Applied Soft Computing 114 (2022) 108117

Contents lists available at ScienceDirect

Applied Soft Computing

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

TCRAN: Multivariate time series classification using residual channel attention networks with time correction

Hegui Zhu *, Jiapeng Zhang, Hao Cui, Kai Wang, Qingsong Tang

College of Sciences, Northeastern University, Shenyang, 110819, China

ARTICLE INFO

Article history:
Received 3 May 2021
Received in revised form 20 September 2021
Accepted 8 October 2021
Available online 26 November 2021

Keywords:
Adaptive channel feature adjustment mechanism
Inter-module adaptive feature adjustment mechanism
Multivariate time series classification
Time corrected residual attention network
Time residual channel attention block

ABSTRACT

Currently, the most popular and effective approach to solve multivariate time series classification(MTSC) tasks is based on deep learning technology. However, the existing deep learning-based algorithms ignore the unique time characteristics of time series in the process of network training, and do not consider the features correlation in different convolutional layers. So they cannot obtain the convincing feature representation ability and result in unsatisfactory classification accuracy. To solve this problem, we propose a new time corrected residual attention network(TCRAN) which can fully extract the long-term time-dependence information to enhance the discriminative power of the network. The hallmark of TCRAN is that we employ the time residual channel attention block(TRCAB) as the basic structure, which incorporates the adaptive channel feature adjustment mechanism(AFM) and the bi-directional gated recurrent unit(Bi-GRU) into the deep residual structure to adaptively extract time-dependent features. Meanwhile, to integrate the overall dependency information between different layers, we also employ an inter-module adaptive feature adjustment mechanism(IAM) in the TCRAN. The experiments results with 15 multivariate time series datasets illustrate that the proposed TCRAN can achieve the highest average classification accuracy of 0.7276 and improve accuracy by 1.64% compared to the state-of-the-art method. All these verify the effectiveness of TCRAN.

1. Introduction

Time series data is widely available in all areas of human life. With the development of sensors and data storage technologies in recent years, time series data mining has become an increasingly important research field. Time series data mining mainly includes two major categories: time series prediction and time series classification. Time series prediction can play a guiding role for things that have not happened, like future energy prediction [1], and time series classification assists people to judge events that have happened, such as human activity recognition [2]. Among them, multivariate time series classification (MTSC) is one of the main tasks in the field of time series data mining, which contains more information about system operation compared to univariate time series. It has better performance in determining whether an event will occur or not, such as a patient's arrhythmia alarm [3] and whether a machine has malfunctioned [4], etc. MTSC task aims to classify the multivariate time series samples to be measured into pre-defined categories, and has been used in numerous fields such as anomaly detection in industry [4], human motion recognition in sports [5], estimation of sea state in meteorology [6], etc.

In recent years, many researchers have provided different approaches in MTSC task. Traditional MTSC methods are mainly relied on distance similarity or features. For example, Rakthanmanon et al. [7] calculated the DTW distance similarity between series and then used the KNN classifier to finish the classification task, Ye et al. [8] transformed the sequence into many subsequences for feature selection, but the computation and storage cost was too large. In addition, the huge feature space makes feature selection operation difficult, and further affects the classification performance.

Because of the successful applications of deep CNN in various fields, such as automatic ship classification [9], human action recognition [10], and speech emotion recognition [11], so more researchers are applied it to MTSC task. For example, Zheng et al. [12] firstly proposed MCDCNN by using multi-channel convolutional neural networks to learn features individually from different dimensions separately. After that, researchers have proposed many deep learning methods for MTSC with good performance, such as Liu et al. [13] proposed MVCNN with a tensor scheme considers the interaction between a group of time series variables, Hang et al. [14] proposed TapNet with attention prototype network. The improvement of classification effectiveness of deep learning-based methods is mainly relying on deep architectures, which can extract deeper features than traditional MTSC method. Moreove, deep CNN has powerful learning ability

* Corresponding author.
E-mail address: zhuhegui@mail.neu.edu.cn (H. Zhu).

https://doi.org/10.1016/j.asoc.2021.108117
1568-4946/

to learn the complex mapping between the original multivariate time series and the corresponding classes.

As we know, time-related information is one of the most important features in time series data, it has been demonstrated by Karim et al. [15]. But we realize that in the existing methods, the time information has lost gradually during the deep feature extraction process. Moreover, deep learning-based MTSC methods ignored the time features correlation of the intermediate layers and different dimensions in network training progress. Therefore, the feature representation ability and classification accuracy are poor. In addition, the existing methods treat channel features equally and do not consider the correlation between feature channels, which also limits the feature representation ability. Though there are some methods such as MLSTM-FCN [15] which use LSTM [16] to enhance the representation of features in terms of time information, however, these methods still have difficulties in preserving time information in the feature extraction process.

In general, the existing approaches have some drawbacks: (1) the existing network structure does not extract deeper information; (2) time-related information is not fully explored during network training; (3) CNN performs the same processing on different channels, which hinders the representational power of CNN; (4) the current models do not consider the correlation of features between different layers.

To solve these problems, in this paper, we propose a novel time corrected residual attention-based deep learning network (TCRAN), which employs the time residual channel attention block (TRCAB) as the basic structure of the network and incorporates an inter-module adaptive feature adjustment mechanism (IAM). TRCAB fuses the adaptive channel feature adjustment mechanism (AFM) and the Bi-GRU [17] sub-module into the basic residual block. TRCAB is essentially a residual structure, which can build very deep networks to extract deeper features. Also, since Bi-GRU can extract rich time-correlated information, by embedding the Bi-GRU sub-module in TRCAB, it enables the network training process to make full use of the temporal correlation of time series for temporal correction of deep features. Therefore TRCAB not only allows building deeper networks, but also preserves time-related features. Meanwhile, the IAM takes advantage of the interdependencies between different TRCABs to consider the global information of the network. The main contributions are shown as follows:

(1) We propose a novel multivariate time series classification network called time corrected residual attention network (TCRAN), which can enhance the feature representation ability and improve the classification accuracy for MTSC problem.

(2) We design a new time residual channel attention block (TRCAB), which can learn deeper features, retain time correlation information in the process of deep feature extraction, and adaptively adjust feature information by using correlations between different channels.

(3) We introduce an inter-module adaptive feature adjustment mechanism (IAM) to learn the hierarchical features by considering correlations of different TRCABs for obtaining the overall information of the network.

(4) The extensive experiments on 15 datasets from the latest UEA multivariate time series classification archive have shown that the proposed TCRAN can achieve the best performance compared to the current state-of-the-art methods.

The rest content of this paper is structured as follows, Section 2 introduces the relevant works. Section 3 describes the network structure of the proposed model. Section 4 evaluates the performance of the proposed model with some comparable models by experiments and simulation. Finally, Section 5 concludes the main results.

