Information Sciences 610 (2022) 472–487

Contents lists available at ScienceDirect

Information Sciences

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

DA-Net: Dual-attention network for multivariate time series classification

Rongjun Chen [a], Xuanhui Yan [a,*], Shiping Wang [b], Guobao Xiao [c]

[a] Fujian Internet of Things Laboratory for Environmental Monitoring, School of Computer and Cyberspace Security, Fujian Normal University, Fuzhou, Fujian, China
[b] College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China
[c] Electronic Information and Control Engineering Research Center of Fujian, College of Computer and Control Engineering, Minjiang University, Fuzhou 350108, China

ARTICLE INFO

Article history:
Received 23 February 2022
Received in revised form 27 July 2022
Accepted 30 July 2022
Available online 8 August 2022

Keywords:
Multivariate time series classification
Deep learning
Attention
UEA datasets

ABSTRACT

Multivariate time series classification is one of the increasingly important issues in machine learning. Existing methods focus on establishing the global long-range dependencies or discovering the local critical sequence fragments. However, they often ignore the combined information from both global and local features. In this paper, we propose a novel network (called DA-Net) based on dual attention to mine the local–global features for multivariate time series classification. Specifically, DA-Net consists of two distinctive layers, i.e., the Squeeze-Excitation Window Attention (SEWA) layer and the Sparse Self-Attention within Windows (SSAW) layer. For the SEWA layer, we capture the local window-wise information by explicitly establishing window dependencies to prioritize critical windows. For the SSAW layer, we preserve rich activate scores with less computation to widen the window scope for capturing global long-range dependencies. Based on the two elaborated layers, DA-Net can mine critical local sequence fragments in the process of establishing global long-range dependencies. The experimental results show that DA-Net is able to achieve competing performance with state-of-the-art approaches on the multivariate time series classification.

1. Introduction

A time series, which is ubiquitous in a real-valued and ordered world, is considered one of the most challenging problems in data mining [1,2]. The main tasks of time series data mining include, but are not limited to: time series prediction [3], anomaly detection [4], clustering [5], and classification [6]. In particular, time series classification can be classified into Univariate Time Series Classification (UTSC) and Multivariate Time Series Classification (MTSC). Data generated by a single sensor are referred to as univariate time series, and data generated by multiple sensors are referred to as multivariate time series. MTSC has been widely used in real-life applications. For instance, in daily life, wearable mobile sensors collect the behavioral activities of the human body like running, walking and swimming during sports to improve people's lives in multiple fields such as environmental assisted living, elderly rehabilitation, and smart home [7]. In medical science, the classification of biomedical signals is used to help patients to control the prosthetic hands [8]. In computer science, engineers

* Corresponding author.
E-mail address: yan@fjnu.edu.cn (X. Yan).

https://doi.org/10.1016/j.ins.2022.07.178
0020-0255/

monitor and analyze data from power systems, which helps to find faults in time and advance the safety [9]. There is an urgent demand for the development of MTSC in real-life applications.

A vast variety of MTSC approaches have been developed in the literature. They can de divided into traditional machine learning-based methods (e.g., Shapelet [6], DTW-1NN [10], Boss [11] and TSF [12]) and deep learning-based methods (e.g., TapNet [13], Multi-Channel MHLF [14], TFDM [15] and MLSTM-FCN [16]). Deep learning-based methods are able to handle the issue concerning low-dimensional features mapping into high-dimensional directly from raw time series data, and they require less domain expertise than traditional machine learning-based methods for MTSC. These deep learning-based methods perform well, but some drawbacks apply to long sequences. Long sequence data are ubiquitous in daily life. For example, speech data are often collected with a sampling frequency of 8 kHz, including 480 k sample points in a 1-min sound stream. While some deep learning-based methods, such as TapNet and TFDM, are based on Convolutional Neural Network (CNN) [17], which uses a limited convolutional kernel size and cannot capture the information of the global long-range dependencies. There are also methods, such as Multi-Channel MHLF and MLSTM-FCN, which try to use Long Short-Term Memory (LSTM) [18] to address this problem. However, they classify the time series in a step-by-step manner, which affects the inference speed and causes the vanishing gradient with increasing sequence length.

In this paper, we propose to introduce Transformer-like applications [19,20] for MTSC to handle the global long-range dependencies. Transformer-like applications are a single-structured architecture combining encoders and decoders. They utilize self-attention mechanism to extract the global relational contexts and ameliorate the vanishing gradient problem. Among these Transformer-based methods, Swin Transformer [20] is one of the best-characterized Transformers because of its effectiveness in scaling down architecture. Swin Transformer introduces the hierarchical feature maps and applies the window partitioning scheme to reduce the time complexity of Transformer. It also employs the shifted windowing mechanism, which confines the self-attention computation in non-overlapping local windows to improve classification capacity while permitting cross-window connection. Thus, we adopt Swin Transformer to deal with the expensive calculation complexity, feature interaction, and, foremost, long-range dependencies for the global context of MTSC.

However, there are two problems hindering the implementation. The first problem is that, the local discriminating features are ignored in the implementation process. While it is essential to mine the global long-range dependencies, the importance of local features is also of great interest. Many previous works [1,21,22] indicate that local patterns in the time series can offer the distinguishing features for MTSC. Ye and Keogh [6] proposed to find the distinguishing subsequences, namely time series shapelets, which are in some sense maximally representative of a class. They demonstrated that the shapelet-based nearest neighbor method can provide an accurate, interpretable, and much faster classification decision for UTSC. In order to make classification more efficient, it is required to pay more attention to the sequence patches, as depicted in Fig. 1 (to simplify the plot, the univariate time series instead of multivariate time series). Unfortunately, Swin Transformer perfectly addresses the global long-range dependencies, but it is incapable of capturing the local key sequence patches.

To this end, we propose a novel attention mechanism called “Squeeze-Excitation Window Attention” (SEWA). The goal is to capture the local key sequence patches by explicitly establishing the interdependencies between windows to tackle the above issues. SEWA provides the local feature information by a two-step process: squeeze and excitation operations. The squeeze operation aggregates feature-windows to produce a window descriptor. The excitation operation utilizes MultiLayer Perceptron (MLP) to obtain descriptors with contextual window information and rescale origin feature-windows. SEWA grants different weights to the different windows by the window-window relationships.

The second problem is that, the window size of Swin Transformer will affect the results. Increasing the window size will cause an increase calculation complexity of the network [23]. To break the limitation of Swin Transformer, we design an extension of self-attention mechanism: Sparse Self-Attention within Windows (SSAW). More specifically, SSAW measures the distribution of dot-product scores by Kullback–Leibler divergence. The activated dot-product scores (having higher

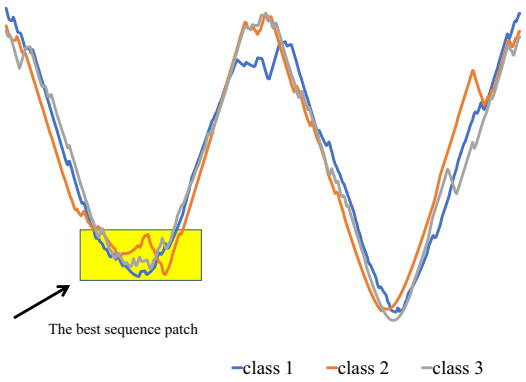


Fig. 1. The sequences in the window are the distinguishing window of classification.

weights) are selected, thus reducing the interference of redundant information. Besides, all query patches use the time-block tokens, concatenating non-overlap neighbor timestamps as the basic elements of processing within a window, instead of the timestamp tokens, using a timestamp as the basic elements. After that, the calculation complexity only changes slightly even if we increase the window size.

In this paper, we design a dual-attention network (DA-Net) for MTSC, as illustrated in Fig. 2, where the dual-attention block consists of our two proposed attention mechanisms: SEWA and SSAW. On the one hand, DA-Net utilizes the SEWA layer to discover the local features by the window-window relationships and dynamically weighting the windows. Stacking the SEWA layer allows the network to perform feature recalibration. On the other hand, DA-Net combines the SSAW layer with a larger window size to achieve high performance while reducing calculation complexity for global long-range dependencies.

