Journal of King Saud University – Computer and Information Sciences 35 (2023) 101783

HOSTED BY

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Time series classification with random temporal features

Cun Ji [a,*], Mingsen Du [a], Yanxuan Wei [a], Yupeng Hu [b], Shijun Liu [b], Li Pan [b], Xiangwei Zheng [a]

[a] School of Information Science and Engineering, Shandong Normal University, Jinan, China
[b] School of Software, Shandong University, Jinan, China

A R T I C L E I N F O

Article history:
Received 14 July 2023
Revised 28 August 2023
Accepted 25 September 2023
Available online 28 September 2023

Keywords:
Time series classification
Random feature
Temporal feature
Feature selection
Feature importance measures

A B S T R A C T

Time series classification exists in widespread domains such as EEG/ECG classification, device anomaly detection, and speaker authentication. Although many methods have been proposed, efficient selection of intuitive temporal features to accurately classify time series remains challenging. Therefore, this paper presents TSC-RTF, a new time series classification method using random temporal features. First, to ensure the intuitiveness of the features, TSC-RTF selects subsequences containing important data points as candidates for intuitive temporal features. Then, TSC-RTF uses random sampling to reduce the number of candidates significantly. Next, TSC-RTF selects the final temporal features using a random forest to ensure the validity of the final temporal features. Finally, a deep learning classifier is trained by TSC-RTF to achieve high accuracy. The experimental results show that the proposed method can compete with the state-of-the-art methods.

1. Introduction

Digital sensors have been developed to collect data in many areas, including healthcare (Sun et al., 2023; Lu et al., 2021), manufacturing (Hsu and Liu, 2021), smart cit (Ji et al., 2020), and intelligent nutrition (Zhang et al., 2023). Usually, these observations are in the form of time series (Ji et al., 2022b). Recently, an increasing amount of research has been devoted to extracting valuable knowledge from time series.

Time series classification (TSC) is one of the research hotspots in the time series mining community (Fawaz et al., 2019; Ruiz et al., 2020; Middlehurst et al., 2023). In general, TSC involves the assignment of class labels to new time series (Esling and Agon, 2012). TSC faces significant challenges due to high noise, high dimensionality, and continuous updating (Ji et al., 2019a).

Over the past few years, many methods have been proposed to solve TSC problems in widely used domains such as EEG/ECG classification, detecting device anomalies, authenticating speakers, etc (Prieto et al., 2015).

* Corresponding author.
E-mail address: jicun@sdnu.edu.cn (C. Ji).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

Improving the classification accuracy is the focus of most of these TSC methods. However, the basis of the classification is also needed for the domain expert. In feature-based methods, representative features are considered as the basis for the classification. For example, doctors classify the ECG time series in Fig. 1 as hypocalcemia or non–hypocalcemia based on whether it has a specific feature (Ji et al., 2022b). By discovering representative features to provide reasons for classification decisions, domain experts can easily accept feature-based TSC methods.

Many feature-based TSC methods have been proposed recently. However, these methods still face the following challenges: 1) **Intuitive features**. Often, the features are not intuitive and are obtained by special computation. For example, some methods have chosen approximate and sample entropy as the features for classifying ECG time series. These features, however, are only directly available to domain experts with computation. On the other hand, intuitive features (such as Q, R, S, and T waves) are more readily accepted by them. 2) **Fast feature selection**. Due to the large number of intuitive feature candidates and the complexity of the evaluation (Ji et al., 2019a), selecting some intuitive features (such as shapelets (Ye and Keogh, 2009)) is time-consuming. Another fundamental challenge for the TSC is to find a fast way to select features. 3) High accuracy. Accuracy is one of the key metrics used to evaluate TSC. The accuracy of traditional classifiers in combination with intuitive features is not ideal (Ji et al., 2022b). It is also a major challenge for TSC to use intuitive features to achieve high accuracy.

https://doi.org/10.1016/j.jksuci.2023.101783
1319-1578/

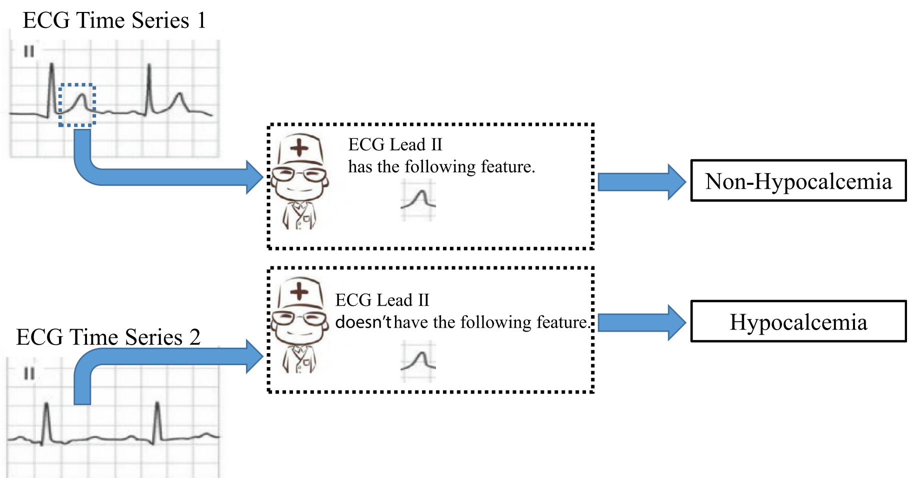


Fig. 1. Judgment process of hypocalcemia. In this process, the doctor classifies the ECG time series as hypocalcemia or non–hypocalcemia based on whether it has a specific feature.

To address the above three challenges, this paper proposes a novel TSC with Random Temporal Features (TSC-RTF) method. First, in order to ensure the intuitiveness of the features, TSC-RTF selects subsequences that contain important data points as candidates for intuitive temporal features. Then, in order to significantly reduce the number of candidates, TSC-RTF uses a random sampling technique. Next, TSC-RTF selects the final temporal features using a random forest to evaluate all candidates simultaneously to ensure the validity of the final temporal features. Finally, the TSC-RTF trains a deep learning classifier in order to achieve a high level of accuracy.

The main contributions of this study can be summarized as follows:

- Firstly, a novel TSC method, TSC-RTF, was proposed to classify time series using intuitive temporal features accurately.
- Secondly, an approach for generating temporal feature candidates has been proposed. In this approach, subsequences that contain important data points have been generated as candidates for the intuitive temporal features. Thus, the intuitiveness of the final temporal features is ensured.
- Thirdly, random sampling is used in TSC-RTF. Using random sampling, TSC-RTF significantly reduces the number of intuitive temporal feature candidates.
- The experimental results on the UCR TSC archive (Dau et al., 2018) have shown that TSC-RTF achieves competitive performance with state-of-the-art methods.

The rest of this paper is structured as follows: A number of related works are described in Section 2. The proposal of the method, TSC-RTF, is presented in Section 3. The experimental results are shown in Section 4. And our conclusions are given in Section 5.