2. Related work

Usually, the algorithms used to solve MTSC task can be classified into distance-based algorithm, feature-based algorithm and deep learning based-algorithm.

2.1. Distance-based algorithm

Distance-based algorithms for MTSC tasks measure the similarity between series by defining a distance metric, and then use classifiers to classify. Various distance metrics such as Euclidean distance [18], short time series distance [19] and dynamic time warping (DTW) distance [20] have been used for multivariate time series similarity comparison. Among them, DTW is regard as the optimal distance metric in MTSC tasks, and the 1NN classifier is the most suitable classifier for pairing with DTW [21,22]. However, since we should compute and save the similarity between every two series before classification, which requires expensive computational and storage costs. Although, researchers have proposed improvements for the above problem, such as DTW-LB [23], this problem still has not been fundamentally solved.

2.2. Feature-based algorithm

Feature-based algorithms extract features from time series data and use traditional classifiers for classification. For example, Weng et al. [24] proposed the two-dimensional singular value decomposition (2dSVD) method, which calculated the covariance matrix of multivariate time series data and used its eigenvectors as features. Baydogan et al. [25] provided SMTS which employed a tree-based symbol generation method to measure the interrelationships among all attributes of multivariate time series. Ye et al. [8] argued that it was a part of subseries of a time series rather than the whole sequence played a decisive role in MTSC, and then they proposed time series shapelets, which recursively searched for the most representative subsequence of each class. However, the extraction of subsequences is too slow and therefore cannot be applied to multidimensional time series classification tasks. To solve this problem, Wistuba et al. [26] provided the Ultra-fast Shapelets (UFS) method, which could find subsequences quickly for MTSC. Furthermore, in order to reduce the computational cost, Karlsson et al. [27] employed the tree-based integration method named Generalized Random Shapelet Forests (GRSF) for MTSC. Recently, Schafer et al. [28] advised the WEASEL-MUSE algorithm, which could improve the classification accuracy and outperform all current Shapelet-based algorithms. Although these methods have improved the classification accuracy, the huge feature space still brings many problems, such as difficulty in selecting features and large computational burden, etc.

2.3. Deep learning-based algorithm

Deep learning-based algorithms automatically learn feature representations and do not require huge feature engineering. It can effectively solve the problems of distance-based and feature-based algorithms. Zheng et al. [12] employed multichannel deep convolutional neural network (MCDCNN), which was a pioneering work applied CNNs to multidimensional time series classification tasks. Due to the successful application of MCDCNN in MTSC task, researchers have proposed many effective network architectures for the MTSC problem in recent years [29]. Tanisaro et al. [30] suggested the Time Warping Invariant Echo State Networks (TWIESN) to improve classification accuracy. Qian

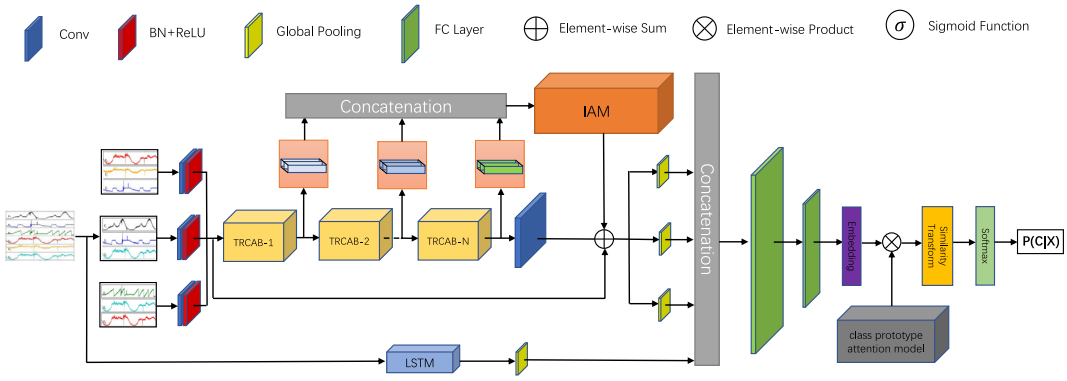


Fig. 1. Network architecture of the proposed TCRAN.

et al. [31] advised DMS-CNN to dynamically extract multidimensional features. Karim et al. [15] designed the MLSTM-FCN, which combined the full convolutional network and LSTM module. Hang et al. [14] proposed a new model TapNet with attention prototype network, which can solve the insufficient sample label problem. More recently, Li et al. [32] suggested ShapeNet to improve the existing shaplets method, which can solve the problem that MTSC cannot compare different variables from different lengths.

In the MTSC task, the time–dependent between series are more informative for distinguishing the different categories. However, few algorithms consider this information of mid–layer features, and do not exploit the global information between different layers, which leads to poor feature characterization ability. If our network focuses more on this type of content, there should be hope for improvement. To investigate such mechanism in deep CNN, we design the new network structure called TCRAN, which will be elaborated in the next section.

3. Proposed model

The framework of the proposed TCRAN is shown in Fig. 1. It consists of four sequential processes: deep feature extraction with time correction by TRCAB, global feature extraction by IAM, feature fusion and the final class prototype classification network. The algorithmic process of TCRAN is illustrated in Algorithm 1. In the following, we will introduce the implementation details of each part in detail.

3.1. Deep feature extraction with time correction

Since random dimension permutation (RDP) is an effective method in extracting the interaction features in multiple dimensions [33,34]. For a multivariate time series $X \in R^{d\times l}$, where d and l are the dimension and length of X respectively. Firstly, we use the RDP method [14] to randomly divide the time series data of different dimensions into different g groups by

$$G_i = \{\varphi^i(1), \varphi^i(2), \ldots, \varphi^i(\tilde{d})\}, i = 1, 2, \ldots, g. \quad (1)$$

where $\tilde{d}$ is the dimension of each group, φ means the random permutation.

Then, we use one-dimensional convolution operation to each group $G_i \in R^{\tilde{d}\times l}$ and obtain shallow features $F_0^{(i)}$ of the group G_i.

