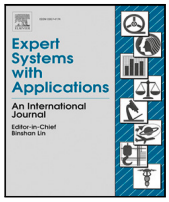
Expert Systems With Applications 248 (2024) 123452

Contents lists available at ScienceDirect

Expert Systems With Applications

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



Multivariate time series classification based on fusion features

Mingsen Du [a], Yanxuan Wei [a], Yupeng Hu [b], Xiangwei Zheng [a,c], Cun Ji [a,c,*]

[a] School of Information Science and Engineering, Shandong Normal University, Jinan, China
[b] School of Software, Shandong University, Jinan, China
[c] Shandong Provincial Key Laboratory for Distributed Computer Software Novel Technology, Jinan, China

ARTICLE INFO

Dataset link: http://timeseriesclassification.com

Keywords:
Multivariate time series classification
Kendall coefficient
Attention
Fusion features
Graph neural network

ABSTRACT

In various areas of real life, Multivariate Time Series Classification (MTSC) is widely used. It has been the focus of attention of many researchers, and a number of MTSC methods have been proposed in recent years. These methods tend to focus on the features in only a single domain. However, they have ignored the correlation and complementarity between the features in a space of multiple domains. In this paper, a novel MTSC method based on fusion features (MTSC_FF) is presented to address this problem. Firstly, MTSC_FF extracts the frequency domain features using an attention layer based on continuous wavelet transform. In parallel, MTSC_FF extracts long-range dependency features from the time domain, using a sparse self-attention layer. Simultaneously, MTSC_FF obtains spatial correlations between multivariate time series dimensions via Kendall coefficient. Then, a graph neural network is used to fuse all features. Finally, the fusion features are used to predict the classification labels by means of the fully connected layer. The experimental results obtained on the UEA datasets show that the proposed method can achieve high accuracy. In addition, the proposed method can visualize the classification-dependent features, which is an improvement in the interpretability of the results.

1. Introduction

Time series is a set of data collected from sensors and organized in chronological order (Prieto, Alonso-González, & Rodríguez, 2015). It comes from various fields, including gene sequence research (Aach & Church, 2001), post-operation recovery detection (Tormene, Giorgino, Quaglini, & Stefanelli, 2009), error message recognition (Yu, Zeng, Xue, & Ma, 2022), sleep record classification (Chambon, Galtier, Arnal, Wainrib, & Gramfort, 2018), intoxication detection (Li, Jin, & Zhao, 2015), behavior recognition (Ma, Li, Zhang, Gao and Lu, 2019), etc.

One of the main tasks in time series analysis is time series classification. Related methods usually predict labels for unknown time series based on knowledge from existing labeled instances. Over the years, an increasing number of researchers have started to focus on time series classification.

Based on the classification objects, the current time series classification methods can be divided into the univariate time series classification methods and the multivariate time series classification (MTSC) methods (Ircio, Lojo, Mori, & Lozano, 2020; Wang et al., 2022). The former classifies the univariate time series, and the latter classifies the multivariate time series. The univariate time series classification methods usually perform the classification by mining local features (Ji et al., 2022) or relationships (Xiao et al., 2021) (i.e., long-range dependency features). In comparison with univariate time series, multivariate time series have richer information about relationships in different dimensions. As a result, MTSC is more challenging.

A number of MTSC methods have been proposed over the years. For example, some methods directly extract the local or long-range dependency features of the original time series in the time domain (Karim, Majumdar, Darabi, & Harford, 2019). Meanwhile, some methods classified time series based on the feature in the frequency domain using Fourier Transform (FT) or Wavelet Transform (WT) (Yang, Yuan, & Wang, 2022). In addition, some methods used the hidden dependencies between the dimensions of the multivariate time series for the classification (Duan et al., 2022).

In spite of the considerable efforts made by researchers, the following challenges still have to be overcome by these methods: **(1) Efficiently extracting long-range dependency features.** The observations in the time series that follow are influenced by the observations that precede them. For example, certain roads are more likely to be influenced by traffic information from neighboring areas, and the current traffic flow affects the following traffic flow. However, due to the disadvantages of large computation and gradient explosion,

* Corresponding author at: School of Information Science and Engineering, Shandong Normal University, Jinan, China.
E-mail addresses: 2021317071@stu.sdnu.edu.cn (M. Du), 2022317121@stu.sdnu.edu.cn (Y. Wei), huyupeng@sdu.edu.cn (Y. Hu), xwzheng@sdnu.edu.cn (X. Zheng), jicun@sdnu.edu.cn (C. Ji).

https://doi.org/10.1016/j.eswa.2024.123452
Received 19 September 2023; Received in revised form 25 January 2024; Accepted 8 February 2024
Available online 9 February 2024

current models such as transformer, recurrent neural networks (RNNs) as well as long short-term memory networks (LSTMs) cannot effectively extract features with long-range dependency features. Among them, Transformer, as a gradually popular model in the field of time series classification, has excellent results for long-range dependency extraction. However, due to the matrix operation in performing self-attention computation is space complexity is huge (Liu, Ren et al., 2021). Because the length of some datasets is particularly long, the computation of feature matrices consumes a huge amount. Therefore, efficient extraction of effective long-range dependencies will be that important. **(2) Fusion of time and frequency domain features.** Most of the current methods have a focus on features in the time domain or the frequency domain. However, these methods ignore the correlation and complementarity between features in a multi-domain space. Numerous methods (Du, Wei, Zheng, & Ji, 2023) use only time-domain methods, and additional frequency-domain features can be used as regularization to further improve feature extraction. This leads to low classification accuracy (Huang, Zhang, Fan, & Xi, 2021). **(3) Mining correlations between dimensions.** Multivariate time series have richer information between different dimensions. The use of graph models to explore hidden dependencies among multivariate time series is promising (Duan et al., 2022) due to the rapid development of graph-based methods, such as graph neural network (GNN). For example, in 3D skeleton action recognition (Ma, Tian, Wei, Wang and Ng, 2019), there will be certain connections or spatial correlations between the corresponding dimensions of different limbs. Numerous multivariate approaches focus on individual dimensions, not realizing that the relationships among dimensions are also critical. Most existing methods have extracted features from each dimension (Karim et al., 2019). Attention-based (Liu, Ren et al., 2021) method can obtain inter-dimensional weights, and an attention layer over all channels captures correlations between channels in all time steps. In contrast, GNN is a model specifically designed for graph-structured data that captures the complex relationships between nodes. When based on multivariate time series graph structures through a quantitative approach, GNN learns node representations by performing information transfer and aggregation over nodes. Transformer still performs well in processing sequence data and natural language processing tasks. The models of GNN and Transformer are therefore combined to fully utilize their respective strengths to process data that contains both graph, sequence and time series structures. And finding means to quantitatively represent the spatial correlations between dimensions becomes crucial.

To address the above challenges, a novel MTSC method based on Fusion Features (MTSC_FF) is proposed in this paper. Firstly, MTSC_FF extracts the frequency domain features through an attention layer with the help of continuous wavelet transform (CWT). In parallel, MTSC_FF uses a sparse self-attention layer to extract long-range dependency features from the time domain. At the same time, MTSC_FF obtains the spatial correlations between the d multivariate time series through the Kendall coefficient. And then, all the features are fused by means of the GNN. Finally, the fusion features are used to predict the classification labels through the fully connected (FC) layer.

The main contributions of this paper are as follows:

1. We propose MTSC_FF, which fused the time domain features, the frequency domain features, and the spatial correlations among the multivariate time series dimensions to obtain high MTSC accuracy.
2. We use the Kendall coefficients to represent the spatial correlation among the dimensions of multivariate time series.
3. Considerable experiments on various types of publicly available datasets demonstrate the effectiveness of our method. And we explains the classification results by visualizing proposed features.

The remainder of this paper is structured as follows. Section 2 introduces the related work. Section 3 describes our method in detail. The experimental results are shown in Section 4. And our conclusions are provided in Section 5.

2. Related work

Over the years, a number of MTSC methods have been proposed. According to feature types, these methods are classified into time domain feature-based, frequency domain-based, dimensions relationship-based, and fusion feature-based methods.

2.1. Time domain feature-based methods

Time domain feature-based methods use features that directly extract local features from time series as the basis for classification. For example, Ye and Keogh (2009, 2011) extracted some representative segments (i.e., shapelet) for classification. Some temporal features are adopted by Ji et al. (2022) with a fully convolutional network (FCN).

Recently, researchers have attempted to improve classification accuracy by utilizing the relationships of local features (long-range dependency features). Hao and Cao (2020) introduced a temporal attention based network to mine long-term and short-term dependencies of time series. Karim et al. (2019) combined squeeze and excitation module with LSTM and FCN model to capture relationships of features. Chen, Yan, Wang, and Xiao (2022) a network based on sparse self-attention and attention to extract the local features and their relationships. Hong, Yan, Chen, et al. (2022) introduced a reset unit to model the long dependency relationships between the local features. Xiao et al. (2021) adopted an LSTM-based attention model to mine the relationships of the features of multivariate time series. Liu, Ren et al. (2021) proposed an extension of the current transformer networks with gating to model the channel-wise and step-wise correlations. Zhang, Hou, OuYang, and Zhou (2022) transformed time series into multiscale recurrence plot to obtain rich time-correlated features from the time domain, and used FCN for classification.

2.2. Frequency domain feature-based methods

Frequency domain-based methods extract features from the frequency domain (Li, Bissyande, Klein, & Le Traon, 2016).