2. Related work

2.1. Categories of TSC methods

TSC exists in many areas (Ji et al., 2022b), such as the classification of EEG and ECG, gesture recognition, the detection of motor faults, and many more. Consequently, an extensive number of TSC methods have been introduced by researchers (Middlehurst et al., 2023). These methods are broadly divided into three categories (Abanda et al., 2019):

- **Distance-based TSC methods**. Distance-based TSC methods classify time series on the basis of the measure of similarity between the instances (Abanda et al., 2019). A one-nearest neighbor classifier with dynamic time-warping distance (1NN-DTW) is usually used as a benchmark for TSC (Bagnall et al., 2017).
- **Model-based TSC methods**. Model-based TSC methods classify time series in accordance with the closest matching model. Two main types of models for TSC are generative models and discriminative models (Baldán and Benítez, 2023; Foumani et al., 2023).
- **Feature-based TSC methods**. Feature-based TSC methods classify time series on the basis of a few representative features (Xiao et al., 2021). There are usually three steps involved in this type of method: 1) feature extraction, 2) transformation of time series into feature vectors, and 3) classification based on the feature vectors.

Among them, the feature-based TSC methods used the representative features for the explanation of the classification results (Lu et al., 2022). In this way, the domain experts' acceptance of the feature-based TSC methods may be more accessible.

2.2. Feature-based TSC methods

Researchers have been trying to propose different types of features for the TSC in the last few years. The features used in TSC include statistical features (Lubba et al., 2019; Baghizadeh et al., 2020; Nanopoulos et al., 2001; Baldán and Benítez, 2023), structural features (Wu et al., 2021; Deng et al., 2013), frequency domain features (Zhang et al., 2005), distance features (Abanda et al., 2019; Kate, 2016), convolutional kernel features (Dempster et al., 2020; Dempster et al., 2021; Dempster et al., 2023), dictionary features (Middlehurst et al., 2019; Lucas et al., 2019; Le Nguyen et al., 2019), and temporal features (Ye and Keogh, 2009; Rakthanmanon and Keogh, 2013).

Of these, temporal features are the most intuitive and easy to interpret (Amouri et al., 2023). The temporal features are discriminating subsequences of the original time series and can be the maximum representation of time series in a class (Ji et al., 2022b).

As part of the original time series, temporal features help domain specialists identify if similar features are present in the time series. In the last few decades, many researchers have been interested in TSC methods based on temporal features.

2.3. Temporal Feature-based TSC methods

A number of temporal feature-based TSC methods have been proposed in recent years. Hao et al. (2023) designed a temporal channel to extract temporal features and then classified time series on the basis of the temporal features and other features. Yang et al. (2023) used multiple attention mechanisms to extract temporal features. Gated linear units were used to extract temporal features by Liu et al. (2023). Du et al. (2023) selected local temporal features by using a time partition and a CBAM block. However, the focus of these methods is mainly on multivariate time series.

For univariate time series, the most commonly used temporal feature is the shapelet (Wei et al., 2023). Various classification strategies have been used to improve the classification accuracy of shapelet-based methods. According to Ye and Keogh (2009), Ye and Keogh (2011), shapelet features are embedded in a decision tree. A random forest is constructed by randomly choosing shapelets to classify time series (Karlsson et al., 2016). Some researchers have constructed a random forest by pairing the shapelets at random (Yuan et al., 2022; Shi et al., 2018). In order to allow different classifiers to be used for classification, the researchers then tried to separate the process of extracting the shapelets from the classification process (Lines et al., 2012). Following this, advanced classifiers are used to improve classifying accuracy. Shapelet features were used to construct an XGBoost classifier by Ji et al. (2019b). Ma et al. (2019) constructed triple shapelet networks by combining triple types of shapelets. An ensemble method is proposed through the combination of the discrete wavelet transform with shapelet features (Yan et al., 2020). In order to improve the accuracy, Ji et al. (2022a), Ji et al. (2022b) combined the fully convolutional network classifier with shapelet features. Although researchers have made some attempts, combining deep learning classifiers with shapelet features is still in its infancy.

Despite the intuitively interpretable nature of shapelet features, selecting shapelets is time-consuming. Researchers have recently attempted to speed up the shapelet selection process by pruning the candidate shapelets (Li et al., 2020; Fang et al., 2018), filtering (Li et al., 2023; Wei et al., 2023), learning (Grabocka et al., 2014; Hou et al., 2016; Wang et al., 2019), reducing measurement complexity (Ji et al., 2022b; Lines and Bagnall, 2012), and random sampling (Renard et al., 2015; Gordon et al., 2015).

This study focuses on speeding up shapelet selection by random sampling. Random shapelets are proposed in combination with the construction of decision trees (Renard et al., 2015). Gordon et al. (2015) introduced random order shapelet sampling to speed up decision tree construction. Karlsson et al. (2016) proposed the gRSF for the construction of a random forest by means of the random selection of shapelets. Random Pairwise Shapelet Forests (PRSF) (Yuan et al., 2022; Shi et al., 2018) are constructed by random pairwise shapelet sampling. The Compressed Random Shapelet Forest (CRSF) was proposed by Yang et al. in order to compress the feature space of the shapelets (Yang et al., 2023). However, there is a high degree of randomness when using random sampling. We should use the random sampling in combination with some other strategies.

3. Our method

In this study, TSC-RTF is proposed for the rapid selection of intuitive features and more accurate classification. There are four main steps in TSC-RTF, as shown in Fig. 2:

- **Candidate generation**. In this step, a few discriminative subsequences are generated as candidates for the temporal features. It is with this step that the intuitiveness of the final temporal features is guaranteed.
- **Random sampling**. In this step, a part of the candidates is selected at random. TSC-RTF significantly reduces the number of intuitive temporal feature candidates by random sampling. By doing this, TSC-RTF can significantly speed up the process of selecting features.
- **Evaluation**. It is in this step that the final temporal features are selected. This is done by evaluating the randomly generated candidates with the help of a random forest. This step ensures the validity of the final temporal features.
- **Training**. This step involves training a deep learning classifier. This step aims to ensure that the TSC-RTF has a high level of accuracy.

3.1. Candidate generation

TSC-RTF generates a number of discriminative subsequences as candidates for temporal features. A discriminative subsequence is defined as a subsequence that contains one or more of the important data points. E.g., P, Q, R, S, and T are common important data points, as shown in Fig. 3. In ECG classification, P waves (which contain P), QRS waves (which contain Q, R, and S) and T waves (which contain T) are commonly used as key features.

In a similar way, intuitive temporal feature candidates are generated with a few important data points. In TSC-RTF, the segmentation points are used as important data points. As shown in Fig. 4, the following steps are used by TSC-RTF to generate intuitive temporal feature candidates:

Step 1 **Identification of important data points**. This step identifies some segmentation points as important data points. The segmentation method proposed by Chung et al. (2004) is used to obtain the segmentation points.

Step 2 **Discriminative subsequence extraction**. The subsequences between two important data points are extracted as candidates for the temporal features. This step ensures that there is at least one important point in each of the candidates. All candidates are thus visually recognizable.

Step 3 **Filtering**. In this step, subsequences that are too long or too short will be discarded. Only the subsequences that meet the length requirements are retained.