The principal contributions of this paper are outlined as below:

- We develop a novel dual-attention network (called DA-Net) to exploits local distinguishing features and the global long-range dependencies. To our best knowledge, it is the first work to interactively leverage them for MTSC.
- We propose two novel attention mechanisms, i.e., Squeeze-Excitation Window Attention (SEWA) and Sparse Self-Attention within Windows (SSAW). SEWA discovers the local distinguishing features by mapping contextual feature-windows to current window partitioning. Additionally, SSAW mines the global long-range dependencies by reducing the computation complexity of self-attention within the windows, thereby expanding the window size.
- The qualitative and quantitative experiments demonstrate that DA-Net can offer the effectiveness of classification ability on the popular public datasets (i.e., UEA datasets [24]). Especially, DA-Net yields $+11.6\%$ accuracy in PEMS-SF dataset (one of UEA datasets) over current state-of-the-art algorithms.

The remainder of this paper is structured as follows. Section 2 covers the related work about many cutting-edge methodologies; Section 3 presents task description and the proposed methodology details; Section 4 lists the datasets for the experiments and thoroughly verifies the proposed method and state-of-the-art performance with the experimental data. Finally, Section 5 concludes this paper and outlines the future lines of work.

2. Related work

We briefly describe some MTSC studies related to our work, including traditional machine learning-based and deep learning-based methods.

2.1. Machine learning-based methods

MTSC is a traditional topic in the machine learning community. Numerous approaches have been proposed to handle this problem. We summarize four levels of conventional methods as follows:

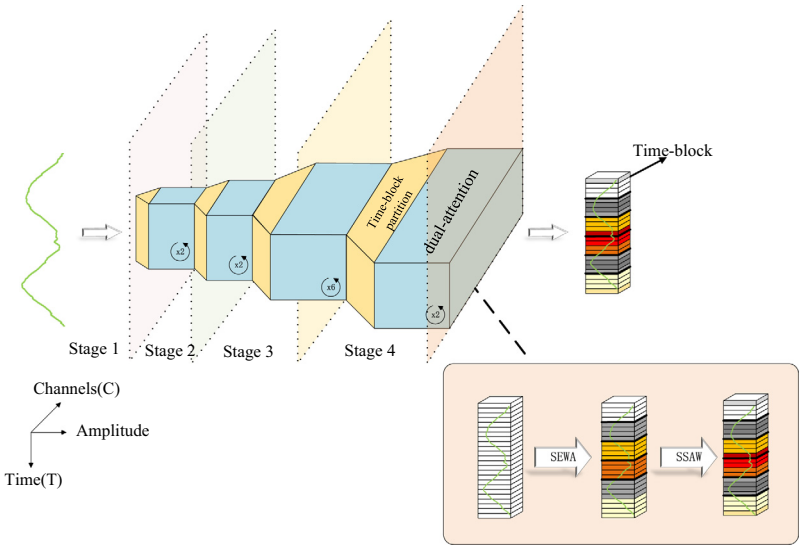


Fig. 2. The overall framework of DA-Net. Different colors represent different levels of attention, and red areas represent being given high weighting.

Distance-based methods. This category employs two typical algorithms, one based on Euclidean Distance (ED) and another based on Dynamic Time Warping [10] (DTW), grounded on distance similarity. Distance-Based similarity is typically coupled with the One-Nearest Neighbor (1-NN) in a class-agnostic manner. The experimental results indicate that 1-NN associated with distance similarity methods is the best measure for evaluation on small datasets. Not merely that, the wealth of experiments have shown that the simple nearest neighbor classification is challenging to beat [12].

Shapelet-based methods. Shapelets [6] are subsequences of time series, which represent the features of time series with segments. gRSF [21] and LTS [22] are designed to go for the best candidate shapelet. S2LFS [25] attempts to represent different classes with different discriminatory feature subsets. These methods help us recognize which features we should focus on when it comes to some specific application domains.

Dictionary-based methods. Dictionary-based methods, e.g., Boss [11] and WEASEL + MUSE [26], represent the global or local sequence patterns with the symbolic features(unigrams, bigrams and dimension identification). Each subsequence is adopted by SFA [27] algorithm to obtain these features. Following a feature selection with chi-square test, the features are served as the input of logistic regression to classify the time series.

Interval-based methods. These methods infer potential features on intervals of time series. For example, TSF [12] and RISE [28] algorithms classify the potential variation information embedded in the intervals in the time series. TSF uses the mean, variance, and slope of each instance as the initial features to train a random forest classifier.

2.2. Deep learning-based methods

Traditional machine learning-based methods are effective, but they require more handcrafted features. Deep learning, capitalizing on task-oriented, has achieved great success in recent years. Its biggest advantage is that deep learning can extract features with the help of neural networks. Among these methods of neural networks, the most representative approach is the CNNs.

CNNs have shown good results in time series classification due to their powerful local feature capture capabilities. For instance, OS-CNN [29] adopts 1D-CNN to discover the local features of time series; InceptionTime [30] uses an inception structure to cover the multi-scale representation of time series; RTFN [31] focuses on relationships between local features to discover the intrinsic time series from different data positions. These methods aim to find the local patterns by high-dimensional nonlinear feature extraction. However, the focus of time series is the long-range dependencies, which are often not considered in these approaches.

Several studies on the long-range dependency information have been proposed. Tscaps [32] proposes a multi-process collaborative architecture that uses different scaled capsule routings to discover the temporal relationships. Att-dGRU-SE-FCN [33] comprises attention mechanism, dGRU, SE block, and FCN to model the long-range dependencies. TCN [34] applies the dilated convolution and residual connection to broaden the receptive field. Besides this, researchers investigate a new flood of artificial intelligence in addition to convolutional-based neural networks – Transformer. Transformer-like applications, e.g., BERT and GPT2 [35], have gained prominence through their remarkable results in the Nature Language Processing (NLP) due to the effective self-attention mechanism. Some scholars have applied the Transformer to motor imagery EEG classification [36] and raw optical satellite time series classification [37]. Zerveas applied a Transformer-based framework [19] in unsupervised time series representation learning. Shankaranarayana used 1D convolutions augmented with Transformer [38] to enhance the representation of global information. All of them achieve decent performance.

Recently published findings have shown that modeling the local–global features is essential. For example, CTNet [39] extracts the global spatial contexts by SCM and the local channel contexts by CCM. LMP [40] extracts the short-term propagation motion and context of the target with the local modules, and explores the long-term feature correlations with the global modules. MLSTM-FCN [41] proposes a dual-network, which uses CNN to capture the local features and LSTM with attention to capture the global features.

Although existing methods perform well, they fail to handle the long sequence with the local discriminate sub-sequences adequately. To address this issue, we propose dual attention to discover the local patterns and the global information and for MTSC.

3. Proposed methodology

In this section, we first revisit the definition of MTSC. Then, we present a comprehensive illustration of DA-Net, from the network structure to the workflow of all proposed mechanisms.

3.1. Problem description

Preset a set of multivariate time series $X = \{X_1, X_2, \cdots, X_N\}$of Nreal-valued instances, where multivariate time series $X_i$has T timestamps $t \in [1, T]$and C dimensions $c \in [1, C]$ (the multivariate time series data are referred to as situations with c exceeding 1; the univariate time series data are referred to as situations with c equal to 1), can be presented as a matrix, like

$$X_i = \begin{pmatrix} x_{(1,1|i)} & x_{(2,1|i)} & \cdots & x_{(T,1|i)} \\ x_{(1,2|i)} & x_{(2,2|i)} & \cdots & x_{(T,2|i)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{(1,C|i)} & x_{(2,C|i)} & \cdots & x_{(T,C|i)} \end{pmatrix}$$

We define a mapping $y = f^*(x;\theta)$, where the objective of classification model is to approximate function f^*, by learning the nonlinear embedding parameters θ of model.