FT is one standard method to convert time series into the frequency domain. After converting the raw data into time–frequency images by short-time Fourier transform (STFT), Shao, Huang, and Zhu (2020) used ResNet-50 to extract the frequency domain features. Yang et al. (2022) combined the coefficients of discrete Fourier transform to obtain the helpful frequency domain features. Li, Chowdhury, Shang, Gupta, and Hong (2021) integrated STFT into deep models by initializing convolutional filter weights as the Fourier coefficients.

WT is another standard method to convert time series into the frequency domain. Chen et al. (2021) used multilevel discrete wavelet decomposition to mine time–frequency domain features. Batal and Hauskrecht (2009) applied DWT on time series data and selected the most distinguishing set of coefficients to denote the origin time series.

2.3. Dimension relationship-based methods

Compared with univariate time series, multivariate time series have richer dependency features among dimensions. Dimension relationship-based methods use hidden correlations among dimensions to obtain the spatio correlation features. Ma, Tian et al. (2019) used an attention-based network to capture discriminative sample-specific spatial features at each time step for MTSC. Duan et al. (2022) combined GNN and an encoder–decoder-based variational graph pooling network, thus creating adaptive centroids for graph coarsening. Yang, Chen, Song, and Gong (2017) approximated the sequential dynamics and explicitly learned the causal correlation relationships among multiple variables. Zha, Lai, Zhou, and Hu (2022) used dynamic time warping (DTW) as a similarity criterion to treat each sample as a graph node. Then they represents time series classification as a node-level classification problem in the graph, where the nodes in the graph correspond

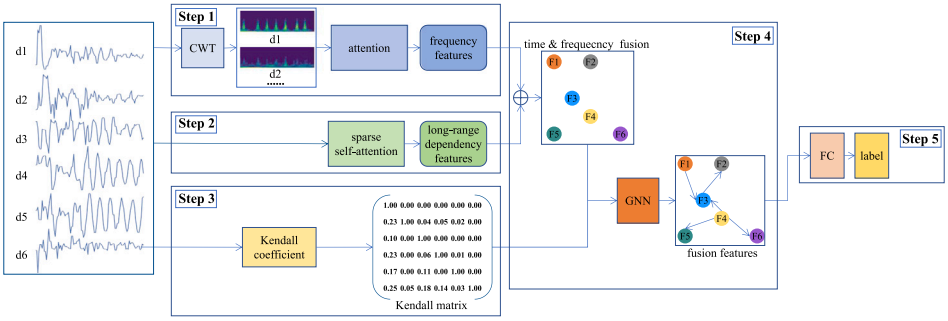


Fig. 1. Overall structure of MTSC_FF, which has five steps.

to each sample and the links correspond to pairs of distance based similarities. Zuo, Zeitouni, and Taher (2021) modeled spatio-temporal dynamic features. Ding, Sun, and Zhao (2023) employed a multimodal based graph attention network as well as a time convolution module to mine the spatial correlation and temporal correlation.

2.4. Fusion features-based methods

Fusion feature-based methods combine features for various domains to improve classification accuracy. El-Sappagh, Abuhmed, Islam, and Kwak (2020) proposed an ensemble network based on CNN and a bidirectional LSTM network, which jointly predicted multiple variables based on the fusion of various multimodal time series. Jiang, Liu, and Lian (2022) proposed a multimodal fusion transformer, which used the Gramian angular field to convert time series to image, then used CNN to mine multimodal features from time series as well as images separately to fuse both. Chambon et al. (2018) used CNN to exploit multivariate and multimodal polysomnography signals, which could exploit the temporal context of each 30 s window of data. Wang, Jiang et al. (2020) used LSTM along with an attention block to learn time features, and a CNN along with an attention block to get fusion features of both time features from time series and graph features from area graphs. Iwana and Uchida (2020) et al. combined DTW as distance features with original time series for multimodal feature fusion, and used CNN as feature classifier.

3. The proposed method

3.1. Overview

As shown in Fig. 1 and Algorithm 1, MTSC_FF mainly includes five steps:

Step 1. **Extracting frequency domain features**. This step extracts the frequency domain features based on the time–frequency image. In this step, MTSC_FF first transforms multivariate time series into time–frequency images through CWT and then utilizes an attention layer to obtain frequency domain features (Refer to Section 3.2).

Step 2. **Extracting long-range dependency features**. This step extracts long-range dependency features from the time domain by means of a sparse self-attention layer (Refer to Section 3.3).

Step 3. **Calculating the spatial correlations**. This step calculates the spatial correlations among dimensions by Kendall coefficients and represents spatial relationships through a Kendall matrix (Refer to Section 3.4).

Step 4. **Feature fusion**. This step fuses fused the time domain features, the frequency domain features, and the spatial correlations among the multivariate time series dimensions with the help of a GNN (Refer to Section 3.5).

Step 5. **Classification**. This step classifies time series based on the fusion features through an FC layer (Refer to Section 3.6).

Algorithm 1 MTSC_FF

Require: training set: D, epochs: $epoch$
Ensure: MTSC_FF classifier: C;
1: Frequency image set $=F$
2: **for** each time series T in D **do**
3: **for** each dimension Dim in T **do**
4: Frequency image $Fi = CWT(Dim)$
5: $F = F \cup Fi$
6: **end for**
7: **end for**
8: Frequency domain features $FD = attention(F)$ ▷ Step 1: Obtaining frequency domain images and extracting frequency domain features
9: Time domain features $TD = sparse\ \ self-attention(D)$ ▷ Step 2: Extracting time features
10: **for** each time series T in D **do**
11: Spatial correlation $SC = Kendall\ \ coefficient(T)$ ▷ Step 3: Calculating the Kendall coefficient based spatial correlations
12: **end for**
13: Time and frequency fusion $TF = FD \oplus TD$
14: Feature fusion $FF = GIN(TF)$ ▷ Step 4: All feature fusion
15: C=training(D,FF,$epoch$) ▷ Step 5: MTSF_FF training
16: **return** C

3.2. Extracting frequency domain features

MTSC_FF extracts frequency domain features through the following steps:

Step 1: Acquisition of time–frequency image based on CWT

MTSC_FF converted every dimension of multivariate time series to a time–frequency image through CWT. The conversion formula of CWT is shown in Eq. (1). In Eq. (1), a is the scale factor, b is the translation factor, t is the index of the time series, $f(t)$ is the observed value at time T, $\Psi^{\frac{t-b}{a}}$ is the wavelet sequence obtained by the action of the mother wavelet $\Psi(t)$ with a and b, $|a|^{-\frac{1}{2}}$ is used as the normalization factor to maintain the relative magnitude of energy on different scales, and $W_f(a,b)$ is the final WT coefficient.

$$W_f(a,b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{\infty} f(t)(\Psi^{\frac{t-b}{a}})\mathrm{d}t \tag{1}$$

Step 2: Frequency domain features extraction

MTSC_FF extracts frequency domain features through an attention layer. This layers extracts differentiated features by multiplying the attention maps by the input feature maps. The attention maps are obtained for the channel and spatial perspectives:

In channel perspective, the attention focuses on the weight of each channel (Woo, Park, Lee, & Kweon, 2018). As shown in Fig. 2, the time–frequency image is embedded to obtain the embedding (E, $C \times H \times W$, where C is the channel dimension, H and W represent the height and width dimensions of the time–frequency image, respectively). Then, through average pooling as well as maximum pooling, the feature map are aggregated to obtain (E', $C \times 1 \times 1$). The channel attention is

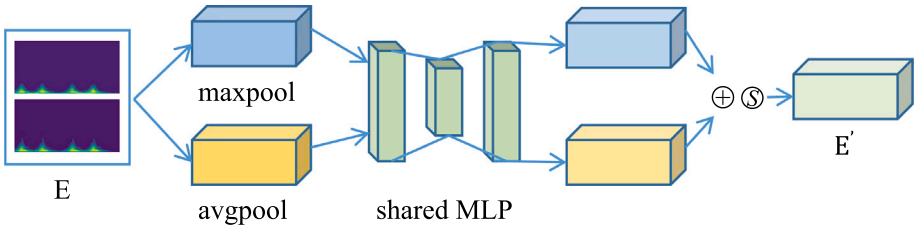


Fig. 2. Channel attention.

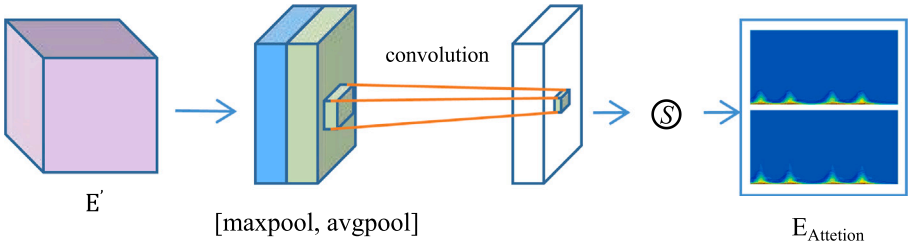


Fig. 3. Spatial attention.