3.2. Random sampling

One reason why temporal feature extraction is so time-consuming is the large number of temporal feature candidates (Ji et al., 2022a). To overcome this challenge, TSC-RTF uses a random sampling strategy. TSC-RTF selects a fraction of the temporal feature candidates at random. The number of random candidates is $c * k$, where k is the final number of time features and c is a constant.

The number of candidates is significantly reduced by random sampling. This means that the speed of the selection of the temporal features is increased. Simultaneously, it ensures an optimal number of candidates to evaluate by keeping c constant.

3.3. Evaluation

The fact that traditional methods evaluate candidates individually is another reason for slow temporal feature selection. In order to overcome this challenge, we used the approach that we proposed in our previous paper (Ji et al., 2022b). In this approach, all

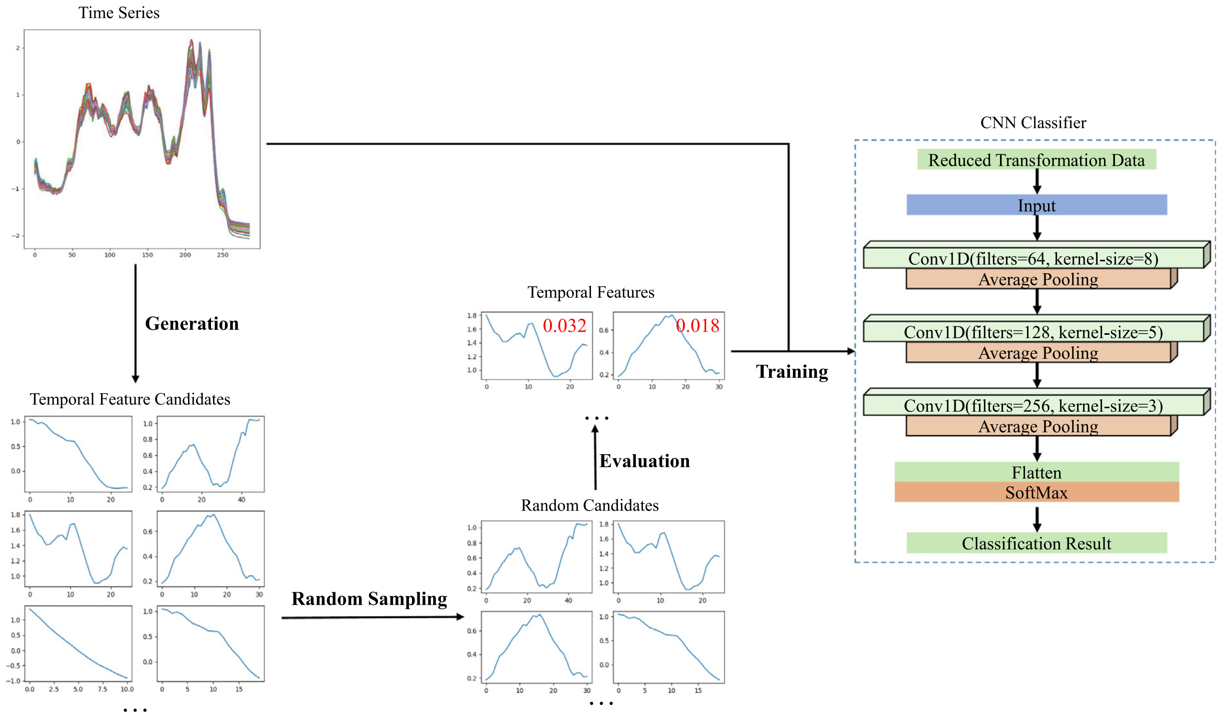


Fig. 2. The training processes of TSC-RTF: 1) **candidate generation**, which generates temporal feature candidates through some discriminative subsequences; 2) **random sampling**, which aims to reduce the number of intuitive temporal feature candidates; 3) **evaluation**, which evaluates the random candidates and selects the final temporal features; and 4) **training**, which trains a deep learning classifier.

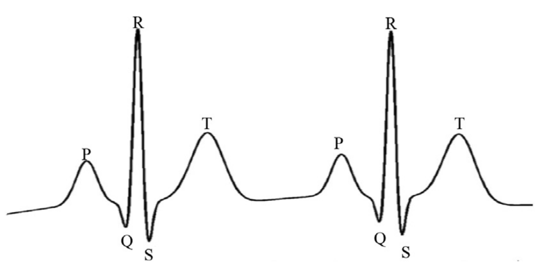


Fig. 3. Important data points of ECG. The important data points are marked as P, Q, R, S, or T.

candidates were evaluated simultaneously using a random forest. There are three steps involved in this approach:

Step 1: Transformation. The time series is transformed from the time domain to the time feature space. The form of the time series T in the temporal feature space is given by Eq. (1), where $tf_1, \cdots, tf_i, \cdots, tf_{c*k}$ are the randomly sampled temporal feature candidates. The value v_{tf_i} is obtained by calculating the Euclidean distance between T and tf_i.

$$T_{tran} = \{v_{tf_1}, \cdots, v_{tf_i}, \cdots, v_{tf_{c*k}}\} \tag{1}$$

Step 2: Candidate Evaluation. In this step, a random forest is used to simultaneously evaluate all candidate temporal features. First, a random forest is trained using the transformed time series. The Gini impurity of each tree node is calculated in the next step. The Gini impurity (Nembrini et al., 2018) is the probability that a random sample of data is incorrectly partitioned. The Gini impurity of a tree node can be calculated as

$$G = \sum_{i=1}^{N_C} p_i(1 - p_i), \tag{2}$$

where N_C is the number of classes and p_i is the frequency of class c_i. In the following, we can get the importance measure of each node. Assume that the tree node *node* is divided into two subnodes, $node_l$ and $node_r$. The importance measure of *node* is calculated as follows

$$IM_{node} = G_{node} - G_{node_l} - G_{node_r}, \tag{3}$$

where G_{node} is the Gini impurity of *node*, G_{node_l} is the Gini impurity of $node_l$, and G_{node_r} is the Gini impurity of $node_r$. Next, the importance measure of the temporal feature candidate IM_{tf_i} is obtained by summing the nodes of the tree using the temporal feature candidate tf_i as the decision conditions in the random forest. Finally, the importance measure is normalized as

$$IM_{tf_i} = \frac{IM_{tf_i}}{\sum_{j=1}^{N_{tf}} IM_{tf_j}}, \tag{4}$$

where N_{tf} is the number of temporal feature candidates.

Step 3: Feature Selection. TSC-RTF selects the final temporal features based on their importance measure with the following three conditions:

- Candidates having a large importance measure are prioritized and selected.
- Either the candidates with the same importance measure are selected at the same time, or they are not selected at all.
- The selection candidates should have an importance measure greater than 0.