Below, we will further perform deep feature extraction with TRCAB module with time correction on $F_0^{(i)}$. As shown in Fig. 1, the deep feature extraction operation is consist of N TRCABs, a layer convolution operation and a long-skip connection (LSC). So,

Algorithm 1 The pseudo-code of TCRAN

Input: Multivariate time series X
Output: The probability P of each category
1: Set the parameters for the TCRAN
2: Initialize the weights of TCRAN
3: For the input X, the RDP method is used to generate g groups G_i randomly, $i = 1, 2, \ldots, g$
4: Apply LSTM and global average pooling operations on X for extracting the original time features F_t
5: **for** each group G_i **do**
6: Obtain the shallow features $F_0^{(i)}$ by applying one-dimensional convolution operation
7: Acquire intermediate layer features $F_j^{(i)}$ by N TRCAB modules $j = 1, 2, \ldots, N$, respectively
8: Concatenate the intermediate layer features $F_j^{(i)}$, and apply IAM to extract globally relevant information $F_{global}^{(i)}$
9: For the feature $F_N^{(i)}$ obtained from the Nth TRCAB, a layer of CNN is used to refine the feature to obtain the deep feature $F_{deep}^{(i)}$
10: After summing the shallow features $F_0^{(i)}$, deep features $F_{deep}^{(i)}$, and global features $F_{global}^{(i)}$, the features $F_p^{(i)}$ are obtained by global average pooling
11: **end for**
12: Concatenate F_t with $F_p^{(i)}$
13: Use two fully connected layers to obtain the low-dimensional embedding F_{emb}
14: The class prototype attention module is used to learn the prototype of each class
15: Calculate the squared Euclidean distance D between the time series embedding F_{emb} and each class prototype
16: Obtain probability P by $P = Softmax(D)$
17: **return** P

we can get the time-corrected deep feature $F_{deep}^{(i)}$ of the group G_i by

$$F_{deep}^{(i)} = F_0^{(i)} + Conv(H_N^{(i)}(H_{N-1}^{(i)}(\cdots H_1^{(i)}(F_0^{(i)})))), \quad i = 1, \ldots, N, \quad (2)$$

where $H_i(\cdot)$ and $Conv(\cdot)$ denote the TRCAB operation and the convolution operation, respectively.

TRCAB is a residual structure integrated by AFM and Bi-GRU sub-modules (the detail content of TRCAB and AFM can be found in Sections 3.1.1 and 3.1.2), which not only learns the deep representation of features from shallow features, but also automatically corrects features using time information and channel correlation.

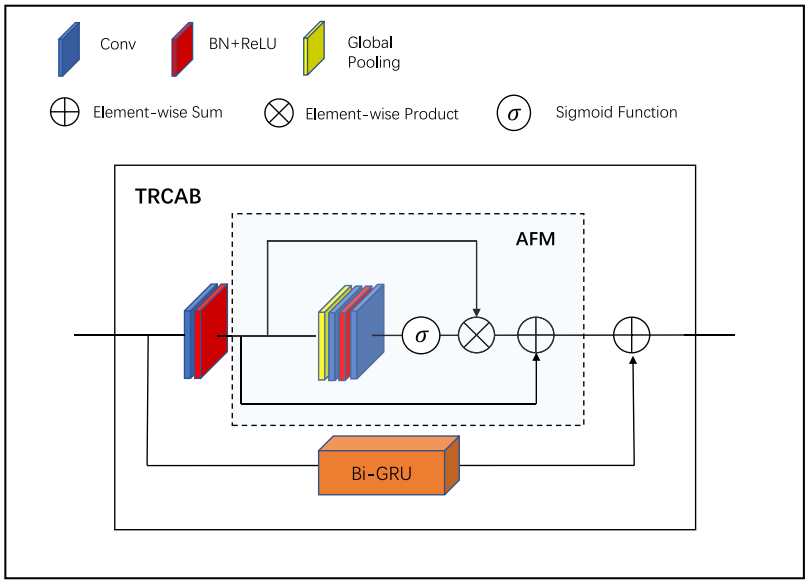


Fig. 2. Architecture of the proposed TRCAB.

3.1.1. Time residuals channel attention block (TRCAB)

As we know, stacking residual blocks can be used to construct deep convolutional neural networks to improve network performance. However, in MTSC task, stacking residual blocks only can deep the network hierarchy and do not improve the classification performance. In addition, the time-dependent information is not fully considered in the process of deep feature extraction. To overcome this drawback, we employ TRCAB as the basic structure of TCRAN, which can form very deep networks and perform time correction on deep features to resolve the long-term dependence of series. The details of TRCAB is shown in Fig. 2.

In detail, TRCAB is classified into three procedures. Firstly, we do a one-dimensional convolution operation on the input, and then feed it into AFM to extract the correlation information between different channels that adaptively adjust the deep features. Furthermore, in another branch, we use Bi-GRU as a sub-module to extract the time-dependent information of the input of TRCAB. Finally, we integrate the outputs of AFM and Bi-GRU to get the time-corrected depth features. Therefore, for the first TRCAB of the group G_i, we have

$$F_1^i = H_{bi-gru}(F_0^i) + H_{afm}(H_{conv1}(F_0^i)), \tag{3}$$

where $H_{bi-gru}(\cdot)$ and $H_{afm}(\cdot)$ are Bi-GRU operation and AFM respectively, $H_{conv1}(\cdot)$ denotes the one-dimensional convolution operation followed by batch normalization and ReLU.

3.1.2. Adaptive channel feature adjustment mechanism (AFM)

Influenced by the successful application of the channel attention mechanism in computer vision [35,36], we make some adjustments of channel attention mechanism for MTSC, and we name it as adaptive channel feature adjustment mechanism (AFM). The structure of AFM is shown in Fig. 3, and the detailed content of AFM is shown in the following.

For the kth TRCAB module, suppose the input of AFM is $\hat{X} = [x_1, x_2, \cdots, x_c]$, $\hat{X} \in R^{l\times c}$, and l is the length of the time series, c is the number of channels of the first convolutional layer in TRCAB. The cth channel of the feature map $\hat{X}$ is denoted by $x_c \in R^{l\times 1}$.

Firstly, we do one-dimensional adaptive averaging pooling on $\hat{X}$ and obtain the channel description $S_{avg} \in R^{c\times 1}$, which can consider the global spatial information between different channels. Then, we calculate the attention weight coefficients w between different channels with a gating mechanism by

$$w = \sigma(W_2 \cdot ReLU(W_1(S_{avg}))), \tag{4}$$

where σ is the Sigmoid function, and W_1, W_2 are the weights of the convolutional layers.

In addition, we also use short skip connections to form the residual structure, which allows features to contain more information. This process can be shown by

$$F_{ca} = H_{ca}(\hat{X}) = w \cdot \hat{X} + \hat{X}, \tag{5}$$

where $H_{ca}(\cdot)$ is the AFM operation.

3.2. Global feature extraction by IAM

Although TRCABs can utilize the time-dependent information during deep feature extraction, these operations do not take advantage of the interdependencies between different TRCABs. Therefore, the proposed network loses some of its global information. For this purpose, we designed an inter-module adaptive feature adjustment mechanism (IAM) to learn the relationship between the feature maps extracted by different TRCABs, where the feature maps from each TRCABs are recognized as a response to a specific categories. Note that the feature maps of different TRCABs are different from each other.