3.2. Hierarchical structure

The overall framework of DA-Net is shown in Fig. 2. There are four time-block partition layers and dual-attention blocks within the framework. Time-block partition layers aim to build a hierarchical structure to reduce the time series length. Dual-attention blocks aim to learn local–global features by establishing the window-window relationships and applying Sparse-attention.

The time-block partition layer is executed at the beginning of each stage. Here we denote the time series of the first stage as $X \in \mathbb{R}^{T\times C}$. The time-block partition layer first concatenates 4 non-overlap neighbor timestamps as a time-block, which can be considered a token in NLP, thus obtaining $\frac{T}{4}$ time-blocks. Then each time-block is flattened and projected to a $4C$-dimensional embedding. Finally, the time series data $X \in \mathbb{R}^{\frac{T}{4}\times 4C}$ are fed into the dual-attention block, where the dimensions of time series keep the same after the block. In consecutive stages 2, 3 and 4, the process of stage 1 is repeated, with the output of time series data $\frac{T}{16}\times 16C, \frac{T}{64}\times 64C$ and $\frac{T}{256}\times 256C$, respectively. The feature transformation is shown in Fig. 3.

The dual-attention block can be split into 2 consecutive modules. Fig. 4 shows an illustration of a series of essential layers inside each module. As shown in Fig. 4(a), the first module consists of a SEWA layer, a SSAW layer, a Layer Normalization (LN) layer, and a MLP layer. The only difference between the first and second modules (see Fig. 4(b)) is the introduction of shifted window layer [20], which shifts time-blocks within the window to resolve the problem that long-time dependencies are restricted to local window partitioning.

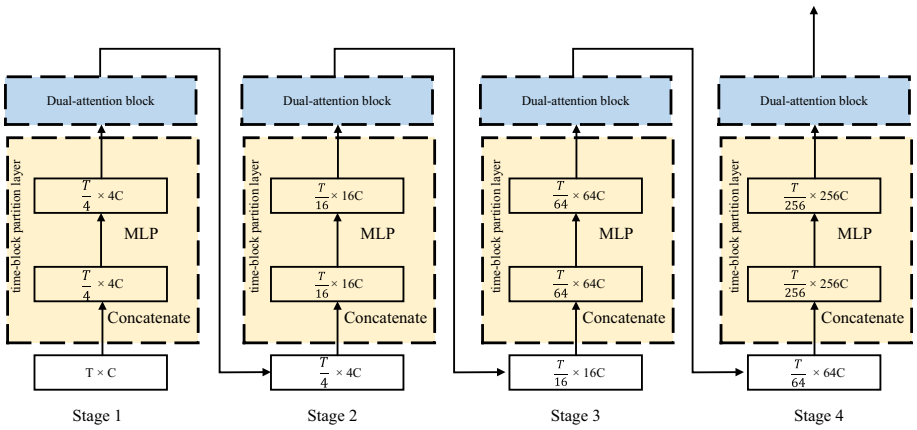
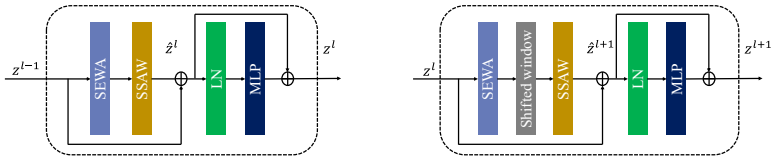


Fig. 3. The feature transformation of DA-Net. The length of the input data for each stage is reduced by a factor of 4.



(a) The first module of dual-attention block (b) The second module of dual-attention block

Fig. 4. A whole dual-attention block involves two components, where $\hat{z}^l$ and z^l denote the output features of SSAW layer and MLP layer for block l, respectively.

3.3. SEWA layer

Lack of local attention to the sequence patches can lead to a failure in the associated information extraction [42]. Traditional shapelet-based methods not only enhance the interpretation of sequences, but also drop irrelevant sub-sequences. However, these local attention lack the guidance of the global information. The SEWA layer, inspired by the success of SeNet [33], is applied to the Transformer that enables the network to focus on the local distinguishing features with the aid of global window-features. The SEWA layer divides features by windows into $X \in \mathbb{R}^{M\times C\times W}$, where M stands for the number of windows, C stands for the number of channels, and W stands for the number of the time block within a window. The SEWA layer is a computational unit building on a feature transformation mapping an input $X \in \mathbb{R}^{M\times C\times W}$ to feature $S \in \mathbb{R}^{M\prime\times C\prime\times W\prime}$. The two-step process of SEWA layer contains squeeze and excitation operations, as shown in Fig. 5.

Squeeze: This step aims to get information between windows based on long-range dependency relationships. We average all time-blocks of the windows to obtain the weights of each window. More specifically, we use global average pooling to aggregate window information and generate a window-wise contextual descriptor for each window. The squeeze operation is defined as follows:

$$Z = F_{sq}(X) = \frac{1}{C\times W}\sum_{c}^{C}\sum_{w}^{W}X(c,w). \tag{1}$$

Excitation: Although it is possible to act the window-wise descriptors directly on the windows, current descriptors do not have contextual window information. They cannot establish local distinguishing features with global information. To address this issue, we follow squeeze operation with two linear projections to learn the weight of contextual window information:

$$H = F_{ex}(Z) = W_2\mathrm{ReLU}(W_1Z), \tag{2}$$

where W_1 and W_2 stand for the learning parameters of linear projections.

Finally, the SEWA layer applies the window-wise contextual descriptors to the original features and thus suppressing non-significant windows and amplifying significant ones. We use the sigmoid function to scale the weights varying from 0 to 1. The normalized weights are weighted to the original features by a final multiplication operation on each window. The equation can be defined as follows:

$$S = F_{scale}(H,X) = X\mathrm{sigmod}(H). \tag{3}$$

It is capable of implicitly embedding the high-level features of multivariate time series into the windows, as the SEWA layer sums the values from time-blocks and channels.

3.4. SSAW layer

To capture the global long-range dependencies, Multi-head attention based on the window (W-MHA) [20] traversals all the queries in a window for calculating each dot-product pair with the complexity of $O\left(M^2\right)$. By increasing the window size

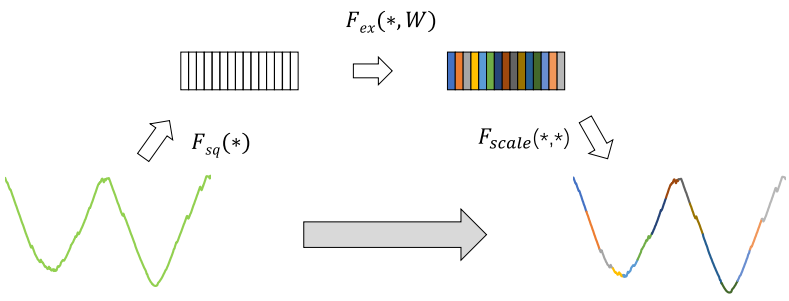


Fig. 5. Each window is squashed by $F_{sq}(*)$ into a descriptor. $F_{ex}(*,W)$ embeds global window information to obtain the window-wise contextual descriptors. Finally, $F_{scale}(*,*)$ assigns weights to origin features. Different colors represent different window attention.

M, the receptive view of the window is expanded. The widen window can enhance the representation of sequences and improve the network classification results (see more discussions in Section 4.5.1). However, W-MHA is subject to the following limitations:

(1) M cannot be increased indefinitely as the calculation is not sparse. Memory will limit its computational resources.
(2) The dominant dot-product pairs follow the long-tail distribution (see more discussions in Section 4.4.2). The majority of scores are inhibited status. Only a very small percentage of scores play a positive role when dot-product operations are performed on values [43].

We propose the SSAW layer to reduce the computation complexity and thus expand the window size. SSAW selects top-u by Kullback–Leibler divergence followed by self-attention calculation. SSAW can handle longer series than W-MHA.