3.4. Classifier training

In order to achieve precise TSC accuracy, a CNN classifier is trained in the proposed method. As illustrated in Fig. 5, the classifier takes the reduced transformed data as its input. The transformed time series format (acquired in step 1 of Section 3.3) is

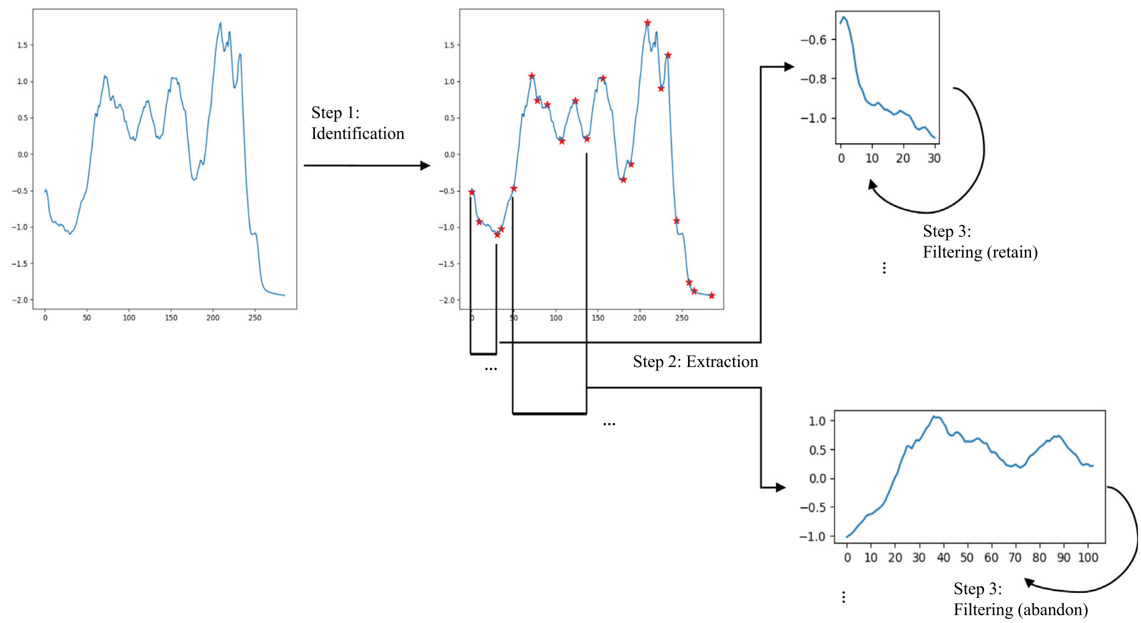


Fig. 4. Steps of candidate generation: 1) **identification**, which identifies the important data points; 2) **extraction**, which extracts subsequences between two important data points as temporal feature candidates; and 3) **filtering**, which discards candidates that do not meet the length requirements.

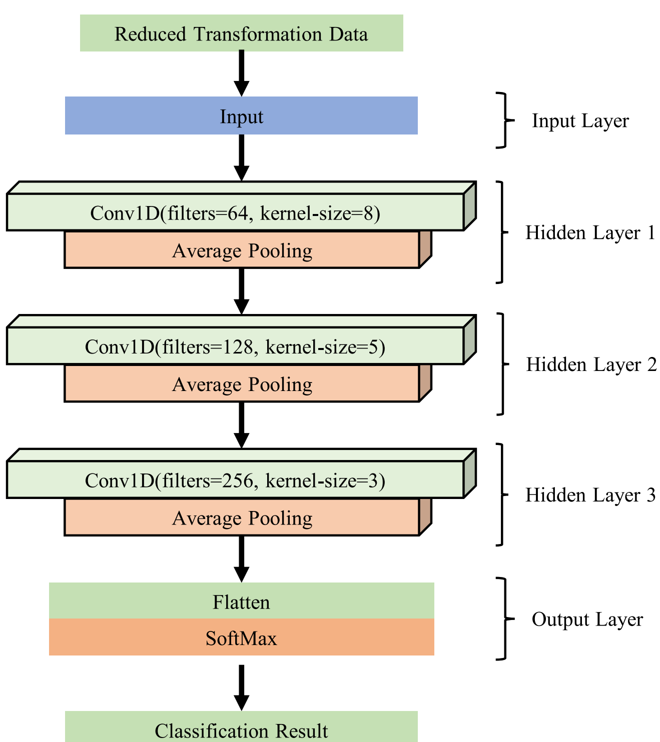


Fig. 5. The architecture of CNN. There are one input layer, three hidden layers, and one output layers in this architecture.

compressed by the final temporal features. Only the dimensions corresponding to the final temporal feature are preserved. The reduced transformed data is then used to train the CNN classifier. The architecture of the CNN classifier is depicted in Fig. 5:

- **Input layer**. The input layer is the place where the reduced transform data is input.
- **Hidden layers**. As illustrated in Fig. 5, the CNN classifier uses three hidden layers to process the input data.
  – **Hidden layer 1**. In this hidden layer, there is a convolutional layer with 64 filters. The size of the filter kernel is set to 8. In the hidden layer 1, the ReLU is used as the activation function. An average pooling is employed within this layer to generate channel-wise statistics.
  – **Hidden layer 2**. In this hidden layer, a convolutional layer with 128 filters is present, with a filter kernel size set at 5. The ReLU activation function and the average pool are likewise incorporated in this hidden layer.
  – **Hidden layer 3**. In this hidden layer, a convolutional layer with 256 filters is present, with a filter kernel size set at 3. The ReLU activation function and average pooling are implemented as well.
- **Output layer**. The CNN classifier employs a SoftMax function and a flattening layer, as illustrated in Fig. 5, to forecast the labels of the time series.

4. Experiments

4.1. Experimental setup

We conducted all experiments using TensorFlow 2.10 on an NVIDIA GeForce GTX 960 M graphics card in Python 3.8. The results presented are averaged over five replicates. Our codes and parameters are publicly available on GitHub for reproducibility[1]. Anyone is capable of reproducing the experiment results autonomously.

4.2. Comparison results with the state-of-the-art methods

We contracted TSC-RF with six state-of-the-art methods:

- **1NN-DTW**, the standard benchmark for TSC (Fawaz et al., 2019; Bagnall et al., 2017).
- **DTW-F** (Kate, 2016) and **catch22** (Lubba et al., 2019), two representative feature-based methods.

[1] Our code: https://github.com/Ji-Cun/TSC_RTF.

- **FS** (Rakthanmanon and Keogh, 2013), the benchmark of temporal feature-based methods.
- **PRSF** (Yuan et al., 2022; Shi et al., 2018) and CRSF (Yang et al., 2023), the representative methods based on random temporal features.

A critical difference diagram is implemented to assess the accuracy of these methods for a clear comparison. As can be seen in Fig. 6, the TSC-RTF method has the smallest average rank among these approaches. Thus, as shown in Fig. 6, the TSC-RTF method outperforms the baseline methods. Please visit our GitHub repository: https://github.com/Ji-Cun/TSC_RTF for more details.