The structure of the designed IAM is shown in Fig. 4. Take the ith group G_i as an example, first, we concatenate the output $F_j^{(i)}$ of N TRCABs to obtain the input F_{IAM}^i of IAM with the shape $N \times L \times C$, where $j = 1, 2, \ldots, N$, N is the number of TRCABs, L is the length of the time series, and C is the number of output channels of TRCAB. Then, we reshape the F_{IAM}^i into a $2D$ matrix M, $M \in R^{N\times(L\times C)}$, and the correlation $W \in R^{N\times N}$ between the different layers can be given by

$$W_{m,n} = \sigma(M_m \times M_n^{\top}), m, n = 1, 2, \ldots, N, \tag{6}$$

where σ is the sigmoid function, $M^{\top}$ represents the transpose matrix of M and $W_{m,n}$ denotes the correlation index between the mth and nth TRCABs.

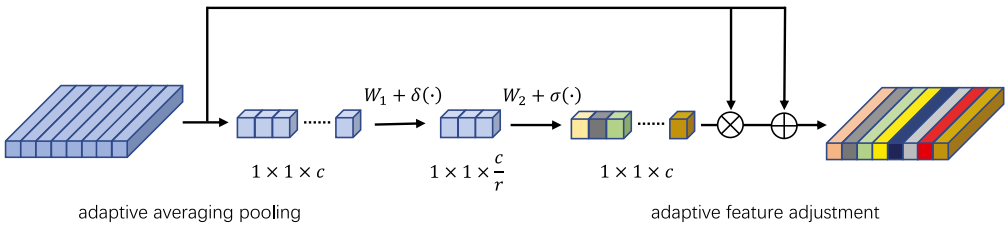


Fig. 3. Detail of the adaptive feature adjustment mechanism (AFM).

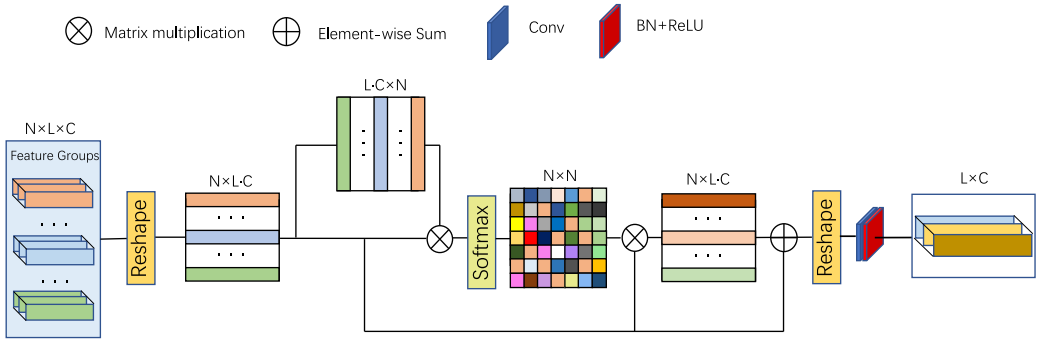


Fig. 4. Architecture of the proposed IAM.

After that we multiply M with the relevant matrix W with a scale factor θ and then we add M, thus we can obtain

$$FM = \theta \sum_{m,n=1}^{N} W_{m,n} M_{n,m} + M, \tag{7}$$

where θ is randomly initialized and gradually updated as the network is trained.

Finally, we reshape the matrix FM with global correlation information and obtain the global features $F_{global}^{(i)}$ by

$$F_{global}^{(i)} = H_{conv}(Re(FM)), \tag{8}$$

where Re is the reshape operation, $H_{conv}(\cdot)$ denotes the one-dimensional convolution operation followed by batch normalization and ReLU.

With IAM, the network can assign different attention weights to different classes of responses, which can automatically improve the representation capability of features.

3.3. Feature fusion

In this section, we will introduce the feature fusion module of the proposed network, where we fuse features to serve as the input of the classification module.

For each group, deep feature extraction with time correction and the global feature extraction are used to obtain deep features F_{deep} and global feature F_{global}. Firstly, we integrate F_{deep} and F_{global} by

$$F_p^{(i)} = H_{pool}(F_{deep}^i + F_{global}^i), i = 1, 2, \ldots, g, \tag{9}$$

where H_{pool} represents global average pooling operation.

Secondly, for the input X of the network, we use LSTM and pooling operation to extract the original time feature F_t with

$$F_t = H_{pool}(H_{lstm}(X)), \tag{10}$$

where H_{lstm} denotes the LSTM operation.

Finally, we concatenate the features $F_p^{(i)}$ and F_t to get the fuse feature $F_{con} \in R^{((f \times g)+h) \times 1}$, f is the number of filters in the last convolution layer, and it can be represented by

$$F_{con} = Con(F_p^{(1)} + F_p^{(2)} + \cdots + F_p^{(g)} + F_t), \tag{11}$$

where $Con(\cdot)$ is the concatenate operation.

3.4. Classification module

For the fusion feature F_{con}, we use TapNet [14] to do the classify task. The classification module is consist of three steps, firstly, we get the low-dimensional embedding features, then the class prototype [37] attention module is used to learn the prototype of each class, finally the input time series is classified according to its squared Euclidean distance from the prototype of each class. According to TapNet, we use two fully connected layers to obtain the low-dimensional embedding F_{emb} by

$$F_{emb} = F_{cn2}(F_{cn1}(F_{con})), \tag{12}$$

where $F_{cn1}(\cdot)$ and $F_{cn2}(\cdot)$ represents two fully connected layers. The prototype of category k is defined by

$$c_k = \sum_i A_{k_i} \cdot H_{k_i}, i = 1, 2, \ldots, |s_k|, \tag{13}$$

where $H_k \in R^{|s_k| \times e}$ means the low-dimensional embedding matrix belonging to category k, $|s_k|$ is the sample size of category k, and A_{k_i} is the weight of the ith time series data sample of category k.

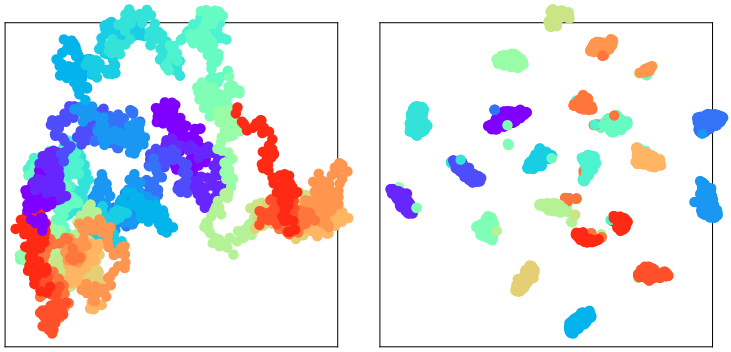
Finally, for the time series data $\bar{x}$ to be classified, the category probability can be obtained by

$$P_\Theta(y = k|\bar{x}) = \frac{\exp(-D(f_\Theta(\bar{x}), c_k))}{\sum_i \exp(-D(f_\Theta(\bar{x}), c_i))} \tag{14}$$

where $D(\cdot)$ is the squared Euclidean distance. Note that the model uses Adam optimizer [38] to minimize the negative log probability $J(\Theta) = -\log P_\Theta(y = k|\bar{x})$ of the true class, which is consistent with TapNet.