Similar to W-MHA, SSAW accepts queries $Q \in \mathbb{R}^{M_Q \times C}$, keys $K \in \mathbb{R}^{M_K \times C}$, and values $V \in \mathbb{R}^{M_V \times C}$ as inputs, where M_Q, M_k and M_V stand for the window size. To be specific, the difference between W-MHA and SSAW is that SSAW reduces the scale of keys and takes top-u dominant queries (see Fig. 6), thus reducing the computation of scores. We denote i-th row in Q, K, V by q_i, k_i, v_i, respectively. Following the formulation and lemma in [43], we calculate the max-mean measurement $\overline{M}$ of the i-th query according to the following formula:

$$\overline{M}(q_i, K) = \max_j \left\{ \frac{q_i k_j^T}{\sqrt{d}} \right\} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{q_i k_j^T}{\sqrt{d}}. \tag{4}$$

The formulation consists of the max-operator and mean-operator. The max-operator implies that i-th query has a notable variance on keys, while the mean-operator suggests that i-th query is closer to the uniform distribution on keys. Subtraction of the two operations aims to prioritize the top-u dominant queries $\overline{Q}$. Here we define the top-u dominant queries $\overline{Q}$ as follows:

$$\overline{Q} = top_u \overline{M}. \tag{5}$$

Finally, we concat the $\overline{Q}$ and the mean scores of V to obtain self-attention feature map. If the i-th query is selected, the corresponding score is calculated. Otherwise, the mean value of V is used instead of the calculation of the score value, which will greatly reduce the computational effort. The output of feature map S is shown below:

$$S = \begin{cases} \text{Softmax} \left(\frac{\overline{Q} K^T}{\sqrt{d}} \right) \cdot V & \text{if } i-\text{th query is top} - u \text{ queries} \\ \text{Mean} \ (V) & \text{if } i-\text{th query is not top} - u \text{ queries} \end{cases} \tag{6}$$

To understand the algorithm better, we present the pseudo-code as shown in Algorithm 1.

Algorithm 1: Sparse Self-Attention within Windows (SSAW)

Require: Tensor $Q \in \mathbb{R}^{M_Q \times C}$, $K \in \mathbb{R}^{M_K \times C}$, $V \in \mathbb{R}^{M_V \times C}$, $u = M_V \ln M_Q$, $U = M_Q \ln M_K$; **Ensure:** Self-attention feature map S;
1: Randomly select U keys as $\overline{K}$;
2: Calculate measurement $\overline{M}$ using $\overline{K}$ by formulation (4);
3: Select top-u $\overline{Q}$ using $\overline{M}$ by formulation (5);
4: Calculate self-attention feature map S using $\overline{Q}$ by formulation (6).

In addition, for theoretically demonstrate the advantage of SSAW, we illustrate the calculation complexity of MHA, W-MHA and SSAW in Section 4.4.3.

4. Experiment

This section illustrates its modularity and flexibility by applying the proposed DA-Net for MTSC. Utilizing a series of well-established benchmarks, we conduct classification experiments to verify the performance of DA-Net. The source code and data of DA-Net are freely available athttps://github.com/Sample-design-alt/DANet.

4.1. Experiment settings

In this section, we illustrate the details of the experiment, the datasets, and the evaluation metrics.

4.1.1. Implementation of experiment

Our network is implemented in Python 3.8, using Pytorch 1.8 on the two Nvidia 3090 GPUs with 24 GB. We build our network with these hyper-parameters: batch size B = 16, window size M = 64, the channel number of hidden layer

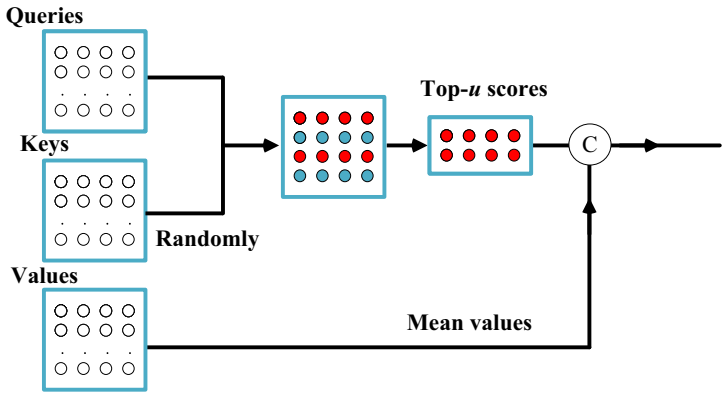


Fig. 6. Sketch of SSAW. The randomly selected keys are multiplied by queries. We select top-u (in this illustrative example, we set $u = 2$ to clarify our methodology) activate scores. Red indicate a huge difference between the query line and the keys, and green indicates no significant difference. Finally we concatenate the activate and mean scores as output. C: concatenation operation.

C = 96, multi-head numbers={3, 6, 12, 6}, and layer numbers={2, 2, 6, 2}. Additionally, the optimizer is performed using ADAM with parameters α=1e-3, β_1=0.9, β_2=0.999, and ϵ=1e-8. Minimization is performed using cross entropy loss function. We use the same strategies in all conducted experiments: optimizer, data augmentation and train/test datasets. All results are obtained with 100 iterations.

4.1.2. Dataset details

We collect benchmark datasets [24] for MTSC to test our methodology. This archive is a trending dataset collaboratively collected between UCR and UEA today. The numerous repository, which contains 30 datasets, is grouped by the application areas: Human Activity Recognition (HAR), Motion, ECG, EEG/MEG, Audio Spectra (AS), and Other. The dimensions of the tested MTSC range from 2 in trajectory to 963 in the traffic flow classification task. The non–homogeneous UEA datasets are packed with different time series lengths from 8 to 17,984 and large sizes from 27 to 50,000. Table 1 shows all the information from the UEA collection. We use the raw datasets to compare with other methodologies to evaluate performance fairly.

4.1.3. Evaluation Metrics

In this paper, we evaluate the proposed methodologies using "AVG acc", "Win", "AVG rank", and "MPCE" indexes [44]. Note that "AVG acc" and "Win" indexes denote the average accuracy of methodology and the number of wins on multiple datasets, respectively. "AVG rank" index is defined to measure the corresponding and best methodology difference. "MPCE" index is a valid method that calculates the mean error rates by taking the factor of category amount into account. The equation is defined as follows:

$$MPCE = \frac{1}{K}\sum_{k=1}^{K}\frac{e_k}{D_k}, \tag{7}$$

where K is the number of datasets, D_k represents the number of categories in the k-th dataset, and e_k represents the error rate on the k-th dataset. Smaller "AVG rank" and "MPCE" values indicate better performance of the methodology. Finally, we utilize statistical methods to evaluate the significant difference. We use the nemenyi test [45] to measure the difference between multiple approaches on multiple datasets compared with the state-of-the-art methods. In addition, we use the wilcoxon sign-rank test [45], which measures the difference between the two approaches, for the ablation experiments.

4.2. Performance

To evaluate the performance of DA-Net, we have competitive with some current well-performing competitors, e.g., ED-1NN [10], DTW-1NN [10], variants of ED-1NN as well as DTW-1NN [10], MLSTM-FCN [26], WEASEL + MUSE [16], TapNet [13], MR-PETSC [46], and SMATE [47]. Note that, we select the 14 datasets from the paper [13] for fair comparability with other published results. We perform DA-Net 10 times for each dataset and return the average accuracy as the evaluation. Table 2 shows the accuracy of all the methodologies mentioned above. The results "N/A" in the table denote that the corresponding methodology fails to execute the results.

Table 1
30 UEA datasets: An overview of the publicly best available MTSC datasets at present.