4.3. Comparison results with TSC-TF

We compared our proposed method, TSC-RTF, which uses random sampling to accelerate performance, against TSC-TF (Ji et al., 2022b), our prior method lacking said strategy, on the initial 43 datasets from the UCR TSC archive (Dau et al., 2018). The comparison results are shown in Fig. 7. They show that TSC-RTF outperformed TSC-TF on 22 datasets, lost to TSC-TF on 19, and tied in one dataset. These results suggest that our proposed approach is as accurate as TSC-TF. At the same time, the two methods did not differ in classification accuracy, as shown by the Wilcoxon signed-rank test (Wilcoxon, 1992). Therefore, the use of random sampling as an acceleration strategy does not lead to a reduction in accuracy.

The temporal feature selection time is proportional to the number of candidates to transform (step 1 of Section 3.3) in TSC-RTF and TSC-TF. Fig. 8 shows a comparison of the number of candidates to be transformed. As seen in Fig. 8, using random sampling can significantly reduce the number of candidates. Furthermore, the efficiency of temporal feature selection is improved.

All in all, using a random sampling strategy speeds up feature selection without compromising accuracy.

4.4. Sensitivity analysis

The number of final temporal features k, the random sampling constant c and the alternative classifier are analyzed in this subsection.

4.4.1. Temporal feature number analysis

In these experiments, the number of the final feature is set to 64, 128, 256, 512, 1024, and 2048. The constant c is set to 5 at the same time. The accuracy and feature selection time of 'Adiac', 'FaceAll', and 'Symbols' with different numbers of temporal features are shown in Fig. 9. Fig. 9(a) shows: 1) accuracy increases as the number of temporal features increases, and 2) accuracy is at a high level when the number of temporal features exceeds 512. Fig. 9(b) shows that as the temporal number of features grows exponentially, the time to select features grows exponentially. It should be noted that the coordinate of the Y axis in Fig. 9(b) is exponential. That is, the feature selection time is proportional to the number of temporal features.

Taking into account accuracy and feature selection time, the recommended number of features is 512. We can also increase the number of features for larger datasets and decrease the number of features for smaller datasets.

4.4.2. Random sampling constant analysis

In this group of experiments, the constant c for random sampling is set to 1, 3, 5, 7, and 9. The number of temporal features is set to 512 at the same time. The accuracy and feature selection time of 'Adiac', 'FaceAll', and 'Symbols' with different c are shown in Fig. 10. The following is shown in Fig. 10(a): 1) The accuracy of 'Adiac' is higher when c is set to 3, 5 or 7. 2) The accuracy of 'FaceAll' is basically unchanged. 3) The accuracy of 'TwoLeadECG'

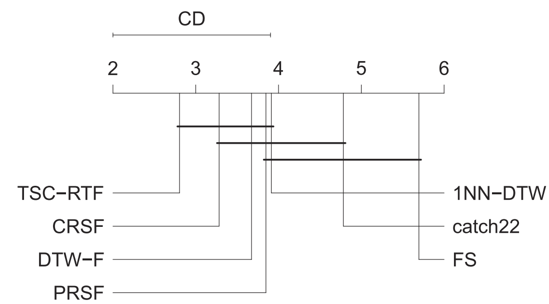


Fig. 6. Critical difference diagram for TSC-RTF and the state-of-the-art methods. It demonstrates that TSC-RTF works better than baselines for TSC-RTF has the smallest average rank among these methods.

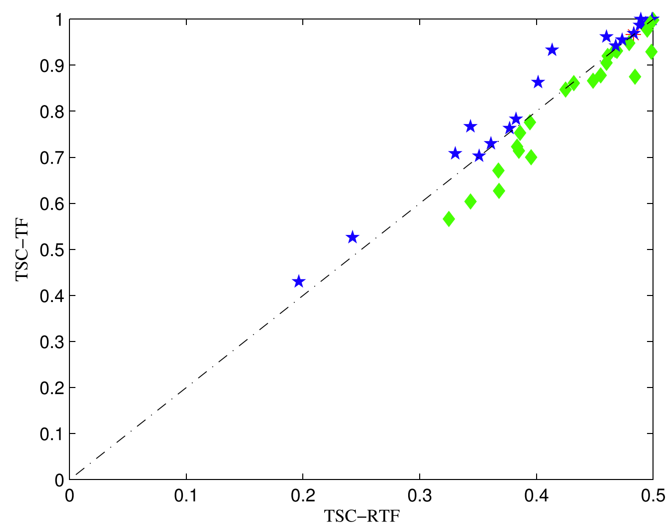


Fig. 7. Accuracy comparison between TSC-TF and TSC-RTF. TSC-RTF won on 22 data sets, lost on 19 data sets and tied on the other data set. These results suggest that our proposed approach is as accurate as TSC-TF.

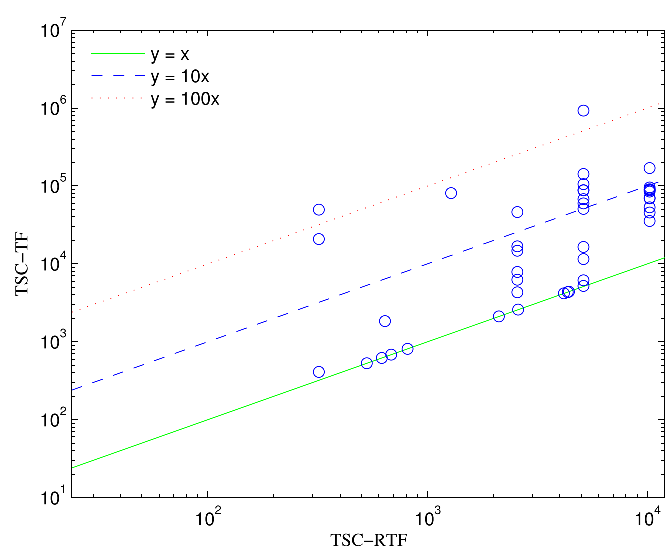
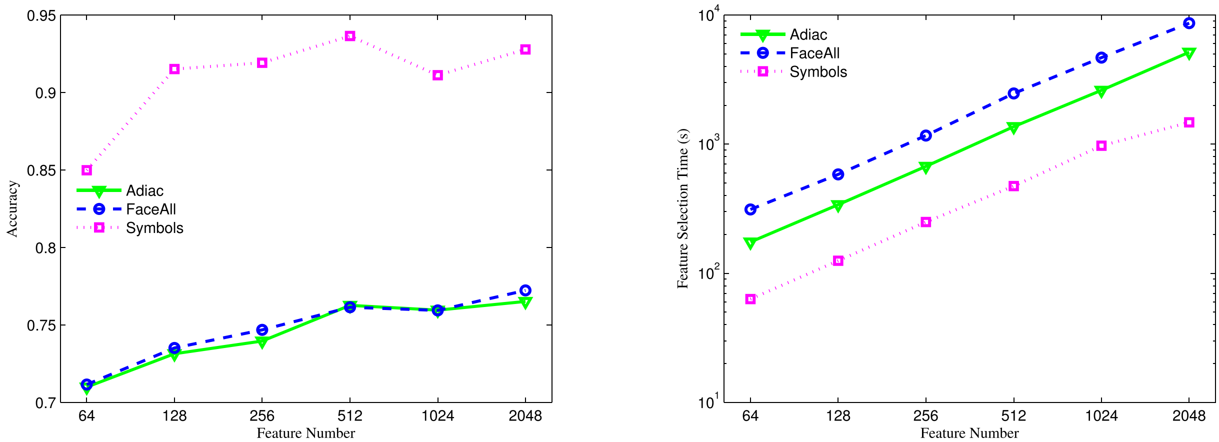
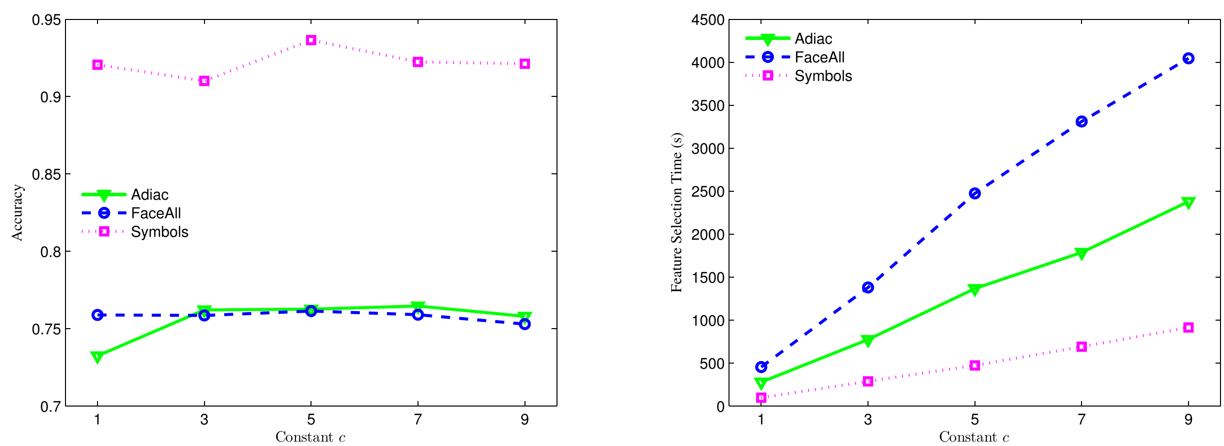