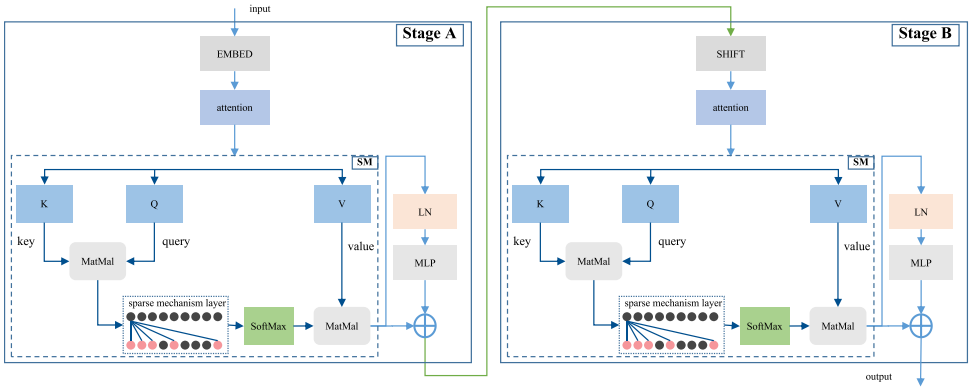


Fig. 4. The overview structure of the sparse self-attention layer.

calculated by Eq. (2). In Eq. (2), MLP is the function of the shared parameters, and $sigmoid$ is the activation function.

$$E' = sigmoid(MLP(AvgPool(E)) + MLP(MaxPool(E))) \tag{2}$$

In spatial perspective, the attention focuses on the weight of each part of the pixel and gets ($E_{Attention}$, $C \times H \times W$). As shown in Fig. 3, the spatial attention is obtained using the average pooling ($1 \times H \times W$) as well as the maximum pooling ($1 \times H \times W$), and then the spatial attention feature map is generated by the 2-dimension convolution. Spatial attention is calculated according to Eq. (3). In Eq. (3), f^a denotes a 2-dimension convolution (filter size is a), (E', $C \times 1 \times 1$) denotes feature from channel attention layer, and $sigmoid$ is the activation function.

$$E_{Attention} = sigmoid(f^a([AvgPool(E'); MaxPool(E')])) \tag{3}$$

3.3. Extracting long-range dependency features

MTSC_FF extracts long-range dependency features from the time domain through a sparse self-attention layer. The structure of the sparse self-attention layer is shown in Fig. 4 and Algorithm 2. As Fig. 4 shows, this layer is divided into two stages: Stage A and Stage B. In Stage A and Stage B, each module is repeated twice except for the EMBED layer and SHIFT layer (Liu, Lin et al., 2021).

The descriptions of each part are as follows:

Algorithm 2 Sparse self-attention

Require: training set: D, number of model layer: N;
Ensure: Time domain features: TF;
for each time series T ($L \times C$) in D **do**
  for each layer in N **do** ▷ Several layers in the model
    $count = 0$
    while $count < 2$ **do** ▷ Stage A and B: Two stages in each layer
      E_1 ($\frac{T}{4} \times 4C$)=$embed(T)$
      E_2 =$attention(E_1)$
      Q, K, V=E_2
      E_3 =$SM(Q, K, V$, sparse matrix $M)$ ▷ Sparse mechanism
      E_4 =$LN(E_3)$
      E_5 =$MLP(E_4)$
      if is $Stage\ \ A$ **then**
        E_6 =$SHIFT(E_5)$ ▷ Shift window mechanism
      end if
      $count++$
    end while
  end for
end for
E=E_5
TF=training(D,N,E) ▷ Training and obtaining time features
return TF

- **EMBED layer**. The EMBED layer is used to obtain the time series embedding of time series. Firstly, the multivariate time series T

($L \times C$, L is the time series length, and C is the dimension of T) is separated by non-overlapping neighboring windows into four time blocks. And then, each time block is flattened to get the time series embedding ($\frac{T}{4} \times 4C$). The above operation restricts the time series computational self-attention to each time block, reducing the time complexity.

- **Attention layer**. The attention layer is used for feature refinement through focusing on more distinguishing features of each window. For more details, refer to Section 3.2.
- **SM layer**. The sparse mechanism (SM) layer introduces a sparse bias M to capture long-range dependency features. The structure of the SM layer is shown in the SM module of Fig. 4. This SM layer adopts a sparse mechanism strategy (Li et al., 2019) to accelerate the self-attention dot product (Beltagy, Peters, & Cohan, 2020; Wang, Li, Khabsa, Fang and Ma, 2020). This strategy limit the dot product of each cell. As shown in Fig. 4, only the bright circles participate in the dot product calculation, while the dark circles do not participate in the operation. The calculation procedure of the SM layer is summarized in Eq. (4). Q is used to query which of K is more important and get the corresponding weight matrix, and then multiply it by V, so that V can focus on the more important information and ignore the less important information. In other words, the process of finding the importance is the process of finding the similarity matching, the greater the similarity means the higher the importance, the more attention to this part. In Eq. (4), Q, K, and V denote the feature matrices obtained by refining features through the previous attention layer, M functions as the bias matrix for sparse computing, and $SoftMax$ is a function to process the initial output results in the classification task.

$$T_{SSA} = Softmax\left(\frac{QK^T}{\sqrt{d_k}} + M\right)V \tag{4}$$

- **LN layer**. A layer normalization (LN) layer. The hidden layers are normalized to a standard normal distribution to speed up training and accelerate convergence.
- **MLP layer**. An multilayer perceptron (MLP) layer. It functions as a fully connected layer.
- **SHIFT layer**. The SHIFT layer adopts a shift window mechanism to allow cross-window connections among small blocks. So, the long-range dependent features are extracted with reducing the computational effort.

3.4. Calculating the spatial correlations

Kendall coefficient (Abdi, 2007) is a nonparametric statistic that measures the correlation between two variables. It measures the order consistency between two variables rather than a linear relationship. In practice, Kendall coefficient can be used in many important research areas. In market research, we can use Kendall coefficient to analyze the degree of customer preference for a product, to formulate a more effective marketing strategy. In medical research, the Kendall coefficient can be used to analyze the relationship between the incidence of diseases and potential risk factors, helping doctors and researchers to better understand the causes of diseases and methods of prevention.

Suppose there is a set of data representing the math scores and physics scores of a group of students, and we want to know whether there is a correlation between these two variables. First, we need to rank these two variables, i.e., rank each student's performance in descending order. Then, we compare each student's math score ranking and physics score ranking for consistency, i.e., we calculate their similarity. Finally, by calculating the proportion of similarity, we can get the Kendall coefficient.

The Kendall coefficient is calculated by comparing the rank order of two variables and calculating the similarity between them. The Kendall coefficient has the following advantages: Firstly, the Kendall coefficient is not affected by the distribution of the data, and it is applicable to various types of data, including continuous data, discrete data, and ordered categorical data. Secondly, Kendall coefficient is not affected by outliers and has better robustness to outliers. In addition, Kendall coefficient can be used to compare the correlation between multiple variables, which can help us understand the relationship between variables more comprehensively. Therefore, it is very practical for calculating the correlation of various characteristic time series.

The spatial correlations among dimensions are calculated through Kendall coefficients (Abdi, 2007). Kendall coefficient is a statistic method that measures the level of correlation of rank variables. The Kendall coefficient between dimensions i and j can be calculated as:

$$KR_{ij} = \frac{c - d}{\frac{1}{2}n(n-1)}, \tag{5}$$

where c is the number of element pairs with consistency in i and j, d is the number of inconsistent element pairs, n is the length of time series. The value range of KR_{ij} is between -1 and 1.

A schematic diagram for constructing the Kendall matrix is shown in Fig. 5. The calculated method for the element K_{ij} in ith row and jth of the Kendall matrix K is shown in Eq. (6). In Eq. (6), c is the threshold value to determine whether the Kendall relationship is valid. If the significance p-value > 0.05 and $KR_{ij} > c$, K_{ij} is equal to KR_{ij}. If the significance p-value > 0.05 and $KR_{ij} < -c$, K_{ji} is equal to KR_{ij}. To make the Kendall matrix more sparse, K_{ij} is defined as 0 in otherwise.

$$K = \begin{cases} K_{ij} = KR_{ij}, & KR_{ij} > c \ and \ p > 0.05 \\ K_{ji} = KR_{ij}, & KR_{ij} < -c \ and \ p > 0.05 \\ K_{ij} = 0, & otherwise \end{cases} \tag{6}$$

3.5. Feature fusion

3.5.1. Fusion features in the time and frequency domains

MTSC_FF fuses the frequency domain features (Section 3.2) and the time domain features (Section 3.3) through a gate fusion mechanism as shown in Eq. (7). In Eq. (7), $\oplus$ is the element-wise addition, $E_{Attention}$ is the frequency domain features, T_{SSA} is time domain features, $dropout_1$ and $dropout_2$ is the dropout-based gate fusion mechanism.

$$ET = dropout_1(E_{Attention}) \oplus dropout_2(T_{SSA}) \tag{7}$$

3.5.2. Fusion features in different dimensions

MTSC_FF fuses features in different dimensions with the help of the Kendall matrix (refer to Section 3.4). The fusion process is shown in Fig. 6. In the fusion process, the features in different dimension are integrated into GIN (graph isomorphism network Xu, Hu, Leskovec, & Jegelka, 2018) with the Kendall matrix.

The final fused feature can be represented as a matrix $ET \in R_{l \times d}$, where d is the feature number in one dimension, l is the dimension number of the multivariate time series. The adjacency of the nodes is determined by the Kendall matrix K at firstly. And then, it be updated by the GIN model (Xu et al., 2018) with Eq. (8). In Eq. (8), k is the number of layers, ϵ^k is the updatable parameter, h_v is the current node feature, $N(v)$ denotes the node neighborhood set, and $h_u(u \in N(v))$ denotes the set of node neighborhood features.

$$h_v^{(k)} = MLP^k((1 + \epsilon^k) \cdot h_v^{k-1} + \sum_{\mu \in N(v)}^{h_u^{(k-1)}}) \tag{8}$$

3.6. Classification

Finally, MTSC_FF uses the final fusion features for classification by FC layer. In this stage, we apply FC to convert fusion features into class labels. Loss function can be calculated using Eq. (9).

$$\ell(X) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j} log p(\hat{y}_{i,j}) \tag{9}$$

In Eq. (9), y is the true class label, $\hat{y}$ is the predicted class label, N is the training set, and M is the number of labels.