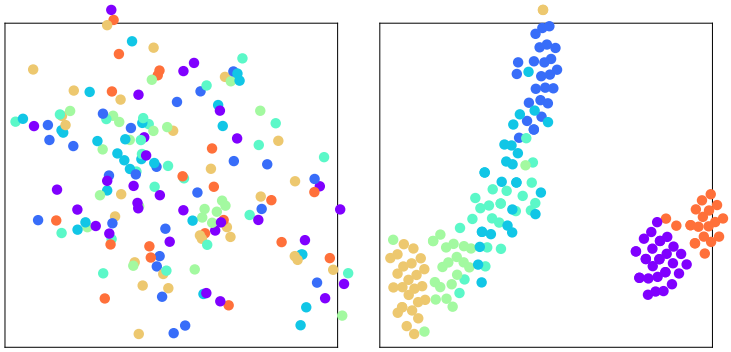
3.5. Visualization of the extracted feature

In order to illustrate the effect of feature extraction by TCRAN, we visualize the feature vectors extracted by TCRAN. Here, we



(a) (b)

Fig. 5. Feature visualization effect of CharacterTrajectories dataset. (a) a layer of CNN; (b) TCRAN.



(a) (b)

Fig. 6. Feature visualization effect of PEMS-SF dataset. (a) a layer of CNN; (b) TCRAN.

choose three representative datasets including CharacterTrajectories, PEMS-SF and PenDigits for visualization. These three datasets represent three typical multidimensional time series data with rich data categories, high data dimensionality and low dimensionality, respectively. Moreover, to illustrate the effectiveness of TCRAN, we choose the features extracted by one layer CNN and TCRAN separately for comparison. Meanwhile we employ T-SNE [39] to reduce the visualization dimension, and the comparable results of the three datasets are shown in Figs. 5–7, where different color represents different categories. From Figs. 5–7, we can see that the initial features extracted by one layer CNN cannot distinguish each classes, however, the features extracted by TCRAN can clearly show the classification boundary of each categories. In addition, for high-dimensional, low-dimensional or category-rich time series data, TCRAN is consistently able to extract features with significant differentiation and obtain more accurate classification results.

4. Experimental results

In this section, we will verify the performance and effectiveness the proposed TCRAN with several advanced algorithms and a series of ablation experiments.

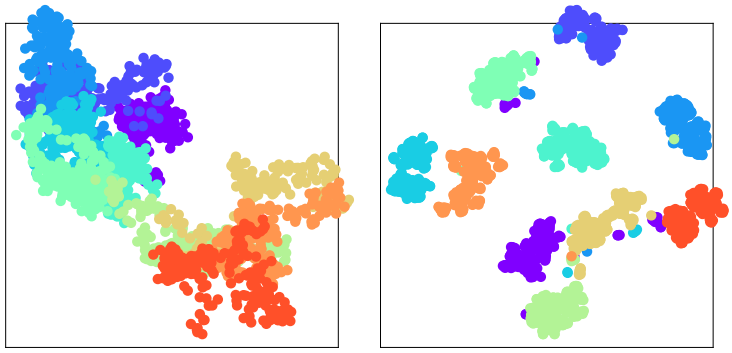
4.1. Settings

Datasets. In order to evaluate the performance of the proposed TCRAN, we choose 15 datasets from the latest UEA multivariate time series classification archive [40]. The details of the datasets are shown in Table 1. The UEA Multivariate Time Series Classification Archive consists of multivariate time series data from different fields (human activity recognition [5], motion classification [41], ECG/EEG classification [3], etc.). The dimension range of the datasets is from 2 (AtrialFibrilation, PenDigits) to 963 (PEMS-SF), and the length range is from 8 (PEMS-SF) to 3000 (MOTORIMAGERY), the category range is from 2 (Heartbeat, MotorImagery, selfregulationSCP2) to 39 (Phoneme), and the datasets size range is from 27 (StandwalkJump) to 10992 (PenDigits).

Evaluation Metrics. In this paper, we use the classification accuracy, average accuracy and the number of Wins/Ties as the evaluation metric to evaluate the performance of TCRAN and some other MSTC algorithms.

Implementation Details. We set the number of TRCAB $N = 3$, the scaling and magnification ratio $r = 16$ in AFM, and the hidden dimension size of Bi-GRU and LSTM is set to 128. And we use the same settings for each TRCAB with a dimension of 256, a convolution kernel size of 1×5 and the same padding.

In TCRAN, we use Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e{-}8$ for training. Due to the large differences in dimension and length of each dataset, we assign different learning rates for different datasets. The setting of learning rate is shown in Table 2. The model is implemented by using PyTorch, and all experiments are performed on the server equipped with the NVIDIA Tesla K80 GPU with CUDA 10.1 and CUDNN-MAJOR 7. The training epochs are set 3000.



(a) (b)

Fig. 7. Feature visualization effect of PenDigits dataset. (a) a layer of CNN; (b) TCRAN.

Table 1
Basic information of 15 MTSC datasets.

Dataset	Num of dimensions	Series length	Num of classes	Train size	Test size	Type
ArticularyWordRecognition	9	144	25	275	300	MOTION
AtrialFibrilation	2	640	3	15	15	ECG
BasicMotions	6	100	4	40	40	HAR
CharacterTrajectories	3	182	20	1422	1436	MOTION
FaceDetection	144	62	2	5890	3524	EEG
HandMovementDirection	10	400	4	160	74	EEG
Heartbeat	61	405	2	204	205	AUDIO
MotorImagery	64	3000	2	278	100	EEG
NATOPS	24	51	6	180	180	HAR
PEMS-SF	963	144	7	267	173	MISC
PenDigits	2	8	10	7494	3498	MOTION
Phoneme	11	217	39	3315	3353	SOUND
SelfRegulationSCP2	7	1152	2	200	180	EEG
SpokenArabicDigits	13	93	10	6599	2199	SPEECH
StandWalkJump	4	2500	3	12	15	ECG

Table 2
Specific setting of learning rate.