Dataset	Character						
	Abbreviation	Type	Train	Test	Dimensions	Length	Classes
ArticularyWordRecognition	AWR	Motion	275	300	9	144	25
AtrialFibrillation	AF	ECG	15	15	2	640	3
BasicMotions	BM	HAR	40	40	6	100	4
CharacterTrajectories	CT	Motion	1422	1436	3	182	20
Cricket	CR	HAR	108	72	6	1197	12
DuckDuckGeese	DDG	AS	50	50	1345	270	5
EigenWorms	EW	Motion	128	131	6	17984	5
Epilepsy	EP	HAR	137	138	3	206	4
EthanolConcentration	EC	HAR	261	263	3	1751	4
ERing	ER	Other	30	270	4	65	6
FaceDetection	FD	EEG/MEG	5890	3524	144	62	2
FingerMovements	FM	EEG/MEG	316	100	28	50	2
HandMovementDirection	HMD	EEG/MEG	160	74	10	400	4
Handwriting	HW	HAR	150	850	3	152	26
Heartbeat	HB	AS	204	205	61	405	2
InsectWingbeat	IW	AS	30000	20000	200	30	10
JapaneseVowels	JV	AS	270	370	12	29	9
Libras	LIB	HAR	180	180	2	45	15
LSST	LSST	Other	2459	2466	6	36	14
MotorImagery	MI	EEG/MEG	278	100	64	3000	2
NATOPS	NA	HAR	180	180	24	51	6
PenDigits	PD	Motion	7494	3498	2	8	10
PEMS-SF	PEMS	Other	267	173	963	144	7
Phoneme	PM	AS	3315	3353	11	217	39
RacketSports	RS	HAR	151	152	6	30	4
SelfRegulationSCP1	SRS1	EEG/MEG	268	293	6	896	2
SelfRegulationSCP2	SRS2	EEG/MEG	200	180	7	1152	2
SpokenArabicDigits	SAD	AS	6599	2199	13	93	10
StandWalkJump	SWJ	ECG	12	15	4	2500	3
UWaveGestureLibrary	UW	HAR	120	320	3	315	8

Compared to numerous other previously published results, DA-Net results are superior. According to the results offered in table, we intuitively observe that DA-Net achieved 8 wins on 14 datasets, and the average accuracy (AVG acc: 0.724) is just 0.4% less than the first place TapNet (AVG acc: 0.728). DA-Net obtains a second-best place in "AVG rank" and the best place in "MPCE". More specifically, our benchmark outperforms the state-of-the-art methodology by more than 10.6% on PEMS and 8.9% on FD datasets. The experimental results show that the application of DA-net in MTSC is practical, which provides a new framework among the convolution-free networks, making it possible to surpass the convolution networks significantly.

To statistical evaluate the difference between DA-Net and multiple methods with multiple datasets, we perform the post hoc test nemenyi based on their "AVG rank" on diverse datasets. We display the critical difference (CD) diagram so that we can better illustrate the difference, as done in [48]. The CD diagram arranges 12 multivariate time series classifiers on a horizontal line in ascending order according to their "AVG rank", as shown in Fig. 7 with a confidence level equal to 0.95. According to the post hoc test nemenyi, DA-Net is statistically significantly different from ED-1NN, ED-1NN (norm), DTW-1NN-I, DTW-1NN-I (norm), MLSTM-FCN, and DTW-1NN-D under statistical significance.

4.3. Ablation experiments

The above experiments show that DA-Net is feasible for MTSC. Furthermore, we ablate the designed layers of DA-Net to evaluate the effect of proposed layers, i.e., SEWA and SSAW. The ablation experiments are conducted on the 30 UEA datasets without settings changed. Table 3 shows the three comparisons of the ablation experiments. The ablation experiments are more interested in the significant differences between the two methodologies, so we use the pairwise wilcoxon signed-rank test, instead of nemenyi test, to measure the statistically significant differences between them. The smaller the statistic *P*-values, the more significant the difference between two methods. *P*-values are reported in the Table 3 bottom.

Our without SEWA: In this study, we investigate the effect of SEWA layer on the results. Fig. 7 and Table 3 experimentally demonstrate that the effect of SEWA layer is significantly improved. The network with the SEWA layer presents an average accuracy increase of 1.5% compared to the network without the SEWA layer, achieving 21 wins on 30 datasets. This result indicates that allocating different weights based on the critical relevance of window partition is essential. This encourages window attention to be operational throughout the network.

Our without SSAW: We further explore the effect of the SSAW layer. In this study, we employ W-MHA (without SSAW) and DA-Net as a comparison experiment to offset the effect of the SSAW layer on the results. Fig. 7 and Table 3 show the

Table 2
Performances comparison with various methodology in each dataset, the optimal performance is marked in bold.

Dataset	Methodology					
	ED-1NN	DTW-1NN-D	ED-1NN (norm)	MLSTM-FCN	DTW-1NN-I	DTW-1NN-I (norm)
AWR	0.970	**0.987**	0.970	0.973	0.980	0.980
AF	**0.267**	0.200	0.247	**0.267**	**0.267**	**0.267**
BM	0.675	0.975	0.676	0.950	**1.000**	**1.000**
CT	0.964	**0.990**	0.964	0.985	0.969	0.969
FD	0.519	0.529	0.519	**0.545**	0.513	0.500
HMD	0.279	0.231	0.278	**0.365**	0.306	0.303
HB	0.620	**0.717**	0.619	0.663	0.659	0.658
MI	**0.510**	0.500	**0.510**	**0.510**	0.390	N/A
NATO	0.860	0.883	0.850	**0.889**	0.850	0.850
PEMS	0.705	0.711	0.705	0.699	**0.734**	**0.734**
PD	0.973	0.977	0.973	**0.978**	0.939	0.939
SRS2	0.483	**0.539**	0.483	0.472	0.533	0.533
SAD	0.967	0.963	0.967	**0.990**	0.960	0.959
SWJ	0.200	0.200	0.200	0.067	**0.333**	**0.333**
AVG acc	0.642(±0.278)	0.672(±0.307)	0.64(±0.280)	0.668(±0.306)	0.674(±0.280)	0.645(±0.326)
Win	0	0	0	0	1	1
AVG rank	8.667	6.400	9.133	6.667	7.267	8.000
MPCE	1.780	1.657	1.789	1.698	1.668	1.870
Dataset	**Methodology**					
	DTW-1NN-D(norm)	WEASAL + MUSE	TapNet	MR-PETSC	SMATE	DA-Net
AWR	0.987	0.990	0.987	**0.997**	0.993	0.980
AF	0.220	0.333	0.333	0.400	0.133	**0.414**
BM	0.975	**1.000**	**1.000**	**1.000**	**1.000**	0.925
CT	0.989	0.990	0.997	0.984	0.987	**0.998**
FD	0.529	0.545	0.556	0.574	0.563	**0.645**
HMD	0.231	0.365	0.378	0.365	**0.527**	0.347
HB	0.717	0.727	**0.751**	0.702	0.727	0.626
MI	0.500	0.500	**0.590**	0.490	**0.590**	0.550
NATO	0.883	0.870	**0.939**	0.917	0.883	0.877
PEMS	0.711	N/A	0.751	0.861	0.763	**0.867**
PD	0.977	0.948	0.980	0.905	0.980	**0.989**
SRS2	0.539	0.460	0.550	0.533	0.556	**0.561**
SAD	0.963	0.982	0.983	0.960	0.982	**0.990**
SWJ	0.200	0.333	**0.400**	**0.400**	0.200	0.400
AVG acc	0.673±(0.305)	0.646(±0.326)	**0.728(±0.256)**	0.721(±0.250)	0.706(±0.292)	0.724(±0.244)
Win	0	1	5	3	2	**8**
AVG rank	4.600	4.000	**2.133**	3.400	2.867	2.733
MPCE	1.650	1.659	1.404	1.457	1.514	**1.391**

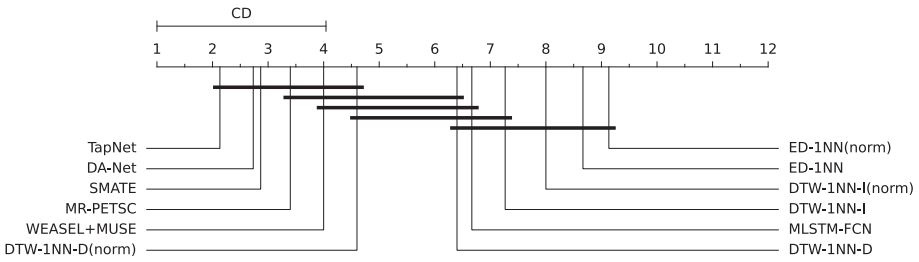


Fig. 7. CD diagram plot of the 12 multivariate time series classifiers on the 14 UEA datasets with confidence equal to 0.95. A thick horizontal line indicates that there is no significant difference in a set of methodologies.

pairwise performance. DA-Net achieves 23 wins and has an average accuracy increase of 3.4% compared with the network without SSAW. It is clear that DA-Net produces a tiny improvement in classification accuracy. The success of DA-Net is considered to be the ability of Sparse-attention.

