (2014) 851–881.
[33] J. Grabocka, M. Wistuba, L. Schmidt-Thieme, Fast classification of univariate and multivariate time series through shapelet discovery, Knowledge and Information Systems 49 (2) (2016) 429–454.
[34] T. Rakthanmanon, E. Keogh, Fast shapelets: A scalable algorithm for discovering time series shapelets, in: Proceedings of the 2013 SIAM International Conference on Data Mining, SIAM, 2013, pp. 668–676.
[35] Z. Fang, P. Wang, W. Wang, Efficient learning interpretable shapelets for accurate time series classification, in: 2018 IEEE 34th International Conference on Data Engineering, IEEE, 2018, pp. 497–508.
[36] L. Hou, J. Kwok, J. Zurada, Efficient learning of timeseries shapelets, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016.
[37] H. Wang, Q. Zhang, J. Wu, S. Pan, Y. Chen, Time series feature learning with labeled and unlabeled data, Pattern Recognition 89 (2019) 55–66.
[38] H. Zhao, Z. Pan, W. Tao, Regularized shapelet learning for scalable time series classification, Computer Networks 173 (2020) 107171.
[39] M. Shi, Z. Wang, J. Yuan, H. Liu, Random pairwise shapelets forest, in: Proceedings of 22nd Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2018, pp. 68–80.
[40] C. Ji, X. Zou, Y. Hu, S. Liu, L. Lyu, X. Zheng, XG-SF: An XGBoost classifier based on shapelet features for time series classification, Procedia computer science 147 (2019) 24–28.
[41] L. Yan, Y. Liu, Y. Liu, Application of discrete wavelet transform in shapelet-based classification, Mathematical Problems in Engineering (2020).
[42] P. Sathianwiriyakhun, T. Janyalikit, C.A. Ratanamahatana, Fast and accurate template averaging for time series classification, in, in: Proceedings of the 2016 8th International Conference on Knowledge and Smart Technology, IEEE, 2016, pp. 49–54.

[43] E. Keogh, J. Lin, Clustering of time-series subsequences is meaningless: implications for previous and future research, Knowledge and information systems 8 (2) (2005) 154–177.
[44] A. Bagnall, L. Davis, J. Hills, J. Lines, Transformation based ensembles for time series classification, in: Proceedings of the 2012 SIAM international conference on data mining, SIAM, 2012, pp. 307–318.
[45] Z. Wang, W. Yan, T. Oates, Time series classification from scratch with deep neural networks: A strong baseline, 2017 International joint conference on neural networks, IEEE, 2017, pp. 1578–1585.
[46] J. Lin, E. Keogh, S. Lonardi, B. Chiu, A symbolic representation of time series, with implications for streaming algorithms, in: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, 2003, pp. 2–11.
[47] J. Lin, E. Keogh, L. Wei, S. Lonardi, Experiencing sax: a novel symbolic representation of time series, Data Mining and knowledge discovery 15 (2) (2007) 107–144.
[48] J. Lines, S. Taylor, A. Bagnall, Time series classification with hive-cote: The hierarchical vote collective of transformation-based ensembles, ACM Transactions on Knowledge Discovery from Data 12 (5).