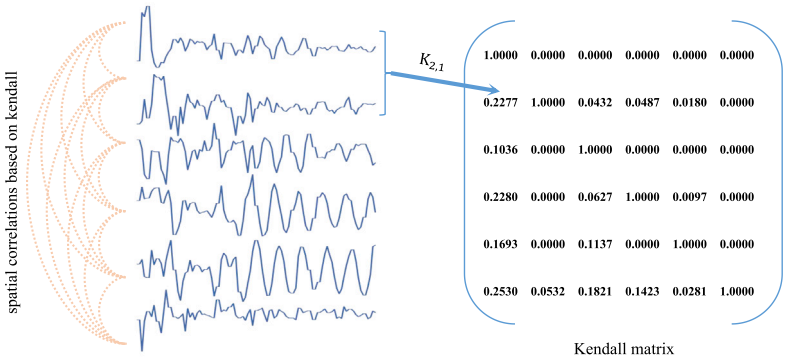
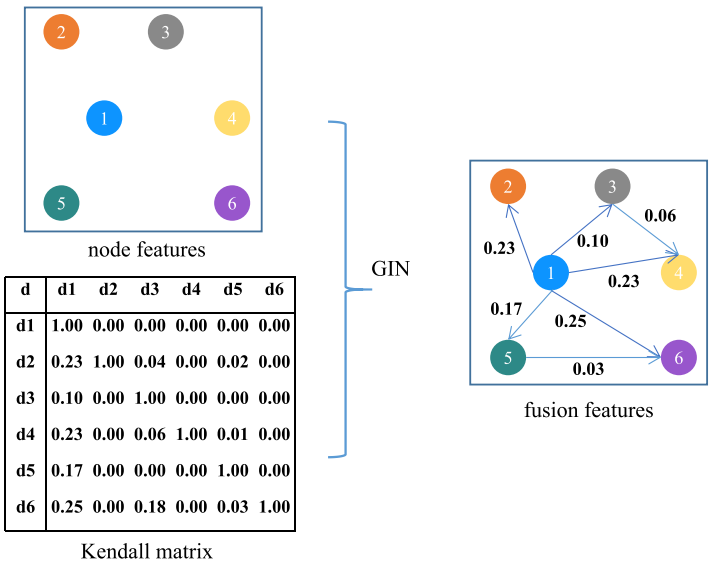


Fig. 5. The demo of Kendall matrix calculating.



d	d1	d2	d3	d4	d5	d6
d1	1.00	0.00	0.00	0.00	0.00	0.00
d2	0.23	1.00	0.04	0.00	0.02	0.00
d3	0.10	0.00	1.00	0.00	0.00	0.00
d4	0.23	0.00	0.06	1.00	0.01	0.00
d5	0.17	0.00	0.00	0.00	1.00	0.00
d6	0.25	0.00	0.18	0.00	0.03	1.00

Kendall matrix

Fig. 6. Fusion features in different dimensions with GIN. The colored dots represent the features in one dimension, and the value on the edges is the corresponding weights.

4. Experiment

4.1. Experimental setting

4.1.1. Datasets

We use 10 multivariate benchmark datasets in the UEA[1] archive for our comparison experiments. The datasets are mainly categorized as: Human Activity Recognition (HAR), Motion, Electrocardiogram (ECG), Electroencephalogram (EEG/MEG), and Audio Spectra (AS). More information of all dataset are presented in Table 1. Brief relevant information about the datasets are as follows:

ArticularyWordRecognition: 12 sensors were used in the dataset, each providing X, Y and Z time series positions with a sampling rate of 200 Hz. The sensors are located on the forehead, tongue; from tip to back in the midline, lips and jaw. The three head sensors (Head Center, Head Right, and Head Left) attached on a pair of glasses were used to calculate head-independent movement of other sensors. Of the total of 36 available dimensions, this dataset includes just 9.

AtrialFibrillation: This dataset of two-channel ECG recordings has been created with the goal of developing automated methods for predicting spontaneous termination of AF. The raw instances were 5 s segments of AF, containing two ECG signals, each sampled at 128 samples per second.

BasicMotions: The data was generated when four students performed four activities whilst wearing a smart watch. It consists of four classes, which are walking, resting, running and badminton. Participants were required to record motion a total of five times, and the data is sampled once every tenth of a second for a ten second period.

CharacterTrajectories: The dataset consists of 2858 character samples. The data was captured using a tablet. 3 Dimensions were kept x, y and pen tip force. The dataset has been numerically differentiated and Gaussian smoothed, with a sigma value of 2, and was captured at 200 Hz.

Cricket: Cricket requires an umpire to signal different events in the game to a distant scorer/bookkeeper. The signals are communicated with motions of the hands. The data, recorded at a frequency of 184 Hz,was collected by placing accelerometers on the wrists of the umpires. Each accelerometer has three synchronous measures for three axes (X, Y and Z).

EthanolConcentration: It is a dataset of raw spectra taken of water-and-ethanol solutions in 44 distinct, real whisky bottles. The concentrations of ethanol are 35%, 38%, 40%, and 45%. The classification problem is to determine the alcohol concentration of a sample contained within an arbitrary bottle. In this formulation, there are four classes, corresponding to the four concentrations.

FaceDetection: Our training data consist of MEG recordings and the class labels (Face/Scramble), from 10 subjects (subject01 to subject10), test data from 6 subjects (subject11 to 16). The data were down-sampled to 250 Hz and high-pass filtered at 1 Hz.

[1] Public datasets: http://timeseriesclassification.com/dataset.

Table 1
Relevant parameters about the 21 benchmark datasets.

Dataset	Type	Train	Test	Dimensions	Length	Classes
ArticularyWordRecognition	Motion	275	300	9	144	25
AtrialFibrillation	ECG	15	15	2	640	3
BasicMotions	HAR	40	40	6	100	4
CharacterTrajectories	Motion	1422	1436	3	182	20
Cricket	HAR	108	72	6	1197	12
EthanolConcentration	HAR	261	263	3	1751	4
FaceDetection	EEG/MEG	5890	3524	144	62	2
HandMovementDirection	EEG/MEG	160	74	10	400	4
Heartbeat	AS	204	205	61	405	2
JapaneseVowels	AS	270	370	12	29	9
Libras	HAR	180	180	2	45	15
LSST	Other	2459	2466	6	36	14
MotorImagery	EEG/MEG	278	100	64	3000	2
NATOPS	HAR	180	180	24	51	6
PEMS-SF	Other	267	173	963	144	7
PenDigits	Motion	7494	3498	2	8	10
SelfRegulationSCP1	EEG/MEG	268	293	6	896	2
SelfRegulationSCP2	EEG/MEG	200	180	7	1152	2
SpokenArabicDigits	AS	6599	2199	13	93	10
StandWalkJump	ECG	12	15	4	2500	3
UWaveGestureLibrary	HAR	120	320	3	315	8

HandMovementDirection: The dataset contains directionally modulated MEG activity, and it was recorded while subjects performed wrist movements in four different directions.

Heartbeat: The heart sound recordings were collected from different locations on the body. The typical four locations are aortic area, pulmonic area, tricuspid area and mitral area, but could be one of nine different locations. The sounds were divided into two classes: normal and abnormal. The normal recordings were from healthy subjects and the abnormal ones were from patients with a confirmed cardiac diagnosis.

JapaneseVowels: 9 Japanese-male speakers were recorded saying the vowels 'a' and 'e'. A '12-degree linear prediction analysis' is applied to the raw recordings to obtain time-series with 12 dimensions. In this dataset, instances have been padded to the longest length, 29. The classification task is to predict the speaker.

Libras: The dataset contains 15 classes of 24 instances each, where each class references to a hand movement type in LIBRAS. The hand movement is represented as a bidimensional curve performed by the hand in a period of time. The curves were obtained from videos of hand movements, with the Libras performance from 4 different people, during 2 sessions.

LSST: The Photometric LSST Astronomical Time Series Classification Challenge is an open data challenge to classify simulated astronomical time-series data in preparation for observations from the Large Synoptic Survey Telescope (LSST), which will achieve first light in 2019 and commence its 10-year main survey in 2022. These simulated time series, or light curves are measurements of an object brightness as a function of time — by measuring the photon flux in 6 different astronomical filters.

NATOPS: The dataset is generated by sensors on the hands, elbows, wrists and thumbs, and it is recorded in the x,y,z coordinates for each of the eight locations.

PenDigits: This is a handwritten digit classification task. 44 writers were asked to draw the digits (0...9), where instances are made up of the x and y coordinates of the pen traced across a digital screen.

SelfRegulationSCP1: The subject was asked to move a cursor up and down on a computer screen, while his cortical potentials were taken. During the recording, the subject received visual feedback of his slow cortical potentials. Cortical positivity lead to a downward movement of the cursor on the screen. Cortical negativity lead to an upward movement of the cursor. Each trial lasted 6 s.

SelfRegulationSCP2: The dataset was taken from an artificially respirated patient. The subject was asked to move a cursor on a computer screen, while his cortical potentials were taken. During the recording, the subject received auditory and visual feedback of his slow cortical potentials. The visual feedback was presented from second 2 to second 6.5. The sampling rate of 256 Hz and the recording length of 4.5 s results in 1152 samples per channel for every trial.

SpokenArabicDigits: Dataset from 8800 (10 digits × 10 repetitions × 88 speakers) time series of 13 Frequency Cepstral Coefficients had taken from 44 males and 44 females Arabic native speakers between the ages 18 and 40 to represent ten spoken Arabic digit.