Baseline: To perceptive the effect of the proposed methodology, we compare it with Swin Transformer as the baseline, which dropout the SEWA and SSAW layers. Table 3 shows that DA-Net yields +7.4% accuracy on the results concerning Swin

Table 3
Ablation study of SEWA and SSAW on the 30 UEA datasets.

Dataset	Baseline (without SSAW and SEWA)	Our without SSAW	Our without SEWA	Our
AWR	0.680	0.933	0.910	**0.980**
AF	0.333	0.200	**0.600**	0.467
BM	0.925	**0.950**	**0.950**	0.925
CT	0.970	0.977	0.976	**0.998**
CR	0.667	0.833	0.819	**0.861**
DDG	0.480	0.400	0.460	**0.520**
EW	0.450	0.450	0.481	**0.489**
EP	0.775	0.804	0.819	**0.833**
EC	0.278	0.693	0.744	**0.874**
ER	0.312	0.316	0.331	**0.338**
FD	**0.675**	0.656	0.639	0.648
FM	0.510	**0.570**	0.510	0.510
HMD	0.324	0.336	**0.378**	0.365
HW	0.112	0.074	**0.175**	0.159
HB	0.649	0.639	**0.659**	0.624
IW	0.545	**0.567**	0.508	**0.567**
JV	0.878	0.900	**0.941**	0.938
LIB	0.722	0.639	0.778	**0.800**
LSST	0.522	0.505	0.539	**0.560**
MI	0.500	**0.550**	0.490	0.500
NA	0.872	0.833	0.856	**0.878**
PD	0.980	**0.981**	0.964	0.980
PEMS	**0.867**	0.806	0.847	**0.867**
PM	**0.093**	0.081	0.073	**0.093**
RS	0.618	0.724	0.763	**0.803**
SRS1	0.874	0.901	0.829	**0.924**
SRS2	**0.561**	0.478	0.494	**0.561**
SAD	0.956	0.961	0.951	**0.980**
SWJ	0.333	0.333	**0.400**	**0.400**
UW	0.750	0.822	0.810	**0.833**
AVG Acc	0.590	0.630	0.649	**0.664**
P-value	9.896*e*-05	4.791*e*-04	4.742*e*-03	-

Transformer. Besides this, we visualize the pairwise comparison in Fig. 8(c). As seen by the experiment, we conclude that DA-Net can significantly improve the results in each dataset — the dataset is significantly further away from $y = x$ compared to the first two ablation experiments.

4.4. Analysis DA-Net

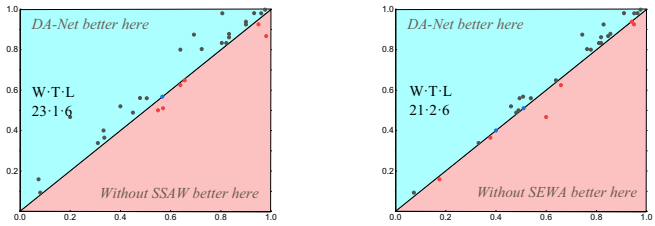
In this section, we use Grad-CAM to analyze the ability of the SEWA layer to capture the local significant sequence patch features, and visualize the self-attention map to explain why SSAW is successful using Sparse-attention. Finally, we report the calculation complexity to analyze the advantages of SSAW.

4.4.1. Analysis SEWA by Grad-CAM

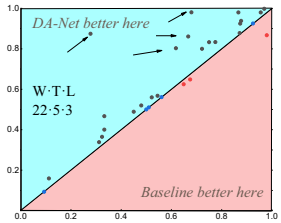
Grad-CAM [49] has been demonstrated to be an effective post hoc model-specific saliency method, which provides a faithful explanation of how much each timestamp contributes to the classification target time series. We use Grad-CAM to support the interpretation of DA-Net. Here, we extract the output of the fourth layer of DA-Net as our feature layer, and calculate the weights against the feature layer by fusing the information from forward and backward propagation. The weights are finally mapped to the input time series to obtain a heat map of the time series, as shown in Fig. 9. As observed in the figure, DA-Net captures the discriminate region of the time series (green box) and suppresses its weight in the interval 60 to 100 at time points (red box). The visualization gives a good illustration of the ability of DA-Net to capture the local–global features.

4.4.2. Analysis SSAW by self-attention feature map

We extract the 2 heads of the first window in the third layer on PEMS dataset to further analyze the proportion of weighting of self-attention feature map in the long sequence. The dataset is chosen for no particular reason. We have validated that this choice has no tangible effect on the outcomes. Fig. 10 visualizes the self-attention feature map, where each pixel represents a dot-product pair of q_i and k_i. The feature map can be classified into two chief status: active and inactive status. 1) The active status is displayed as shown in Fig. 10(a), where only a tiny portion of the pixels are at peak. Most dot-product pairs are uniformly distributed, i.e., only the minority of dot-product pairs work, and most of the dot-product pairs are in the inhibited status, which can be ignored. 2) The inactive status is presented as shown in Fig. 10(b), the value of q_i to



(a) DA-Net with and without SEWA (b) DA-Net with and without SSAW



(c) DA-Net vs Baseline

Fig. 8. Pairwise comparisons of three ablation experiments. A point represents a dataset. The distance of each point from the $y = x$ line indicates the difference between the two methods in terms of performance. The closer to the top left corner, and the better the dataset performs in our methodology. Abbreviation: W-Win, T-tie and L-loss.

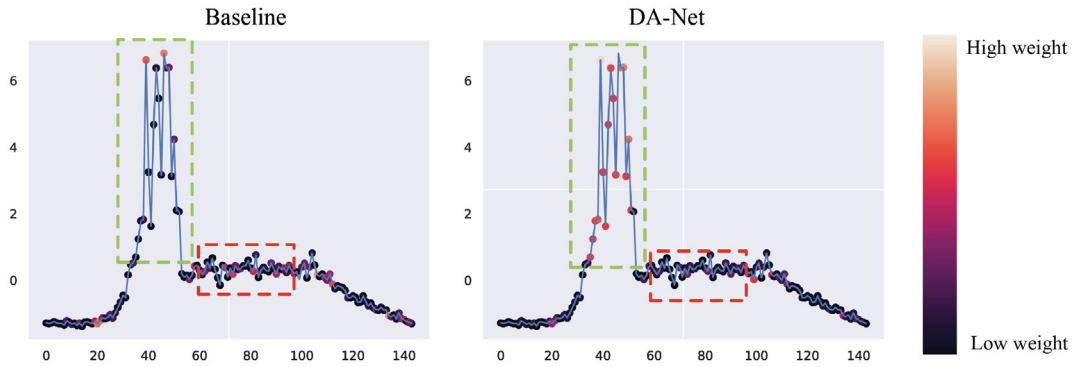
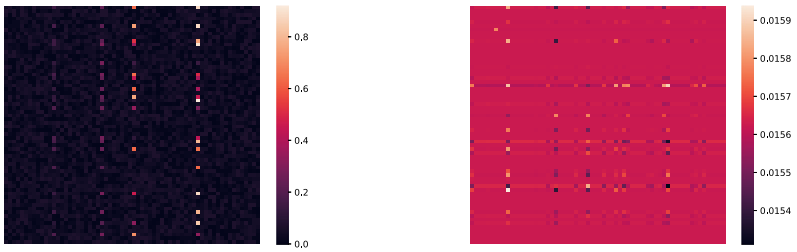


Fig. 9. The heat maps of input time series on the first channel of PEMS dataset.