Dataset	lr
ArticularyWordRecognition	1×10^{-4}
AtrialFibrilation	1×10^{-4}
BasicMotions	1×10^{-4}
CharacterTrajectories	1×10^{-4}
FaceDetection	1×10^{-3}
HandMovementDirection	1×10^{-3}
Heartbeat	1×10^{-4}
MotorImagery	1×10^{-4}
NATOPS	1×10^{-4}
PEMS-SF	1×10^{-4}
PenDigits	1×10^{-4}
Phoneme	1×10^{-4}
SelfRegulationSCP2	1×10^{-6}
SpokenArabicDigits	1×10^{-4}
StandWalkJump	1×10^{-4}

4.2. Classification performance analysis

In this subsection, we compare the proposed TCRAN with 10 state-of-the-art or most popular methods, which are described as follows: (1) **ShapeNet** [32]: The latest algorithm for solving MTSC task; (2) **TapNet** [14]: The latest solution that can solve the semi-supervised MTSC problem; (3) **MLSTM-FCN** [15]: A state-of-the-art generic framework for MTSC task; (4) **WEASEL+MUSE** [28]: The latest BOP based method for MTSC; (5) **ED-1NN**: One of the most popular baselines for MTSC, 1*NN* refers to classification using nearest neighbor classifier; (6) **ED-1NN(norm)**: **ED-1NN** with data normalization, *norm* refers to the data is normalized; (7) **DTW-1NN-I**: It is the most commonly used baseline model based on DTW, *DTW* represent dynamic time warping and *I* means treating each dimension separately; (8) **DTW-1NN-I(norm)**; (9) **DTW-1NN-D**: *D* means treating dimension together; (10) **DTW-1NN-D(norm)**.

We compare the performance of each method from three aspects: classification accuracy, average accuracy and the number of Wins/Ties in different datasets. The results are given in Table 3. It can be seen from Table 3 that TCRAN achieves the best classification accuracy on 13 datasets. In terms of average accuracy, TCRAN achieves the highest average classification accuracy of 0.7276. In the number of Wins/Ties, TCRAN has 13 Wins/Ties, ShapeNet has 6 Wins/Ties, TapNet has 2 Wins/Ties. All these validate the effectiveness and robustness of TCRAN.

4.3. Ablation study

To evaluate the effectiveness of the components of TCRAN, we employ the ablation experiments on 15 datasets. The experiments are conducted in 4 parts, firstly, we add AFM only to the baseline, then, we add the residual structure with AFM, after that we use the TRCAB structure, and finally we use both TRCAB and IAM. The effectiveness of each of the proposed modules can be verified separately through the above experiments. The experimental results are shown in Table 4.

From Table 4, compared with baseline, AFM can get better classification performance than baseline on 7 datasets. However,

Table 3
Comparison of classification accuracy in UEA multivariate time series datasets. The best and second best results are highlighted in bold and underlined.

Dataset	TCRAN	ShapeNet	TapNet	MLSTM -FCN	WEASEL +MUSE	ED-1NN	DTW-1NN-I	DTW-1NN-D	ED-1NN (norm)	DTW-1NN -I(norm)	DTW-1NN -D(norm)
ArticularyWordRecognition	**0.997**	0.987	0.987	0.973	0.99	0.97	0.98	0.987	0.97	0.98	0.987
AtrialFibrilation	**0.4**	**0.4**	0.333	0.267	0.333	0.267	0.267	0.2	0.267	0.267	0.22
BasicMotions	**1**	**1**	**1**	0.95	**1**	0.675	**1**	0.975	0.676	**1**	0.975
CharacterTrajectories	**1**	0.98	0.997	0.985	0.99	0.964	0.969	0.99	0.964	0.969	0.989
FaceDetection	0.581	**0.602**	0.556	0.545	0.545	0.519	0.513	0.529	0.519	0.5	0.529
HandMovementDirection	**0.419**	0.338	0.378	0.365	0.365	0.279	0.306	0.231	0.278	0.306	0.231
Heartbeat	**0.785**	0.756	0.751	0.663	0.727	0.62	0.659	0.717	0.619	0.658	0.717
MotorImagery	**0.61**	**0.61**	0.59	0.51	0.5	0.51	0.39	0.5	0.51	N/A	0.5
NATOPS	**0.972**	0.883	0.939	0.889	0.87	0.86	0.85	0.883	0.85	0.85	0.883
PEMS-SF	**0.803**	0.751	0.751	0.699	N/A	0.705	0.734	0.711	0.705	0.734	0.711
PenDigits	**0.98**	0.977	**0.98**	0.978	0.948	0.973	0.939	0.977	0.973	0.939	0.977
Phoneme	0.199	**0.298**	0.175	0.11	0.19	0.104	0.151	0.151	0.104	0.151	0.151
SelfRegulationSCP2	**0.578**	**0.578**	0.55	0.472	0.46	0.483	0.533	0.539	0.483	0.533	0.539
SpokenArabicDigits	**0.99**	0.975	0.983	**0.99**	0.982	0.967	0.96	0.963	0.967	0.959	0.963
StandWalkJump	**0.6**	0.533	0.4	0.067	0.333	0.2	0.333	0.2	0.2	0.333	0.2
Avg.Value	**0.7276**	0.7112	0.6913	0.6308	0.6155	0.6064	0.6389	0.6369	0.6057	0.6119	0.6381
Wins/Ties	13	6	2	1	1	0	1	0	0	1	0

Table 4
Ablation study about method we proposed for MTSC.

Datasets	Baseline	+ AFM	+ Res + AFM	+ TRCAB	+ TRCAB + IAM
ArticularyWordRecognition	0.987	0.99	0.993	0.997	0.997
AtrialFibrilation	0.333	0.333	0.333	0.4	0.4
BasicMotions	1	1	1	1	1
CharacterTrajectories	0.997	0.998	0.998	1	1
FaceDetection	0.556	0.569	0.578	0.578	0.581
HandMovementDirection	0.378	0.392	0.405	0.446	0.419
Heartbeat	0.751	0.726	0.726	0.766	0.785
MotorImagery	0.59	0.57	0.57	0.61	0.61
NATOPS	0.939	0.961	0.972	0.972	0.972
PEMS-SF	0.751	0.7283	0.803	0.803	0.803
PenDigits	0.98	0.933	0.9728	0.98	0.98
Phoneme	0.175	0.175	0.196	0.196	0.199
SelfRegulationSCP2	0.55	0.561	0.572	0.572	0.578
SpokenArabicDigits	0.983	0.9845	0.988	0.988	0.99
StandWalkJump	0.4	0.3333	0.6	0.6	0.6
Avg	0.6913	0.6836	0.7138	0.7272	0.7276
Win/Ties	2	1	5	10	14

the average accuracy with AFM is 0.77% lower than that of baseline, which indicates that although AFM can enhance the feature representation ability, but it does not improve the discriminative ability of the network well on some longer sequences. That is because long time sequences have a strong time dependence, which cannot be captured by just adding AFM, and the current methods cannot focus on extracting deep feature information during network training.