Fig. 8. Candidates number comparison between TSC-TF and TSC-RTF. These results showed that the number of candidates can be significantly reduced by using random sampling.



(a) Accuracy (b) Feature selection time

Fig. 9. Accuracy and feature selection time comparison with different temporal feature numbers.



(a) Accuracy (b) Feature selection time

Fig. 10. Accuracy and feature selection time compassion with different *c*.

is higher when *c* is set to 5. In short, any dataset can achieve high accuracy when *c* is 5. The feature selection time is shown in Fig. 10 (b). From Fig. 10(b), it can be concluded that the feature selection time increases in an approximately linear way as the value of *c* increases.

The recommended constant *c* is 5 for a comprehensive consideration of classification accuracy and feature selection time.

4.4.3. Alternative classifier analysis

As an alternative to the CNN classifier trained in Section 3.4, the following classifiers were used in this group of experiments.

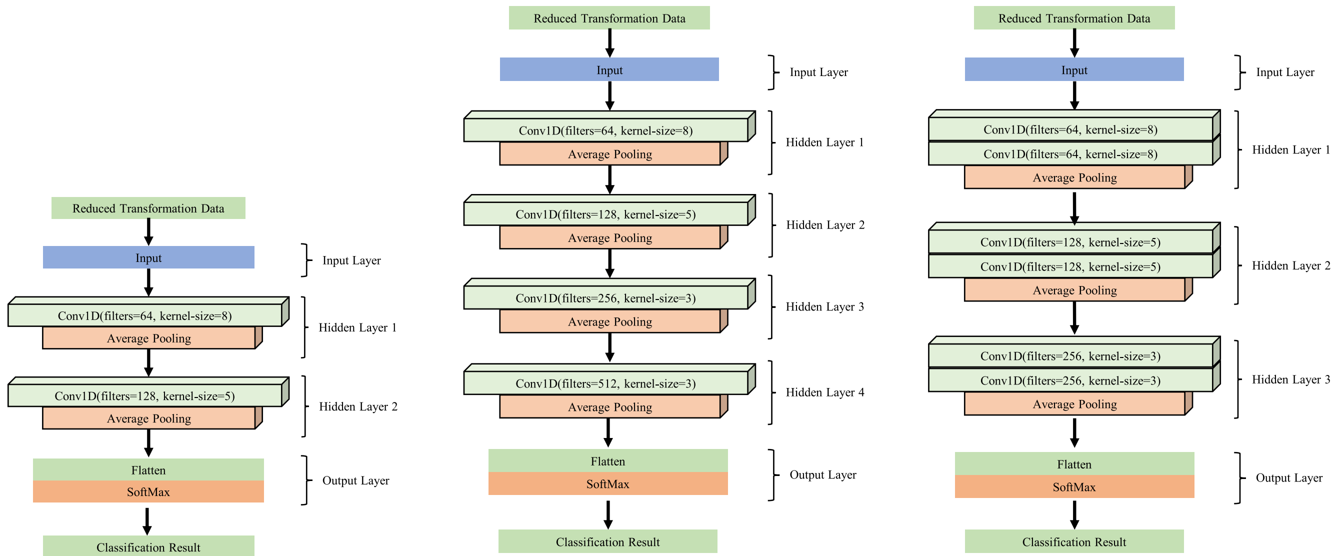
- Less Layer CNN (CNN-LL). The architecture of the CNN-LL is shown in Fig. 11(a). Compared to the CNN in Section 3.4, CNN-LL reduces the hidden layer 3.
- Multi-Layer CNN (CNN-ML). The architecture of the CNN-ML is shown in Fig. 11(b). The CNN-ML adds the hidden layer 4 to the CNN in Section 3.4.
- CNN with more convolutional layers (CNN-MC). The architecture of the CNN-MC is shown in Fig. 11(c). The CNN-MC adds one convolutional layer to each hidden layer compared to the CNN in Section 3.4.

Fig. 12 shows the accuracy and training time of 'Adiac', 'FaceAll', and 'Symbols' with different classifiers. The accuracy of CNN and CNN-LL is higher among them, as shown in Fig. 12(a). This means that an increase in the complexity of the network structure does not lead to an improvement in the accuracy. As shown in Fig. 12 (b), the time required to train CNN is less than that required to train CNN-LL. This means that the training speed is not improved by reducing the complexity of the network structure.

Regarding accuracy and training time, we recommend using a three-layer CNN, as shown in Section 3.4.