(a) Acitvate status (b) Inactivate status

Fig. 10. Visualization of self-attention feature maps in {Head1, Head2}: Window1@Layer3. The picture left is Head1, the picture right is Head2.

key does not fluctuate much and is basically above and below a uniform distribution. The feature map in inactivate status fails to detect the priority of time-block. From the dot-product pair distribution, we observe that the dominant dot-product pairs follow long-tail distribution [43], so we have to retain a small number of them and save most of the calculation.

4.4.3. Analysis SSAW by calculation complexity

To illustrate the computational advantages of SSAW, we analyze the computation complexity of SSAW in relation to the traditional MHA and W-MHA. Assuming that the length of time series is L, the window size is M, and the dimension of the embedding layer is C. The following calculation ignores Softmax/Max/Mean-operator.

MHA: MHA inputs $X \in \mathbb{R}^{L\times C}$ and $w^Q, w^K, w^V \in \mathbb{R}^{C\times C}$. It requires $\Omega\left(LC^2\right)$ to obtain Q, K, and V, respectively. The complexity of obtained score and output dot-product is $\Omega\left(L^2C\right)$. Together with the feed forward layer, Conventional MHA has a complexity of $\Omega(\text{MHA}) = 4LC^2 + 2L^2C$. From the above calculation, we can conclude that global attention is not affordable for long sequences L.

W-MHA: Unlike the traditional MHA, W-MHA only happens within each window, not the entire sequence. It divides the sequence into W ($N = \frac{L}{M}$) windows and computes attention separately for each window. We feed input as $X \in \mathbb{R}^{M\times C\times W}$, and the procedure is similar to MHA. The result of the computation complexity is $\Omega(\text{W-MHA}) = N \cdot \left(4MC^2 + 2M^2C\right) = 4LC^2 + 2LMC$.

SSAW: The QKV and linear layer are computed in the same way as W-MHA. The distinction between them is that our method applies Sparse-attention instead of MHA based on the window. Sparse-attention evaluates score and output with a complexity of $\Omega(\text{SA}) = MlogMC$. The total complexity of SSAW is $\Omega(\text{SSAW}) = 4LC^2 + 2LlogMC$. It is easy to see that the calculation complexity of SSAW decreases $logM$ than W-MHA.

4.5. Further analysis of the effectiveness of DA-Net

In this section, for given PEMS dataset, we further analyze the effect of window size on results and the effect on each category in terms of encoder feature space and Receiver Operating Characteristic (ROC) curve.

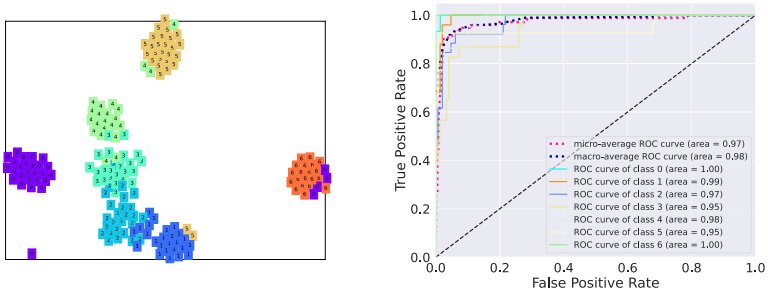
4.5.1. Analysis of window size

We show the effect of window size in Fig. 11, where four curves represent the window size $[32, 48, 64, 96]$, to investigate the impact of different window size on the results. Although there is a tiny difference in architecture, it is interesting that the impact on results is non-trivial. As the gradual increase of window size, the effect of PEMS gradually boosts. This experiment indicates that the size of the window has a non-negligible effect on the final results.

4.5.2. Encoder feature and ROC analysis for each category

Reduced-dimension representation affords the DA-Net an explanatory perspective and allows one to visualize where the advantages of model stand. For a simple sanity check of the feature quality, we reduce the high-dimensional encoder features to 2-dimensions using t-SNE [50] and label these time series feature points with a scatter plot in the form of 2-dimensional image as shown in Fig. 12(a). The figure shows the distribution of 173 test samples from 7 labels in the PEMS dataset reduced from 768-dimensional features to 2-dimensional features, with the labels color-coded. From the results, we can conclude that: 1) the distribution is varied among the different labels, and label 6 has immense variability with the remaining labels; 2) this result also implies that samples in label 6 are classified accurately with no effort.

Fig. 11. Accuracies of ablation experiments on PEMS dataset with the window size $[32, 48, 64, 96]$.



(a) Visualization of the 768-dimension encoder features in a 2-dimensional space by t-SNE.

(b) ROC curve of DA-Net in window 64. The ROC curve closer to the upper left corner represents the better performance in this class.

Fig. 12. Impact of window size and ROC curve.

Finally, we draw ROC curve and analyze each label's classification in window 64 (see Fig. 12(b)). The area under the curve for labels 0 and 6 reaches the top, consistent with our analysis of t-SNE: the distribution of labels 0 and 6 is furthest from the other labels.

5. Conclusion

This paper proposes novel the SEWA and SSAW layers, which gather the local and global features to construct a dual-attention network (DA-Net) for MTSC. The SEWA layer collects contextual window features to mine the window-window relationships, dynamically recalibrating features and learning the local features. In addition, the SSAW layer reduces the computation complexity of within windows by Sparse-attention, which makes the network generalizable to mine the global long-range dependencies. Our experiments demonstrate that DA-Net can achieve excellent results and outperform state-of-the-art approaches. Finally, we ablate individual layers to measure the performance to illustrate the effectiveness of overall structure.

DA-Net is able to achieve excellent results due to its ability to mine the potential local–global features. However, we assume that the window size is fixed when mining local discriminating features. The window size of DA-Net is a hard-parameter scheme, which fails to expand or compress adaptively. Therefore, our next step will introduce DTW to design a method that can dynamically split windows.

CRediT authorship contribution statement

Rongjun Chen: Conceptualization, Methodology, Software, Data curation, Writing - original draft, Visualization, Investigation. **Xuanhui Yan:** Validation, Supervision, Writing - review & editing. **Shiping Wang:** Validation, Writing - review & editing. **Guobao Xiao:** Software, Supervision, Validation, Writing - review & editing.

Data availability

No data was used for the research described in the article.

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.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 62072223 and 61772004, and supported by the Natural Science Foundation of Fujian Province under Grant 2020J01131199, and supported by the Guiding Foundation of Fujian Province of China No. 2020H0011.