StandWalkJump: Short duration ECG signals are recorded from a healthy 25-year-old male performing different physical activities to study the effect of motion artifacts on ECG signals and their sparsity. The raw data was sampled at 500 Hz, with a resolution of 16 bits. A Spectrogram of each instance was then created with a window size of 0.061 s and an overlap of 70%. There are 3 classes: standing, walking and jumping, each consists of 9 instances.

UWaveGestureLibrary: A set of eight simple gestures generated from accelerometers. The data consists of the X,Y,Z coordinates of each motion. Each series is 315 long.

4.1.2. Experimental parameters

In all experiments, MTSC_FF uses 1 layer of attention layer, sparse self-attention layer and GIN. Our experiments are run in python and pytorch. Cross-entropy loss function was used for minimization and Adam optimizer was used for optimization. The experimental results were obtained through 50 epochs.

4.1.3. Reproducibility

For reproducibility, codes and relevant parameters were released out on Github.[2] The results can be independently replicated.

4.2. Experimental performance

4.2.1. Comparison experiments

We compare the MTSC_FF with the following method:

- ED-1NN (Chen et al., 2013): a kind of traditional Euclidean distance based algorithm.
- DTW-1NN (Chen et al., 2013): a variant of the semi-supervised Dynamic Time Warping (DTW) algorithm.
- MLSTM-FCN (Karim et al., 2019), combining the LSTM, FCN and Attention block into a MTSC model by augmenting the FCN block with a squeeze-and-excitation block to further improve accuracy.

[2] https://github.com/dumingsen/MTSC_FF.

Table 2
Accuracy comparison results.

Dataset	SMATE	TapNet	MLSTM-FCN	WEASEL+MUSE	1NN-ED	1NN-DTW	DA-Net	MR-PETSC	MF-Net	MTSC_FF
ArticularyWord-Recognition	0.993	0.987	0.973	0.990	0.970	0.980	0.980	**0.997**	0.983	0.983
Atrial-Fibrillation	0.133	0.333	0.267	0.333	0.267	0.267	0.467	0.400	0.466	**0.533**
BasicMotions	**1.000**	**1.000**	0.950	**1.000**	0.675	**1.000**	0.925	**1.000**	0.950	0.950
Character-Trajectories	0.984	0.997	0.985	0.990	0.964	0.969	**0.998**	0.941	0.958	0.986
Cricket	0.986	0.958	0.917	**1.000**	0.944	0.986	0.861	**1.000**	0.944	0.986
Ethanol-Concentration	0.399	0.323	0.373	0.430	0.293	0.304	**0.874**	0.555	0.250	0.293
FaceDetection	0.647	0.556	0.545	0.545	0.519	0.513	0.648	0.574	0.664	**0.670**
HandMovement-Direction	**0.554**	0.378	0.365	0.365	0.279	0.306	0.365	0.338	0.500	0.541
Heartbeat	0.741	**0.751**	0.663	0.727	0.620	0.659	0.624	0.702	0.682	0.693
JapaneseVowels	0.965	0.965	0.976	0.973	0.924	0.959	0.938	N/A	0.970	**0.978**
Libras	0.849	0.850	0.856	0.878	0.833	**0.894**	0.800	0.845	0.850	0.861
LSST	0.582	0.568	0.373	**0.590**	0.456	0.575	0.560	0.560	0.468	0.478
MotorImagery	**0.590**	**0.590**	0.510	0.510	0.390	N/A	0.500	0.490	0.540	0.550
NATOPS	0.922	**0.939**	0.889	0.870	0.860	0.850	0.878	0.917	0.927	0.889
PEMS-SF	0.803	0.751	0.699	N/A	0.705	0.734	0.867	0.861	**0.884**	**0.884**
PenDigits	0.980	0.980	0.978	0.948	0.973	0.939	0.980	0.905	**0.983**	0.979
SelfRegulation-SCP1	0.887	0.739	0.874	0.710	0.771	0.765	0.924	0.788	0.911	**0.928**
SelfRegulation-SCP2	**0.567**	0.550	0.472	0.460	0.483	0.533	0.561	0.533	0.533	0.511
SpokenArabic-Digits	0.979	0.983	**0.990**	0.982	0.967	0.960	0.980	0.960	**0.990**	**0.990**
StandWalkJump	**0.533**	0.400	0.067	0.333	0.200	0.333	0.400	0.400	0.400	0.467
UWaveGesture-Library	0.897	0.894	0.891	**0.916**	0.881	0.868	0.833	0.800	0.862	0.875
ACC	0.761	0.738	0.696	0.728	0.665	0.720	0.760	0.728	0.748	**0.763**
Win	5	4	1	4	0	2	2	3	3	**6**

The classifiers compared in the table are as follows: ED-1NN (Chen, Hu, Keogh, & Batista, 2013), DTW-1NN (Chen et al., 2013), MLSTM-FCN (Karim et al., 2019), WEASEL+MUSE (Schäfer & Leser, 2017), TapNet (Zhang, Gao, Lin, & Lu, 2020), MR-PETSC (Feremans, Cule, & Goethals, 2022), SMATE (Zuo et al., 2021), DA-Net (Chen et al., 2022) and MF-Net (Du et al., 2023).

- WEASEL+MUSE (Schäfer & Leser, 2017): its novelty lies in specific way of extracting and filtering multivariate features from MTS by encoding context information into each feature.
- TapNet (Zhang et al., 2020): designing a random group permutation method combined with multi-layer convolutional networks to learn the low-dimensional features from MTS.
- MR-PETSC (Feremans et al., 2022): it constructs an embedding based on sequential pattern occurrences and learn a linear model. The discovered patterns form the basis for interpretable insight into each class of time series.
- SMATE (Zuo et al., 2021): a novel semi-supervised model for learning the interpretable spatio-temporal representation from weakly labeled MTS.
- DA-Net (Chen et al., 2022): a novel network based on dual attention to mine the local–global features for MTSC.
- MF-Net (Du et al., 2023): a novel network based on self-attention and GNN to mine the local–global-spatial based multi-features.

Table 2 shows the accuracy of these methods. The results of comparison methods are from SMATE (Zuo et al., 2021), DA-Net (Chen et al., 2022), MR-PETSC (Feremans et al., 2022) and MF-Net (Du et al., 2023). In Table 2, "AVG" denotes the average accuracy achieved by the corresponding classifier on 21 datasets, "Win" denotes the number of datasets where the corresponding classifier got the best accuracy, and the highest accuracy for each dataset is bolded. "N/A" denotes that the corresponding methodology fails to execute the results.

On the basis of the comparison experimental results in Table 2, we can observe obviously that MTSC_FF achieves 6 wins on 21 datasets and MTSC_FF has the highest average accuracy among these methods.

To show the comparison results more intuitively, we performed post hoc test nemenyi (Benavoli, Corani, & Mangili, 2016) based on the rank of different datasets, and we give the critical difference (CD) plot according to the accuracy column of each method in Table 2. The CD plot ranks the 21 MTSC methods in ascending order. As shown in Fig. 7, MTSC_FF has the second smallest rank. The CD diagram likewise illustrates that our method is in the first rank.

By pairwise wilcoxon signed-rank test (Benavoli et al., 2016), the p-value between MTSC_FF and SMATE is 0.148, which is larger and indicates no significant difference between the two methods. SMATE only used spatio-temporal dynamic features in MTS, proved that the temporal dependency and the evolution of the spatial interactions are important for building a reliable MTS embedding. However SMATE ignored the frequency features. TapNet only utilizes local features based on the time domain, and extracting some dimensions with respect to a single sample as a whole. The spatial correlation of the whole is inevitably lost by performing random group alignment. Our method is similar to TapNet and SMATE in terms of accuracy, but we make fuller use of the fusion of time and frequency domain features and spatial correlation features, and we visualize the various features proposed to explain the classification, as demonstrated by the interpretability study in Section 4.2.2.

4.2.2. Feature interpretability study

Interpretation of time domain features. In this section, we use Grad-CAM (gradient-weighted class activation mapping) (Selvaraju et al., 2017) to visualize the time domain-based long-range dependency features of AtrialFibrillation dataset. As shown in Fig. 8, MTSC_FF captures significant local features (green dashed boxed region). The visualization clearly illustrates the ability to capture long-range dependency on a cycle of the AtrialFibrillation dataset, and by observing the activation state of a time segment, we can observe the contribution of a particular cycle or segment to the classification. As shown in Fig. 8, the activation states of time segments in cycle 3 and 4 are brighter and therefore contribute the most to classification. More detailedly, the activation states of cycle 1 and cycle 4 are plotted, and the green dashed box region contributes the most. The long-range dependency features can be extracted by sparse self-attention layer, which can improve the classification accuracy.

Interpretation of frequency domain features. As shown in Fig. 9, to visualize the activation status of the frequency domain features, we also visualize the frequency domain modes corresponding to the time series of each dimension of AtrialFibrillation using Grad-CAM. The brighter colors in Figs. 9(b) and 9(d) indicate more contribution to the classification results, while Figs. 9(a) and 9(c) shows the original frequency domain data. The bright colors in Figs. 9(a) and 9(c) are the most distinguishing features of AtrialFibrillation dataset, indicating the most contribution in performing classification, which corresponds to the original time–frequency image. The attention layer allows to focus on more differentiated features, which can reduce feature redundancy, reduce computational complexity, and improve accuracy.