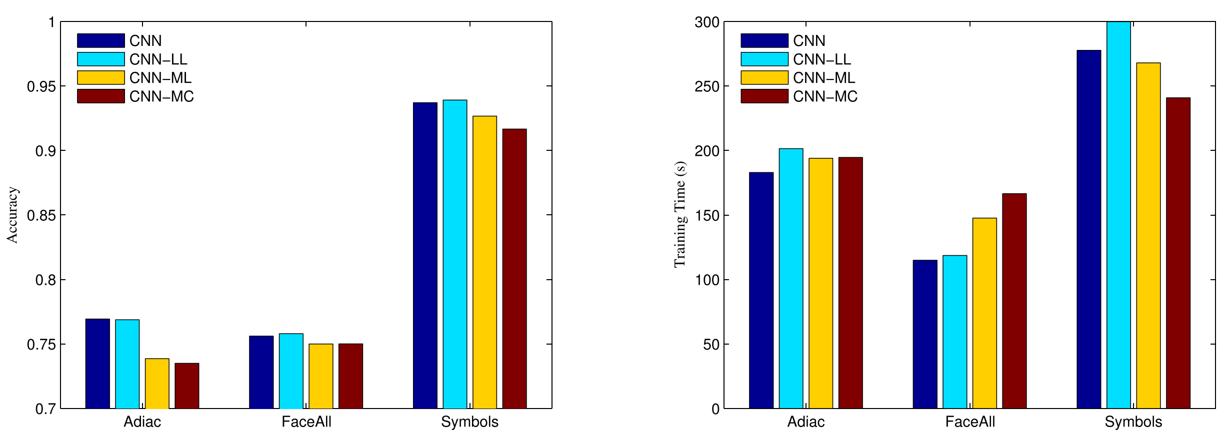
5. Conclusion

Recently, TSC methods based on temporal features have attracted the attention of many researchers because temporal features can be used to explain the classification results. However, the



(a) CNN-LL (b) CNN-ML (c) CNN-MC

Fig. 11. The architectures of CNN-LL, CNN-ML, and CNN-MC.



(a) Accuracy (b) Training time

Fig. 12. Accuracy and training time compassion with different classifiers.

efficient selection of intuitive temporal features for accurate time series classification is still challenging. To this end, this paper proposes TSC-RTF. First, to ensure the intuitiveness of the features, TSC-RTF selects subsequences containing important data points as intuitive temporal feature candidates. Then, TSC-RTF adopts a random sampling method to reduce the number of candidates significantly. Finally, TSC-RTF selects the final temporal features using a random forest to evaluate all candidates simultaneously. This step ensures the validity of the final temporal features. Finally, TSC-RTF trains a deep learning classifier to obtain highly accurate results. Experimental results show that the proposed method is competitive with state-of-the-art methods.

TSC-RTF currently suffers from the following limitations: 1) TSC-RTF is unsuitable for too small datasets due to the need to sample randomly, and 2) the classifier structure used in TSC-RTF is relatively simple. In the future, we will investigate how the sampling rate can be automatically adjusted according to the size of the data set. In addition, we will combine more complex classifiers with random temporal features to improve TSC accuracy in subsequent works.

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

The authors would like to acknowledge the support provided by the Innovation Methods Work Special Project (2020IM020100), and the Natural Science Foundation of Shandong Province (ZR2020QF112).

References

Abanda, A., Mori, U., Lozano, J.A., 2019. A review on distance based time series classification. Data Min. Knowl. Disc. 33 (2), 378–412.

Amouri, H.E., Lampert, T., Gançarski, P., Mallet, C., 2023. Constrained dtw preserving shapelets for explainable time-series clustering. Pattern Recogn. 109804.

Baghizadeh, M., Maghooli, K., Farokhi, F., Dabanloo, N.J., 2020. A new emotion detection algorithm using extracted features of the different time-series generated from st intervals poincaré map. Biomed. Signal Process. Control 59, 101902.

Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E., 2017. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Disc. 31 (3), 606–660.

Baldán, F.J., Benítez, J.M., 2023. Complexity measures and features for times series classification. Expert Syst. Appl. 213, 119227.

Chung, F.-L., Fu, T.-C., Ng, V., Luk, R.W., 2004. An evolutionary approach to pattern-based time series segmentation. IEEE Trans. Evol. Comput. 8 (5), 471–489.

Dau, H.A., Keogh, E., Kamgar, K., Yeh, C.-C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Yanping, Hu, N., Bing nd Begum, Bagnall, A., Mueen, A., Batista, G., 2018. Hexagon-ML, The ucr time series classification archive, https://www.cs.ucr.edu/eamonn/time_series_data_2018/ (October 2018).

Dempster, A., Petitjean, F., Webb, G.I., 2020. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. Data Min. Knowl. Disc. 34 (5), 1454–1495.

Dempster, A., Schmidt, D.F., Webb, G.I., 2021. Minirocket: A very fast (almost) deterministic transform for time series classification. In: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 248–257.

Dempster, A., Schmidt, D.F., Webb, G.I., 2023. Hydra: Competing convolutional kernels for fast and accurate time series classification. Data Min. Knowl. Disc., 1–27

Deng, H., Runger, G., Tuv, E., Vladimir, M., 2013. A time series forest for classification and feature extraction. Inf. Sci. 239, 142–153.

Du, M., Wei, Y., Zheng, X., Ji, C., 2023. Multi-feature based network for multivariate time series classification. Inf. Sci. 639, 119009.

Esling, P., Agon, C., 2012. Time-series data mining. ACM Comput. Surv. 45 (1), 1–34.

Fang, Z., Wang, P., Wang, W., 2018. Efficient learning interpretable shapelets for accurate time series classification. In: 2018 IEEE 34th International Conference on Data Engineering. IEEE, pp. 497–508.

Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L., Muller, P.-A., 2019. Deep learning for time series classification: a review. Data Min. Knowl. Disc. 33 (4), 917–963.

Foumani, N.M., Miller, L., Tan, C.W., Webb, G.I., Forestier, G., Salehi, M., 2023. Deep Learning for Time Series Classification and Extrinsic Regression: A Current Survey. arXiv:2302.02515.

Gordon, D., Hendler, D., Rokach, L., 2015. Fast and space-efficient shapelets-based time-series classification. Intell. Data Anal. 19 (5), 953–981.

Grabocka, J., Schilling, N., Wistuba, M., Schmidt-Thieme, L., 2014. Learning time-series shapelets. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 392–401.

Hao, S., Wang, Z., Alexander, A.D., Yuan, J., Zhang, W., 2023. MICOS: Mixed supervised contrastive learning for multivariate time series classification. Knowl.-Based Syst. 260, 110158.

Hou, L., Kwok, J., Zurada, J., 2016. Efficient learning of timeseries shapelets. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 30.

Hsu, C.-Y., Liu, W.-C., 2021. Multiple time-series convolutional neural network for fault detection and diagnosis and empirical study in semiconductor manufacturing. J. Intell. Manuf. 32, 823–836.

Ji, C., Zhao, C., Liu, S., Yang, C., Pan, L., Wu, L., Meng, X., 2019a. A fast shapelet selection algorithm for time series classification. Comput. Networks 148, 231–240.

Ji, C., Zou, X., Hu, Y., Liu, S., Lyu, L., Zheng, X., 2019b. Xg-sf: An xgboost classifier based on shapelet features for time series classification. Proc. Comput. Sci. 147, 24–28.

Ji, C., Zou, X., Liu, S., Pan, L., 2020. Adarc: An anomaly detection algorithm based on relative outlier distance and biseries correlation. Softw. Practice Exp. 50 (11), 2065–2081.

Ji, C., Hu, Y., Liu, S., Pan, L., Li, B., Zheng, X., 2022a. Fully convolutional networks with shapelet features for time series classification. Inf. Sci. 612, 835–847.