Subsequently, we integrate a residual structure with AFM, and the results in Table 4 illustrate that this method can achieve better classification accuracy over baseline on 10 datasets. Compared with the results of AFM, *Res + AFM* has better performance on 10 datasets, and has the same accuracy on the rest 5 datasets. In terms of average accuracy, *Res + AFM* has improved 2.25% and 3.02% over baseline and AFAM, respectively.

Furthermore, we integrate the Bi-GRU module with *Res + AFAM* to form the proposed TRCAB, we conduct experiments on 15 datasets in the same settings. It can be clearly seen from Table 4 that the proposed TRCAB can have the better classification accuracy, and the average classification accuracy is improved by 3.59%, 4.36%, and 1.34% than baseline, baseline with AFM and baseline with *Res + AFM* respectively.

Finally, we add the proposed IAM to the network, as we can see in Table 4, the average accuracy of adding IAM is 0.7276, which is higher than all the previous results.

In a word, all these experimental results show that the all the proposed components of TCRAN is indeed effective for the MTSC task.

4.4. Analysis on the component of TRCAB

In this section, we evaluate the effectiveness of the component of TRCAB, which is an important part in implementing time correction. Here, we mainly explore the influence of LSTM, Bi-LSTM, GRU and Bi-GRU in TRCAB by classification accuracy, and the experimental results are shown in Fig. 8:

From Fig. 8, we can clearly see that the Bi-GRU has better classification accuracy than other three method. In terms of training time, GRU has the shortest training time and Bi-LSTM has the longest training time. Compared with GRU and Bi-GRU, a better performance can be obtained by using Bi-GRU with only a small increase in computational burden. Therefore, after a comprehensive consideration, we finally choose Bi-GRU as the basic component of TRCAB to extract the long-term dependence information of the series for time correction.

4.5. Analysis on the number of TRCAB

For the number of TRCAB, we conduct affection of different numbers of TRCAB to TCRAN. Specifically, we applied 1, 2, 3, 5 and 10 TRCABs respectively and show the performance on NATOPS dataset. The experimental results are shown in Fig. 9. From Fig. 9, we can see that TCRAN with 3 TRCABs has the best accuracy. So we choose 3 TRCABs in TCRAN. In addition, even if we used 1 TRCAB, TCRAN still has better classification results than baseline, which also illustrates the effectiveness of TRCAB.

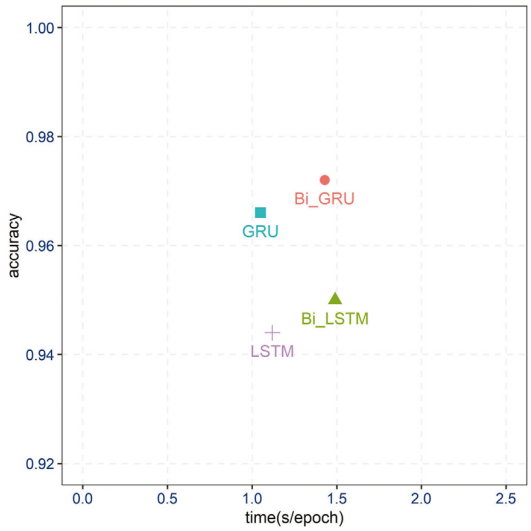


Fig. 8. Component analysis of TRCAB on NATOPS.

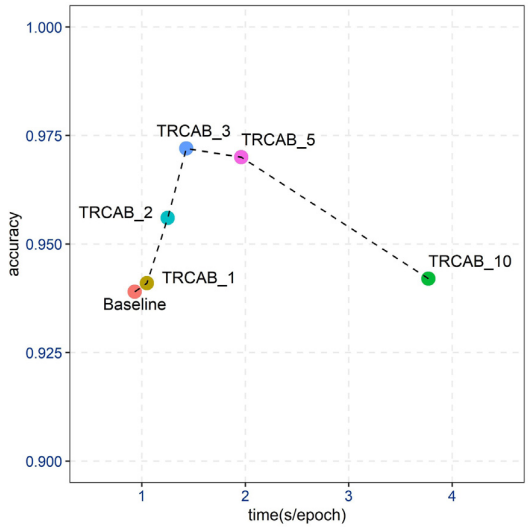


Fig. 9. Research about using different numbers of TRCABs.

5. Conclusion and discussion

In this paper, a novel deep learning algorithm named time corrected residual attention network (TCRAN) was proposed for MTSC task by introducing deep time correction and global information. To address the problem that the existing networks cannot adequately extract the time-dependent features of the time series, we employed TRCAB as the base module of TCRAN, which could build depth networks and used time-dependent information to perform time-correction of deep features. Meanwhile, IAM was introduced to exploit the global feature information by considering the correlation of features extracted by different TRCAB. Extensive experiments on 15 datasets from UEA's latest multivariate time series classification archive illustrated that TCRAN achieved the best average classification accuracy of 0.7276. Morover, TCRAN outperformed 10 of the latest methods in terms of performance, where their best result was 0.7112.

Although TCRAN has shown strong performance and scalability in offline scenarios, we have not evaluated the performance of TCRAN in real-time online scenarios, so, the evaluation of TCRAN in online scenarios needs further research. Meanwhile, in future, it is a challenge to make TCRAN into a lightweight network without affecting its performance. In addition, the unsupervised methods such as contrast learning also can be applied to TCRAN to address datasets with few labels.

CRediT authorship contribution statement

Hegui Zhu: Conceptualization, Methodology, Resources, Algorithm design, Writing – review and editing. **Jiapeng Zhang:** Formal analysis, Writing – original draft, Experimental data validation. **Hao Cui:** Programming, Software. **Kai Wang:** Visualization, Data collection and curation. **Qingsong Tang:** Investigation, Literature research, Supervision.

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 study was funded by the Natural Science Foundation of Liaoning Province, China (NO. 2020-MS-080), the Fundamental Research Funds for the Central Universities (NO. N2005032), the National Key R&D Program of China (NO. 2017YFF0108800).