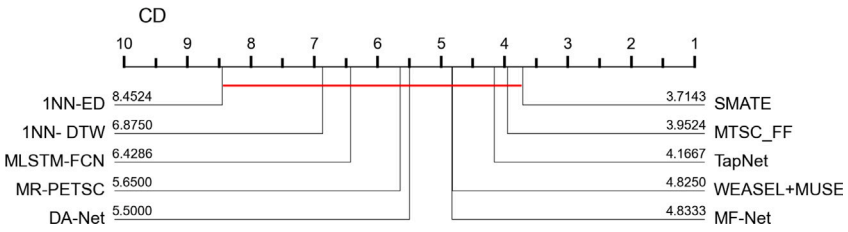


Fig. 7. CD diagram of 10 implementations on 21 UEA datasets.

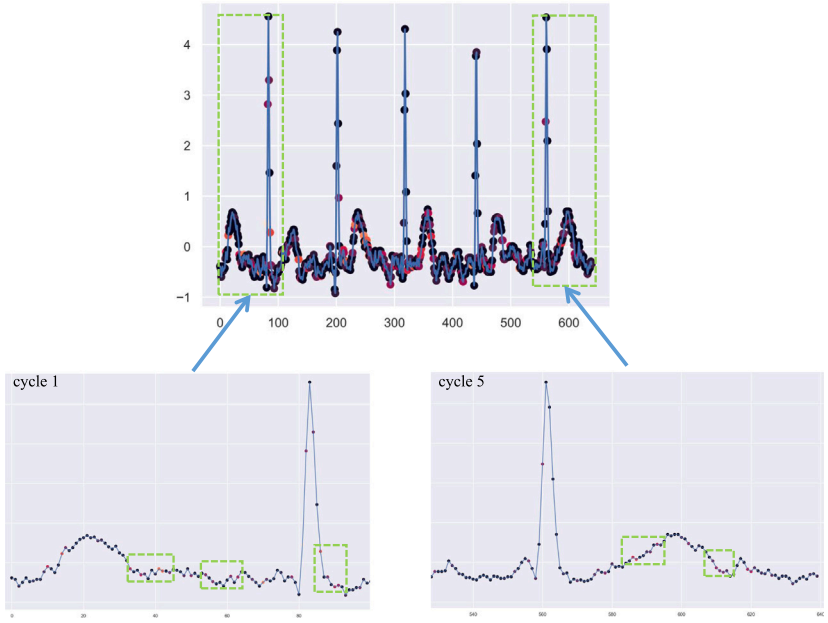
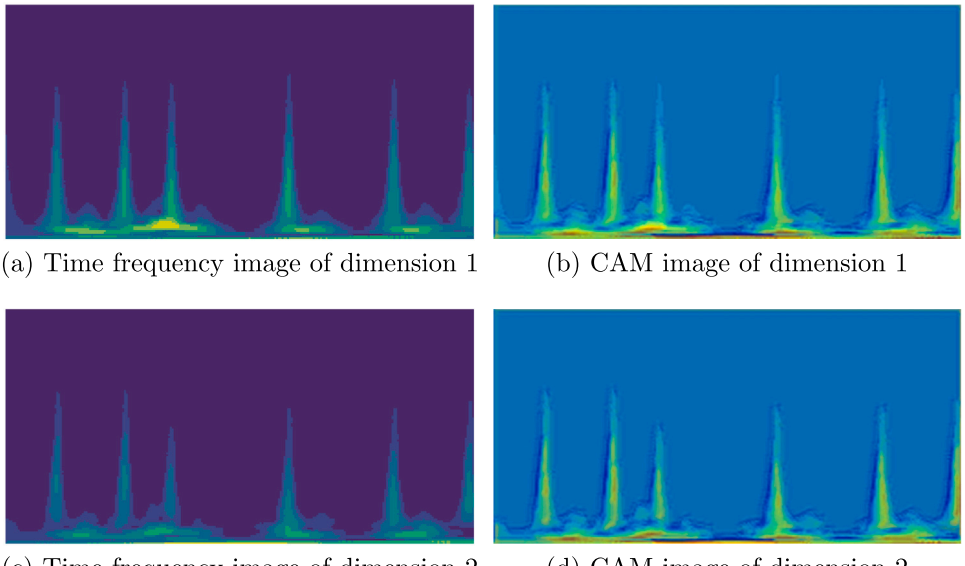


Fig. 8. Visualization of long-range dependency features.

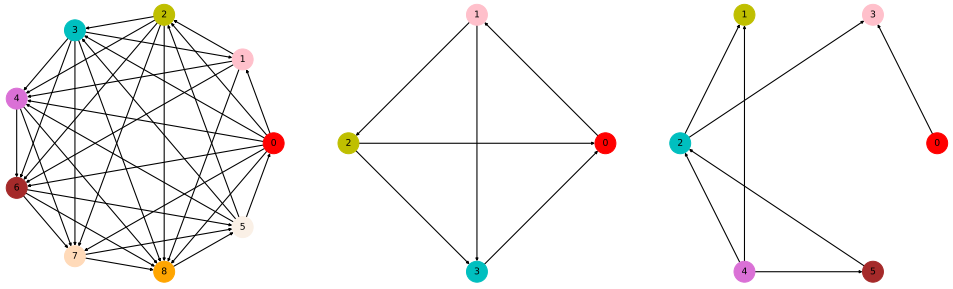


(a) Time frequency image of dimension 1 (b) CAM image of dimension 1

(c) Time frequency image of dimension 2 (d) CAM image of dimension 2

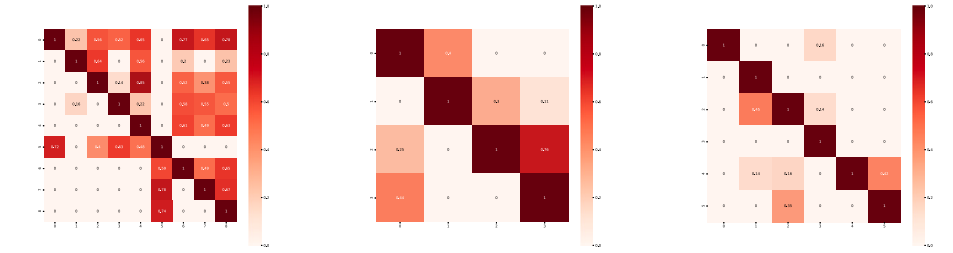
Fig. 9. (a) and (c) is the original time–frequency image, corresponding to dimension 1 and dimension 2, respectively; and (b) and (d) is the activation status image, corresponding to dimension 1 and dimension 2, respectively.

Interpretation of spatial correlations. Fig. 10 shows the visualization of Kendall spatial correlations based on three datasets: ArticularyWordRecognition with 9 dimensions, StandWalkJump with 4 dimensions, and BasicMotions with 6 dimensions, respectively. The Kendall matrix based on Kendall coefficient describes the level of correlation among the multivariate time series dimensions, which can further represent the connection between the dimensions. For example, each dimension in the ArticularyWordRecognition dataset represents



(a) ArticularyWordRecognition (b) StandWalkJump (c) BasicMotions

Fig. 10. Kendall spatial correlation visualization, with colored dots indicating individual dimensions.



(a) ArticularyWordRecognition (b) StandWalkJump (c) BasicMotions

Fig. 11. Kendall spatial based heatmap.

the time series measured by sensors on parts such as the tongue and lips during human vocalization, and we can understand which parts are more tightly linked during the vocal process. Through GIN, we can obtain spatial correlation based on the Kendall matrix, and the reasonable use of spatial correlation can have more positive effect for classification to obtain higher accuracy. In Fig. 11, we visualize the level of dependency between dimensions more intuitively through heatmaps.

4.3. Comparison of features of domains

Time domain (OTD), frequency domain (OFD), time domain+frequency domain (OTF), time domain+spatial correlation (OTS) and frequency domain+spatial correlation (OFS) features are separately individually used to verify the effectiveness of each part that makes up MTSC_FF on the accuracy. The sensitivity experiments were performed on the 21 datasets in Table 1. Table 3 shows the comparison among the each kind of feature. From the results in Table 3 and Fig. 13, the average accuracy of MTSC_FF has improved and higher accuracy has been obtained. Thus the three parts of features together proved to have a positive impact on the study. And we can find that the accuracy of the column of data based on the combination of features (OTF, OFS, OTS) are higher than the average of the individual features, thus validating the effectiveness of the individual feature domains in Table 3 and Fig. 13. In Fig. 13, each colored point indicates a dataset. The closer to the upper left corner the point is, the better MTSC_FF performs. Thus, the point in the area below the $y = x$ line indicates poor performance of MTSC_FF.

In Table 3, we can find that NATOPS works better using only time domain data compared to other models. NATOPS is generated by sensors on the hand, elbow, wrist and thumb on the left and right sides of the body. And these data are x, y and z coordinates of the corresponding positions respectively. Thus the dataset has (8 sensors $*$ 3 = 24) 24 channels or dimensions. Due to the rich spatial correlation of the data, it is important to utilize the correlation appropriately. The Kendall coefficient can calculate the similarity based on the consistency. Although NATOPS has a large number of channels for spatial information, it does not have a richer time or frequency domain information to complement it. The length of NATOPS is 51, and due to the lack of significant periodical variations, only the frequency domain features are not sufficient. Therefore only frequency domain features are not sufficient. Finally, the large dimensionality (as shown in Fig. 12) as well as the non-stationary properties result in a fused ground accuracy relative to using only time-domain features.

4.4. Epoch analysis for MTSC_FF

To analyze the changing of MTSC_FF with epoch during the training process, we obtain the loss and accuracy diagrams for epoch range of 1–50. From Figs. 14 and 15, we show the AtrialFibrillation and StandWalkJump accuracy and loss diagrams. Figs. 14 and 15 show that: (1) the training and testing losses fluctuate in a small range and the accuracy increases as the epoch increases. (2) when the epoch is over 20, the training and testing accuracy converge, thus illustrating the good performance of MTSC_FF. The number of training and testing for StandWalkJump is 15 and 12 respectively. Therefore there are not enough samples to support our model to get enough features to achieve convergence, which ultimately leads to a large and fluctuating test_loss.