Ji, C., Du, M., Hu, Y., Liu, S.L., Pan, L., Zheng, X., 2022b. Time series classification based on temporal features. Appl. Soft Comput. 128, 109494.

Karlsson, I., Papapetrou, P., Boström, H., 2016. Generalized random shapelet forests. Data Min. Knowl. Disc. 30 (5), 1053–1085.

Kate, R.J., 2016. Using dynamic time warping distances as features for improved time series classification. Data Min. Knowl. Disc. 30 (2), 283–312.

Le Nguyen, T., Gsponer, S., Ilie, I., O'Reilly, M., Ifrim, G., 2019. Interpretable time series classification using linear models and multi-resolution multi-domain symbolic representations. Data Mini. Knowledge Discov. 33 (4), 1183–1222.

Li, G., Choi, B., Xu, J., Bhowmick, S.S., Chun, K.-P., Wong, G.L.-H., 2020. Efficient shapelet discovery for time series classification. IEEE Trans. Knowledge Data Eng. 34 (3), 1149–1163.

Li, C., Wan, Y., Zhang, W., Li, H., 2023. A two-phase filtering of discriminative shapelets learning for time series classification. Appl. Intell. 53 (11), 13815–13833.

Lines, J., Bagnall, A., 2012. Alternative quality measures for time series shapelets. In: International Conference on Intelligent Data Engineering and Automated Learning. Springer, pp. 475–483.

Lines, J., Davis, L.M., Hills, J., Bagnall, A., 2012. A shapelet transform for time series classification. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 289–297.

Liu, C., Zhen, J., Shan, W., 2023. Time series classification based on convolutional network with a Gated Linear Units kernel. Eng. Appl. Artif. Intell. 123, 106296.

Lubba, C.H., Sethi, S.S., Knaute, P., Schultz, S.R., Fulcher, B.D., Jones, N.S., 2019. catch22: Canonical time-series characteristics. Data Min. Knowl. Disc. 33 (6), 1821–1852.

Lucas, B., Shifaz, A., Pelletier, C., O'Neill, L., Zaidi, N., Goethals, B., Petitjean, F., Webb, G.I., 2019. Proximity forest: an effective and scalable distance-based classifier for time series. Data Min. Knowl. Disc. 33 (3), 607–635.

Lu, S., Wang, S.-H., Zhang, Y.-D., 2021. Detection of abnormal brain in mri via improved alexnet and elm optimized by chaotic bat algorithm. Neural Comput. Appl. 33, 10799–10811.

Lu, S., Zhu, Z., Gorriz, J.M., Wang, S.-H., Zhang, Y.-D., 2022. Nagnn: classification of covid-19 based on neighboring aware representation from deep graph neural network. Int. J. Intell. Syst. 37 (2), 1572–1598.

Ma, Q., Zhuang, W., Cottrell, G., 2019. Triple-shapelet networks for time series classification. IEEE, pp. 1246–1251.

Middlehurst, M., Vickers, W., Bagnall, A., 2019. Scalable dictionary classifiers for time series classification. In: International Conference on Intelligent Data Engineering and Automated Learning. Springer, pp. 11–19.

Middlehurst, M., Schäfer, P., Bagnall, A., 2023. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. arXiv:2304.13029.

Nanopoulos, A., Alcock, R., Manolopoulos, Y., 2001. Feature-based classification of time-series data. Int. J. Comput. Res. 10 (3), 49–61.

Nembrini, S., König, I.R., Wright, M.N., 2018. The revival of the gini importance? Bioinformatics 34 (21), 3711–3718.

Prieto, O.J., Alonso-González, C.J., Rodríguez, J.J., 2015. Stacking for multivariate time series classification. Pattern. Anal. Appl. 18 (2), 297–312.

Rakthanmanon, T., Keogh, E., 2013. Fast shapelets: A scalable algorithm for discovering time series shapelets. In: proceedings of the 2013 SIAM International Conference on Data Mining. SIAM, pp. 668–676.

Renard, X., Rifqi, M., Erray, W., Detyniecki, M., 2015. Random-shapelet: an algorithm for fast shapelet discovery. In: 2015 IEEE International Conference on Data Science and Advanced Analytics. IEEE, pp. 1–10.

Ruiz, A.P., Flynn, M., Large, J., Middlehurst, M., Bagnall, A., 2020. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Min. Knowl. Disc., 1–49

Shi, M., Wang, Z., Yuan, J., Liu, H., 2018. Random pairwise shapelets forest. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 68–80.

Sun, L., Zhang, M., Wang, B., Tiwari, P., 2023. Few-shot class-incremental learning for medical time series classification. IEEE J. Biomed. Health Informat., 1–27

Wang, H., Zhang, Q., Wu, J., Pan, S., Chen, Y., 2019. Time series feature learning with labeled and unlabeled data. Pattern Recogn. 89, 55–66.

Wei, Y., Wang, Y., Du, M., Hu, Y., Ji, C. 2023. Adaptive shapelet selection for time series classification. In: 2023 26th International Conference on Computer Supported Cooperative Work in Design. IEEE, pp. 1607–1612.

Wilcoxon, F., 1992. Individual Comparisons by Ranking Methods. Springer.

Wu, S., Wang, X., Liang, M., Wu, D., 2021. Pfc: A novel perceptual features-based framework for time series classification. Entropy 23 (8), 1059.

Xiao, Z., Xu, X., Xing, H., Luo, S., Dai, P., Zhan, D., 2021. Rtfn: A robust temporal feature network for time series classification. Inf. Sci. 571, 65–86.

Yan, L., Liu, Y., Liu, Y., 2020. Application of discrete wavelet transform in shapelet-based classification. Mathe. Probl. Eng.

Yang, C., Wang, X., Yao, L., Long, G., Jiang, J., Xu, G., 2023. Attentional gated Res2Net for multivariate time series classification. Neural Process. Lett. 55 (2), 1371–1395.

Yang, J., Jing, S., Huang, G., 2023. Accurate and fast time series classification based on compressed random Shapelet Forest. Appl. Intell. 53 (5), 5240–5258.

Ye, L., Keogh, E., 2009. Time series shapelets: a new primitive for data mining. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, pp. 947–956.

Ye, L., Keogh, E., 2011. Time series shapelets: a novel technique that allows accurate, interpretable and fast classification. Data Min. Knowledge Disc. 22 (1), 149–182.

Yuan, J., Shi, M., Wang, Z., Liu, H., Li, J., 2022. Random pairwise shapelets forest: an effective classifier for time series. Knowl. Inf. Syst. 64 (1), 143–174.

Zhang, H., Ho, T., Huang, W., 2005. Blind feature extraction for time-series classification using haar wavelet transform. In: International Symposium on Neural Networks. Springer, pp. 605–610.

Zhang, Y., Deng, L., Zhu, H., Wang, W., Ren, Z., Zhou, Q., Lu, S., Sun, S., Zhu, Z., Gorriz, J.M., et al., 2023. Deep Learning in Food Category Recognition. Infor. Fus., 101859