References

[1] M. Ishaq, S. Kwon, et al., Short-term energy forecasting framework using an ensemble deep learning approach, IEEE Access 9 (2021) 94262–94271.
[2] J. Yang, M.N. Nguyen, P.P. San, X. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time for human activity recognition, in: IJCAI, vol. 15, Buenos Aires, Argentina, 2015, pp. 3995–4001.
[3] X. Wang, Y. Gao, J. Lin, H. Rangwala, R. Mittu, A machine learning approach to false alarm detection for critical arrhythmia alarms, in: 2015 IEEE 14th International Conference on Machine Learning and Applications, ICMLA, IEEE, 2015, pp. 202–207.
[4] A.M. Tripathi, R.D. Baruah, Anomaly detection in multivariate time series using fuzzy adaboost and dynamic Naive Bayesian classifier, in: 2019 IEEE International Conference on Systems, Man and Cybernetics, SMC, IEEE, 2019, pp. 1938–1944.
[5] D. Minnen, T. Starner, I. Essa, C. Isbell, Discovering characteristic actions from on-body sensor data, in: 2006 10th IEEE International Symposium on Wearable Computers, IEEE, 2006, pp. 11–18.
[6] X. Cheng, G. Li, A.L. Ellefsen, S. Chen, H.P. Hildre, H. Zhang, A novel densely connected convolutional neural network for sea-state estimation using ship motion data, IEEE Trans. Instrum. Meas. 69 (9) (2020) 5984–5993.
[7] T. Rakthanmanon, E.J. Keogh, Data mining a trillion time series subsequences under dynamic time warping, in: IJCAI, 2013, pp. 3047–3051.
[8] L. Ye, E. Keogh, Time shapelets: A new primitive for data mining, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2009, pp. 947–956.
[9] D. Połap, M. Włodarczyk-Sielicka, N. Wawrzyniak, Automatic ship classification for a riverside monitoring system using a cascade of artificial intelligence techniques including penalties and rewards, ISA Trans. (2021).
[10] K. Muhammad, A. Ullah, A.S. Imran, M. Sajjad, M.S. Kiran, G. Sannino, V.H.C. de Albuquerque, et al., Human action recognition using attention based LSTM network with dilated CNN features, Future Gener. Comput. Syst. 125 (2021) 820–830.
[11] S. Kwon, Optimal feature selection based speech emotion recognition using two-stream deep convolutional neural network, Int. J. Intell. Syst. (2021).
[12] Y. Zheng, Q. Liu, E. Chen, Y. Ge, J.L. Zhao, Time series classification using multi-channels deep convolutional neural networks, in: International Conference on Web-Age Information Management, Springer, 2014, pp. 298–310.
[13] C.-L. Liu, W.-H. Hsaio, Y.-C. Tu, Time series classification with multivariate convolutional neural network, IEEE Trans. Ind. Electron. 66 (6) (2018) 4788–4797.
[14] X. Zhang, Y. Gao, J. Lin, C.-T. Lu, Tapnet: Multivariate time series classification with attentional prototypical network, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 6845–6852.

[15] F. Karim, S. Majumdar, H. Darabi, S. Harford, Multivariate LSTM-FCNs for time series classification, Neural Netw. 116 (2019) 237–245.
[16] S. Hochreiter, The vanishing gradient problem during learning recurrent neural nets and problem solutions, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 6 (02) (1998) 107–116.
[17] K. Cho, B. Van Merriënboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder-decoder approaches, 2014, arXiv preprint arXiv:1409.1259.
[18] C. Faloutsos, M. Ranganathan, Y. Manolopoulos, Fast subsequence matching in time-series databases, ACM Sigmod Rec. 23 (2) (1994) 419–429.
[19] C.S. Möller-Levet, F. Klawonn, K.-H. Cho, O. Wolkenhauer, Fuzzy clustering of short time-series and unevenly distributed sampling points, in: International Symposium on Intelligent Data Analysis, Springer, 2003, pp. 330–340.
[20] D.J. Berndt, J. Clifford, Using dynamic time warping to find patterns in time: KDD Workshop, vol. 10, Seattle, WA, USA, 1994, pp. 359–370.
[21] A. Sharabiani, H. Darabi, A. Rezaei, S. Harford, H. Johnson, F. Karim, Efficient classification of long time series by 3-D dynamic time warping, IEEE Trans. Syst. Man Cybern. Syst. 47 (10) (2017) 2688–2703.
[22] M. Shokoohi-Yekta, B. Hu, H. Jin, J. Wang, E. Keogh, Generalizing DTW to the multi-dimensional case requires an adaptive approach, Data Min. Knowl. Discov. 31 (1) (2017) 1–31.
[23] S.-W. Kim, S. Park, W.W. Chu, An index-based approach for similarity search supporting time warping in large sequence databases, in: Proceedings 17th International Conference on Data Engineering, IEEE, 2001, pp. 607–614.
[24] X. Weng, J. Shen, Classification of multivariate time series using two-dimensional singular value decomposition, Knowl.-Based Syst. 21 (7) (2008) 535–539.
[25] M.G. Baydogan, G. Runger, Learning a symbolic representation for multivariate time series classification, Data Min. Knowl. Discov. 29 (2) (2015) 400–422.
[26] M. Wistuba, J. Grabocka, L. Schmidt-Thieme, Ultra-fast shapelets for time series classification, 2015, arXiv preprint arXiv:1503.05018.
[27] I. Karlsson, P. Papapetrou, H. Boström, Generalized random shapelet forests, Data Min. Knowl. Discov. 30 (5) (2016) 1053–1085.
[28] P. Schäfer, U. Leser, Multivariate time series classification with WEASEL+MUSE, 2017, arXiv preprint arXiv:1711.11343.
[29] A.P. Ruiz, M. Flynn, J. Large, M. Middlehurst, A. Bagnall, The great multivariate time series classification bake off: A review and experimental evaluation of recent algorithmic advances, Data Min. Knowl. Discov. (2020) 1–49.
[30] P. Tanisaro, G. Heidemann, Time series classification using time warping invariant echo state networks, in: 2016 15th IEEE International Conference on Machine Learning and Applications, ICMLA, IEEE, 2016, pp. 831–836.
[31] B. Qian, Y. Xiao, Z. Zheng, M. Zhou, W. Zhuang, S. Li, Q. Ma, Dynamic multi-scale convolutional neural network for time series classification, IEEE Access 8 (2020) 109732–109746.
[32] G. Li, B. Choi, J. Xu, S.S. Bhowmick, K.-P. Chun, G.L.-H. Wong, ShapeNet: A shapelet-neural network approach for multivariate time series classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, (9) 2021, pp. 8375–8383.
[33] Y. Gao, J. Lin, Efficient discovery of time series motifs with large length range in million scale time series, in: 2017 IEEE International Conference on Data Mining, ICDM, IEEE, 2017, pp. 1213–1222.
[34] Y. Gao, J. Lin, Hime: Discovering variable-length motifs in large-scale time series, Knowl. Inf. Syst. 61 (1) (2019) 513–542.
[35] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 7132–7141.
[36] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, X. Tang, Residual attention network for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3156–3164.
[37] J. Snell, K. Swersky, R.S. Zemel, Prototypical networks for few-shot learning, 2017, arXiv preprint arXiv:1703.05175.
[38] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.
[39] L. Van der Maaten, G. Hinton, Visualizing data using T-SNE, J. Mach. Learn. Res. 9 (11) (2008).
[40] A. Bagnall, H.A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, E. Keogh, The UEA multivariate time series classification archive, 2018, 2018, arXiv preprint arXiv:1811.00075.
[41] 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.