5. Conclusion

In this paper, we proposed MTSC_FF to improve the MTSC accuracy with the fusion features. Firstly, MTSC_FF extracts the frequency domain features through an attention layer with the help of continuous wavelet transform. In parallel, MTSC_FF uses a sparse self-attention layer to extract long-range dependency features from the time domain. At the same time, MTSC_FF obtains the spatial correlations among the multivariate time series dimensions through the Kendall coefficient.

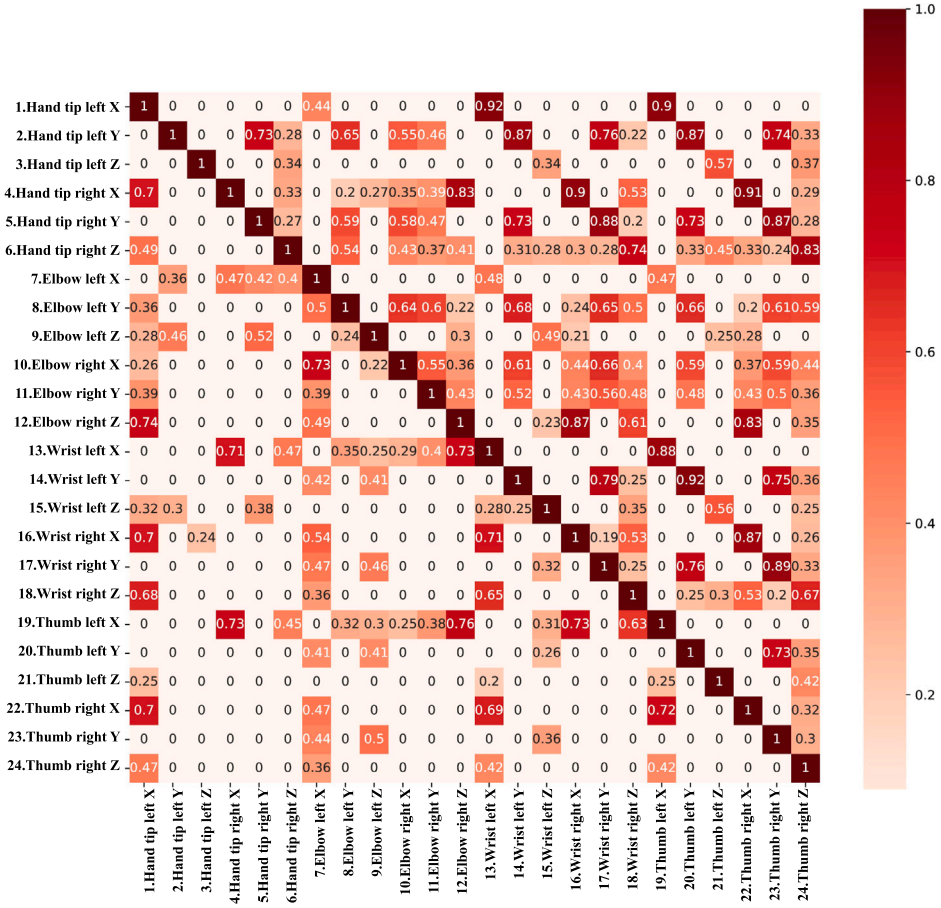
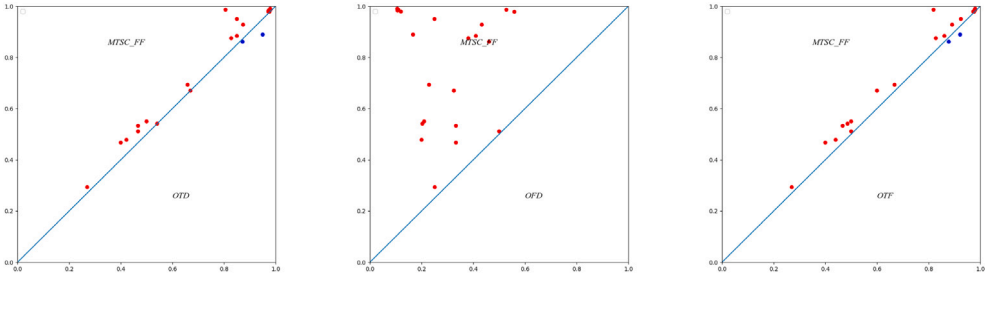
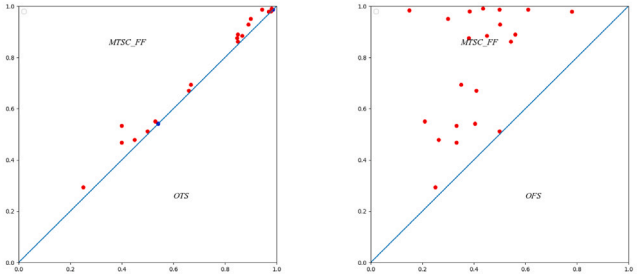


Fig. 12. Heatmap of NATOPS.

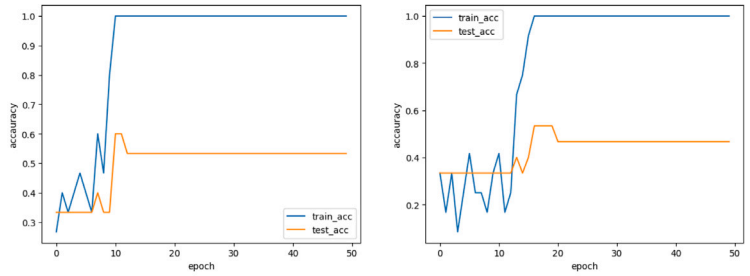


(a) MTSC_FF vs OTD (b) MTSC_FF vs OFD (c) MTSC_FF vs OTF



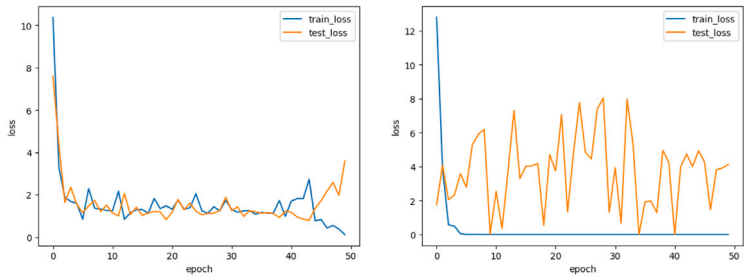
(d) MTSC_FF vs OTS (e) MTSC_FF vs OFS

Fig. 13. MTSC_FF and the accuracy based on other domain features are pairwise compared.



(a) AtrialFibrillation accauracy (b) StandWalkJump accauracy

Fig. 14. Accuracy.



(a) AtrialFibrillation loss (b) StandWalkJump loss

Fig. 15. Loss.

Table 3
Ablation study of OTD, OFD, OTF, OTS, OFS and MTSC_FF on 21 UEA datasets.

Dataset	OTD	OFD	OTF	OTS	OFS	MTSC_FF
ArticularyWordRecognition	0.973	0.107	0.977	0.980	0.150	**0.983**
AtrialFibrillation	0.467	0.333	0.467	0.400	0.333	**0.533**
BasicMotions	0.850	0.250	0.925	0.900	0.300	**0.950**
CharacterTrajectories	0.980	0.110	0.981	**0.986**	0.500	**0.986**
Cricket	0.806	0.528	0.819	0.944	0.611	**0.986**
EthanolConcentration	0.270	0.251	0.270	0.251	0.251	**0.293**
FaceDetection	**0.670**	0.325	0.600	0.660	0.410	**0.670**
HandMovementDirection	**0.541**	0.203	0.486	**0.541**	0.405	**0.541**
Heartbeat	0.659	0.229	0.668	0.668	0.351	**0.693**
JapaneseVowels	0.976	0.559	0.976	0.970	0.781	**0.978**
Libras	0.872	0.461	**0.878**	0.850	0.544	0.861
LSST	0.422	0.200	0.440	0.450	0.264	**0.478**
MotorImagery	0.500	0.210	0.500	0.530	0.210	**0.550**
NATOPS	**0.950**	0.167	0.922	0.850	0.561	0.889
PEMS-SF	0.850	0.410	0.861	0.867	0.451	**0.884**
PenDigits	0.972	0.120	0.973	0.978	0.384	**0.979**
SelfRegulationSCP1	0.874	0.433	0.891	0.891	0.502	**0.928**
SelfRegulationSCP2	0.467	0.500	0.500	0.500	0.500	**0.511**
SpokenArabicDigits	0.979	0.106	0.980	0.981	0.436	**0.990**
StandWalkJump	0.400	0.333	0.400	0.400	0.333	**0.467**
UWaveGestureLibrary	0.828	0.381	0.828	0.847	0.381	**0.875**
AVG acc	0.729	0.296	0.731	0.735	0.412	**0.763**
Win	3	0	1	2	0	**20**

And then, all the features are fused by means of GNN. Finally, the fusion features are used to predict the classification labels through the fully connected layer. Experimental results on the UEA datasets show that the proposed method has high accuracy. And the proposed method can easily visualize the classification-dependent features, thus enhancing interpretability.