References

[1] Z. Liang, H. Wang, Efficient class-specific shapelets learning for interpretable time series classification, Information Sciences 570 (2021) 428–450.
[2] H. Wang, Q.M. Wu, D. Wang, J. Xin, Y. Yang, K. Yu, Echo state network with a global reversible autoencoder for time series classification, Information Sciences 570 (2021) 744–768.
[3] M. Castán-Lascorz, P. Jiménez-Herrera, A. Troncoso, G. Asencio-Cortés, A new hybrid method for predicting univariate and multivariate time series based on pattern forecasting, Information Sciences 586 (2022) 611–627.
[4] M.A. Bashar, R. Nayak, Tanogan: time series anomaly detection with generative adversarial networks, in: IEEE Symposium Series on Computational Intelligence, 2020, pp. 1778–1785.
[5] I.-D. Borlea, R.-E. Precup, A.-B. Borlea, D. Iercan, A unified form of fuzzy c-means and k-means algorithms and its partitional implementation, Knowledge-Based Systems 214 (2021) 106731.
[6] L. Ye, E. Keogh, Time series shapelets: a new primitive for data mining, in: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2009, pp. 947–956.
[7] S. Qiu, H. Zhao, N. Jiang, Z. Wang, L. Liu, Y. An, H. Zhao, X. Miao, R. Liu, G. Fortino, Multi-sensor information fusion based on machine learning for real applications in human activity recognition: state-of-the-art and research challenges, Information Fusion 80 (2022) 241–265.
[8] A. Albu, R.-E. Precup, T.-A. Teban, Results and challenges of artificial neural networks used for decision-making and control in medical applications, Facta Universitatis, Series: Mechanical Engineering 17 (3) (2019) 285–308.
[9] J. Sun, Y. Yang, Y. Liu, C. Chen, W. Rao, Y. Bai, Univariate time series classification using information geometry, Pattern Recognition 95 (2019) 24–35.
[10] Y. Chen, B. Hu, E. Keogh, G.E. Batista, Dtw-d: time series semi-supervised learning from a single example, in: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2013, pp. 383–391.
[11] P. Schäfer, The boss is concerned with time series classification in the presence of noise, Data Mining and Knowledge Discovery 29 (6) (2015) 1505–1530.
[12] H. Deng, G. Runger, E. Tuv, M. Vladimir, A time series forest for classification and feature extraction, Information Sciences 239 (2013) 142–153.
[13] 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, 2020, pp. 6845–6852.
[14] S. Hashida, K. Tamura, Multi-channel mhlf: lstm-fcn using macd-histogram with multi-channel input for time series classification, in: IEEE International Workshop on Computational Intelligence and Applications, 2019, pp. 67–72.
[15] Z. Chen, Y. Liu, J. Zhu, Y. Zhang, R. Jin, X. He, J. Tao, L. Chen, Time-frequency deep metric learning for multivariate time series classification, Neurocomputing 462 (2021) 221–237.
[16] F. Karim, S. Majumdar, H. Darabi, S. Harford, Multivariate lstm-fcns for time series classification, Neural Networks 116 (2019) 237–245.
[17] G. Jin, C. Liu, Z. Xi, H. Sha, Y. Liu, J. Huang, Adaptive dual-view wavenet for urban spatial–temporal event prediction, Information Sciences 588 (2022) 315–330.
[18] Z. Feng, Y. Li, B. Sun, C. Yang, T. Huang, A multimode mechanism-guided product quality estimation approach for multi-rate industrial processes, Information Sciences 596 (2022) 489–500.
[19] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, C. Eickhoff, A transformer-based framework for multivariate time series representation learning, in: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2114–2124.
[20] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, B. Guo, Swin transformer: hierarchical vision transformer using shifted windows, in: Proceedings of the IEEE International Conference on Computer Vision, 2021, pp. 10012–10022.
[21] I. Karlsson, P. Papapetrou, H. Boström, Generalized random shapelet forests, Data Mining and Knowledge Discovery 30 (5) (2016) 1053–1085.
[22] J. Grabocka, N. Schilling, M. Wistuba, L. Schmidt-Thieme, Learning time-series shapelets, in: Proceedings of the ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2014, pp. 392–401.
[23] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, H. Hu, Video swin transformer, arXiv preprint arXiv:2106.13230 (2021) 1–12.
[24] 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 Mining and Knowledge Discovery 35 (2) (2021) 401–449.
[25] Z. Li, J. Tang, Semi-supervised local feature selection for data classification, Science China Information Sciences 64 (9) (2021) 1–12.
[26] M. Middlehurst, J. Large, M. Flynn, J. Lines, A. Bostrom, A. Bagnall, Hive-cote 2.0: a new meta ensemble for time series classification, Machine Learning 110 (11) (2021) 3211–3243.
[27] P. Schäfer, M. Högqvist, Sfa: a symbolic fourier approximation and index for similarity search in high dimensional datasets, in: Proceedings of the International Conference on Extending Database Technology, 2012, pp. 516–527.
[28] J. Lines, S. Taylor, A. Bagnall, Time series classification with hive-cote: the hierarchical vote collective of transformation-based ensembles, ACM Transactions on Knowledge Discovery from Data 12 (5) (2018) 1–34.
[29] W. Tang, G. Long, L. Liu, T. Zhou, J. Jiang, M. Blumenstein, Rethinking 1d-cnn for time series classification: a stronger baseline, arXiv preprint arXiv:2002.10061 (2020) 1–7.
[30] H. Ismail Fawaz, B. Lucas, G. Forestier, C. Pelletier, D.F. Schmidt, J. Weber, G.I. Webb, L. Idoumghar, P.-A. Muller, F. Petitjean, Inceptiontime: finding alexnet for time series classification, Data Mining and Knowledge Discovery 34 (6) (2020) 1936–1962.
[31] Z. Xiao, X. Xu, H. Xing, S. Luo, P. Dai, D. Zhan, RTFN: a robust temporal feature network for time series classification, Information Sciences 571 (2021) 65–86.
[32] Z. Xiao, X. Xu, H. Zhang, E. Szczerbicki, A new multi-process collaborative architecture for time series classification, Knowledge-Based Systems 220 (2021) 106934.
[33] M. Khan, H. Wang, A. Ngueilbaye, Attention-based deep gated fully convolutional end-to-end architectures for time series classification, Neural Processing Letters 53 (3) (2021) 1995–2028.
[34] Z. Zheng, Z. Zhang, L. Wang, X. Luo, Denoising temporal convolutional recurrent autoencoders for time series classification, Information Sciences 588 (2022) 159–173.
[35] Y. Mass, H. Roitman, Ad-hoc document retrieval using weak-supervision with bert and gpt2, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2020, pp. 4191–4197.
[36] J. Sun, J. Xie, H. Zhou, Eeg classification with transformer-based models, IEEE Global Conference on Life Sciences and Technologies (2021) 92–93.
[37] M. Rußwurm, M. Körner, Self-attention for raw optical satellite time series classification, Journal of Photogrammetry and Remote Sensing 169 (2020) 421–435.
[38] S.M. Shankaranarayana, D. Runje, Attention augmented convolutional transformer for tabular time-series, in: International Conference on Data Mining Workshops, IEEE, 2021, pp. 537–541.
[39] Z. Lian, B. Liu, J. Tao, Ctnet: conversational transformer network for emotion recognition, IEEE/ACM Transactions on Audio, Speech, and Language Processing 29 (2021) 985–1000.
[40] Z. Zhu, Z. Li, Online video object detection via local and mid-range feature propagation, in: Proceedings of the International Workshop on Human-centric Multimedia Analysis, 2020, pp. 73–82.
[41] C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, N.V. Chawla, A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 1409–1416.
[42] H. Fan, F. Zhang, R. Wang, X. Huang, Z. Li, Semi-supervised time series classification by temporal relation prediction, in: IEEE International Conference on Acoustics, Speech and Signal Proceedings, 2021, pp. 3545–3549.

[43] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, W. Zhang, Informer: beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 1–15.
[44] Y. Zhang, Y. Hou, K. OuYang, S. Zhou, Multi-scale signed recurrence plot based time series classification using inception architectural networks, Pattern Recognition 123 (2022) 108385.
[45] A. Benavoli, G. Corani, F. Mangili, Should we really use post-hoc tests based on mean-ranks, Journal of Machine Learning Research 17 (1) (2016) 152–161.
[46] L. Feremans, B. Cule, B. Goethals, PETSC: pattern-based embedding for time series classification, Data Mining and Knowledge Discovery (2022) 1–47.
[47] J. Zuo, K. Zeitouni, Y. Taher, Smate: semi-supervised spatio-temporal representation learning on multivariate time series, in: IEEE International Conference on Data Mining, 2021, pp. 1565–1570.
[48] L. Chen, D. Chen, F. Yang, J. Sun, A deep multi-task representation learning method for time series classification and retrieval, Information Sciences 555 (2021) 17–32.
[49] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 618–626.
[50] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (11) (2008) 1–27.