Although the related work has been completed, there are still aspects that can be improved. We have explored only time, frequency domain and spatial correlation features, but not other types of features. Thus, We will explore more types of multivariate time series features and the positive effects for MTSC. In future work, we hope that some improvements can be made in the fusion of time and frequency domain features so that both can be fused more appropriately to make them work better together for MTSC. In terms of spatial correlation, we would like to explore more methods to obtain spatial correlation among various dimensions of multivariate time series. We believe that the accuracy of MTSC can be further improved in the future work.

CRediT authorship contribution statement

Mingsen Du: Conceptualization, Methodology, Validation, Writing – original draft, Writing – review & editing. **Yanxuan Wei:** Methodology, Validation, Writing – original draft, Writing – review & editing. **Yupeng Hu:** Methodology, Validation, Writing – original draft, Writing – review & editing. **Xiangwei Zheng:** Supervision, Project administration. **Cun Ji:** Methodology, Validation, Writing – original draft, Writing – review & editing, Supervision, Project administration.

Declaration of competing interest

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

Data availability

The datasets used or analyzed during the current study are available from the UEA archive: http://timeseriesclassification.com.

Acknowledgments

This work was supported by the Innovation Methods Work Special Project under Grant 2020IM020100, and the Natural Science Foundation of Shandong Province, China under Grant ZR2020QF112. We would like to thank Eamonn Keogh and his team, Tony Bagnall and his team for the UEA/UCR time series classification repository.

References

Aach, J., & Church, G. M. (2001). Aligning gene expression time series with time warping algorithms. *Bioinformatics*, *17*(6), 495–508.

Abdi, H. (2007). The Kendall rank correlation coefficient. In *Encyclopedia of measurement and statistics* (pp. 508–510). Thousand Oaks, CA: Sage.

Batal, I., & Hauskrecht, M. (2009). A supervised time series feature extraction technique using dct and dwt. In *2009 international conference on machine learning and applications* (pp. 735–739). IEEE.

Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150.

Benavoli, A., Corani, G., & Mangili, F. (2016). Should we really use post-hoc tests based on mean-ranks? *Journal of Machine Learning Research*, *17*(1), 152–161.

Chambon, S., Galtier, M. N., Arnal, P. J., Wainrib, G., & Gramfort, A. (2018). A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *26*(4), 758–769.

Chen, Y., Hu, B., Keogh, E., & Batista, G. E. (2013). Dtw-d: time series semi-supervised learning from a single example. In *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 383–391).

Chen, Z., Liu, Y., Zhu, J., Zhang, Y., Jin, R., He, X., et al. (2021). Time-frequency deep metric learning for multivariate time series classification. *Neurocomputing*, *462*, 221–237.

Chen, R., Yan, X., Wang, S., & Xiao, G. (2022). DA-Net: Dual-attention network for multivariate time series classification. *Information Sciences*, *610*, 472–487.

Ding, C., Sun, S., & Zhao, J. (2023). MST-GAT: A multimodal spatial–temporal graph attention network for time series anomaly detection. *Information Fusion*, *89*, 527–536.

Du, M., Wei, Y., Zheng, X., & Ji, C. (2023). Multi-feature based network for multivariate time series classification. *Information Sciences*, Article 119009.

Duan, Z., Xu, H., Wang, Y., Huang, Y., Ren, A., Xu, Z., et al. (2022). Multivariate time-series classification with hierarchical variational graph pooling. *Neural Networks*, *154*, 481–490.

El-Sappagh, S., Abuhmed, T., Islam, S. R., & Kwak, K. S. (2020). Multimodal multitask deep learning model for Alzheimer's disease progression detection based on time series data. *Neurocomputing*, *412*, 197–215.

Feremans, L., Cule, B., & Goethals, B. (2022). PETSC: pattern-based embedding for time series classification. *Data Mining and Knowledge Discovery*, *36*(3), 1015–1061.

Hao, Y., & Cao, H. (2020). A new attention mechanism to classify multivariate time series. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence*.

Hong, B., Yan, Z., Chen, Y., et al. (2022). Long memory gated recurrent unit for time series classification. *volume 2278*, In *Journal of physics: conference series*. IOP Publishing, Article 012017.

Huang, X., Zhang, F., Fan, H., & Xi, L. (2021). Multimodal adversarial learning based unsupervised time series anomaly detection. *Journal of Computer Research and Development*, *58*(08), 1655–1667.

Ircio, J., Lojo, A., Mori, U., & Lozano, J. A. (2020). Mutual information based feature subset selection in multivariate time series classification. *Pattern Recognition*, *108*, Article 107525.

Iwana, B. K., & Uchida, S. (2020). Time series classification using local distance-based features in multi-modal fusion networks. *Pattern Recognition*, *97*, Article 107024.

Ji, C., Du, M., Hu, Y., Liu, S., Pan, L., & Zheng, X. (2022). Time series classification based on temporal features. *Applied Soft Computing*, *128*, Article 109494.

Jiang, H., Liu, L., & Lian, C. (2022). Multi-modal fusion transformer for multivariate time series classification. In *2022 14th international conference on advanced computational intelligence* ICACI, (pp. 284–288). IEEE.

Karim, F., Majumdar, S., Darabi, H., & Harford, S. (2019). Multivariate LSTM-FCNs for time series classification. *Neural Networks*, *116*, 237–245.

Li, D., Bissyande, T. F. D. A., Klein, J., & Le Traon, Y. (2016). Time series classification with discrete wavelet transformed data: Insights from an empirical study. In *The 28th international conference on software engineering and knowledge engineering (SEKE 2016)*.

Li, S., Chowdhury, R. R., Shang, J., Gupta, R. K., & Hong, D. (2021). Units: Short-time fourier inspired neural networks for sensory time series classification. In *Proceedings of the 19th ACM conference on embedded networked sensor systems* (pp. 234–247).

Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y.-X., et al. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in Neural Information Processing Systems*, *32*.

Li, Z., Jin, X., & Zhao, X. (2015). Drunk driving detection based on classification of multivariate time series. *Journal of Safety Research*, *54*, 61–e29.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., et al. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022).

Liu, M., Ren, S., Ma, S., Jiao, J., Chen, Y., Wang, Z., et al. (2021). Gated transformer networks for multivariate time series classification. arXiv preprint arXiv:2103.14438.

Ma, H., Li, W., Zhang, X., Gao, S., & Lu, S. (2019). AttnSense: Multi-level attention mechanism for multimodal human activity recognition. In *IJCAI* (pp. 3109–3115).

Ma, Q., Tian, S., Wei, J., Wang, J., & Ng, W. W. (2019). Attention-based spatio-temporal dependence learning network. *Information Sciences*, *503*, 92–108.

Prieto, O. J., Alonso-González, C. J., & Rodríguez, J. J. (2015). Stacking for multivariate time series classification. *Pattern Analysis and Applications*, *18*, 297–312.

Schäfer, P., & Leser, U. (2017). Multivariate time series classification with WEASEL+MUSE. arXiv preprint arXiv:1711.11343.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618–626).

Shao, J., Huang, Z., & Zhu, J. (2020). Transfer learning method based on adversarial domain adaption for bearing fault diagnosis. *IEEE Access*, *8*, 119421–119430.

Tormene, P., Giorgino, T., Quaglini, S., & Stefanelli, M. (2009). Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation. *Artificial Intelligence in Medicine*, *45*(1), 11–34.

Wang, B., Jiang, T., Zhou, X., Ma, B., Zhao, F., & Wang, Y. (2020). Time-series classification based on fusion features of sequence and visualization. *Applied Sciences*, *10*(12), 4124.

Wang, S., Li, B. Z., Khabsa, M., Fang, H., & Ma, H. (2020). Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768.

Wang, H., Liu, Y., Wang, D., Luo, Y., Tong, C., & Lv, Z. (2022). Discriminative and regularized echo state network for time series classification. *Pattern Recognition*, *130*, Article 108811.

Woo, S., Park, J., Lee, J.-Y., & Kweon, I. S. (2018). Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision* ECCV, (pp. 3–19).

Xiao, Z., Xu, X., Xing, H., Luo, S., Dai, P., & Zhan, D. (2021). RTFN: a robust temporal feature network for time series classification. *Information Sciences*, *571*, 65–86.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks? arXiv preprint arXiv:1810.00826.

Yang, D., Chen, H., Song, Y., & Gong, Z. (2017). Granger causality for multivariate time series classification. In *2017 IEEE international conference on big knowledge* ICBK, (pp. 103–110). IEEE.

Yang, W., Yuan, J., & Wang, X. (2022). SFCC: Data augmentation with stratified Fourier coefficients combination for time series classification. *Neural Processing Letters*, 1–14.

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* (pp. 947–956).

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

Yu, Y., Zeng, X., Xue, X., & Ma, J. (2022). LSTM-based intrusion detection system for VANETs: A time series classification approach to false message detection. *IEEE Transactions on Intelligent Transportation Systems*, *23*(12), 23906–23918.

Zha, D., Lai, K.-H., Zhou, K., & Hu, X. (2022). Towards similarity-aware time-series classification. In *Proceedings of the 2022 SIAM international conference on data mining* SDM, (pp. 199–207). SIAM.

Zhang, X., Gao, Y., Lin, J., & Lu, C.-T. (2020). Tapnet: Multivariate time series classification with attentional prototypical network. *volume 34*, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 6845–6852).

Zhang, Y., Hou, Y., OuYang, K., & Zhou, S. (2022). Multi-scale signed recurrence plot based time series classification using inception architectural networks. *Pattern Recognition*, *123*, Article 108385.

Zuo, J., Zeitouni, K., & Taher, Y. (2021). Smate: Semi-supervised spatio-temporal representation learning on multivariate time series. In *2021 IEEE international conference on data mining* ICDM, (pp. 1565–1570). IEEE.