Received November 9, 2020, accepted November 20, 2020, date of publication November 25, 2020,
date of current version December 8, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3040515

A Novel Channel and Temporal-Wise Attention in Convolutional Networks for Multivariate Time Series Classification

XU CHENG[1,*], (Member, IEEE), PEIHUA HAN[2,*], GUOYUAN LI[2], (Senior Member, IEEE), SHENGYONG CHEN[1], (Senior Member, IEEE), AND HOUXIANG ZHANG[2], (Senior Member, IEEE)

[1]Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology, School of Computer Science and Technology, Tianjin University of Technology, Tianjin 300384, China
[2]Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, 6009 Aalesund, Norway

Corresponding author: Guoyuan Li (guoyuan.li@ntnu.no)
*Xu Cheng and Peihua Han contributed equally to this work.

This work was supported in part by the project ''Remote Control Centre for Autonomous Ship Support'' under Grant 309323 from Research Council of Norway, in part by the project ''Digital Twins for Vessel Life Cycle Service'' under Grant 280703 from Research Council of Norway, in part by the National Natural Science Foundation of China under Grant 62020106004, and in part by the opening foundation of Tianjin Key Laboratory of Intelligence Computing and Novel Software Technology.

ABSTRACT Multivariate time series classification (MTSC) is a fundamental and essential research problem in the domain of time series data mining. Recently deep neural networks emerged as an end-to-end solution for MTSC and achieve state-of-the-art results on several public datasets. It is favored by its hierarchical feature extraction ability and most of the researches focus on designing a network architecture to ensure its performance on MTSC. Despite this, there are seldom investigations on the attention mechanism in MTSC, which has been demonstrated as an effective module to extract features in other domains. In this paper, we propose a residual channel and temporal attention (CT_CAM) module, which aims to refine the feature extracted from the convolutional neural network and thus improve the classification performance. Extensive experiments on 15 public MTSC datasets show that the proposed CT_CAM module achieves competitive performance compared with nine baseline methods and three other attention modules.

INDEX TERMS Multivariate time series classification, convolutional neural network, channel attention, temporal attention.

I. INTRODUCTION

With the advance of sensor technologies, extensive data sequentially ordered by time are received and recorded in our daily life. These time series data are typically recorded by different types of sensors simultaneously over time and form the so-called multivariate time series. Extracting knowledge from multivariate time series has attracted an increasing amount of attention in recent decades. Multivariate time series classification (MTSC) is one of the most significant tasks. MTSC aims to predict classification labels for a certain multivariate time series data, which has many application scenarios in the real world such as clinical time series data analysis [1], human activity recognition [2], [3], sea state estimation [4], [5], and fault diagnosis in machinery system [6], [7].

For MTSC, a plethora of research focuses on feature-based methods that extract a set of features that can represent the time-series patterns. Then a classifier can be trained using these features. These approaches need heavy crafting on feature engineering and there might be different feature extraction schemes for different applications. Moreover, the generated huge feature space usually makes the feature selection step difficult and thus results in low accuracy [8]. Recently, deep neural networks have been utilized to provide an end-to-end solution for time series classification problems and achieve state-of-the-art results on several public datasets [9], [10]. The advantage is that it combines hierarchical feature extraction and classification and therefore it can learn the representation from data directly.

Designing deep neural network architecture is a difficult engineering task but essential because well-designed networks ensure remarkable performance improvement in

The associate editor coordinating the review of this manuscript and approving it for publication was Vicente Alarcon-Aquino.

various applications [11]. Most of the researches in MTSC focus on designing a network architecture by stacking convolutional neural network (CNN) block [12], combining long short-term memory (LSTM) with CNN [10] or adding skip-connection [9]. However, an aspect that lacks investigation in MTSC is the attention mechanism, which has been addressed extensively in natural language processing [13] and computer vision domain [11]. Attention tells where to focus and therefore improves the representation of interests. Attention mechanisms designed for CNN and applied to other domains can play a certain role in time series data, but time series data has its own unique characteristics, which might need a special treatment for the attention mechanisms. The most notable feature of time series is its temporal correlation, while in image processing and other computer vision applications, researchers pay more attention to the spatial correlation between pixels [5]. Intuitively, different sensor modalities come from different domains and they have different importance in different tasks. For example, in human activity recognition, the accelerometer features may be more significant in distinguishing the ''walking'' and ''biking'' activities while the gyroscope features may be more significant in distinguishing the ''turning-left'' and ''turning-right'' activities [14]. Besides, not all timesteps contribute equally to the task. For instance, the features on some timesteps may show a more salient pattern than the others in distinguishing the ''fault'' and ''normal'' status in the problem of fault diagnosis.

In this paper, we propose a novel attention module for MTSC. It consists of two parts: channel calibration attention module (CCAM) and temporal calibration attention module (TCAM), which aims to address importance along channel and time axis, respectively. Therefore the representation power of the network can be enhanced. The proposed attention block can be implemented in state-of-the-art network architecture by simply adding it behind the CNN block. To summarize, our work has the following main contributions: 1) a novel attention module called CT_CAM (channel and temporal calibration attention module), which effectively integrates channel and temporal attention in CNN features, is proposed for MTSC. This module is generic and therefore can be applied to any layer in any CNN architecture such as fully convolutional network (FCN) and Deep residual network (ResNet). By integrating attention layers with both CCAM and TCAM, the proposed attention module can capture spatial and temporal dependencies of the time series data, which amplifies the more important and informative modalities and timesteps during classification. 2) Extensive experiments are performed on 15 public MTSC datasets. All the results of the combination of the proposed CT_CAM and CNN, CT_CAM and FCN, CT_CAM and DenseNet outperform that of baselines. Compared with other attention mechanisms, the proposed CT_CAM achieves state-of-the-art whether it is combined with CNN, FCN and DenseNet. The ablation study demonstrates the importance of the proposed attention module.

The structure of this paper is as follows: Section II gives an introduction to MTSC and attention mechanism. Section III describes the architecture of the proposed approach. The experiment is discussed in Section IV, and the paper is summarized in Section V.

II. RELATED WORK

A. MULTIVARIATE TIME SERIES CLASSIFICATION

Most of the work for MTSC can be grouped into three categories: similarity-based methods, feature-based methods and deep-learning methods. Similarity-based methods, as the name suggests, are to identify time series by calculating the similarity (Euclidean distance or other distance metrics) between two time series. Dynamic Time Warping (DTW) has been reported to be the best competitive methods. There are two widely used version of DTW for MTSC, dependent DTW (DTWD) and independent DTW (DTWI). DTWD measures the squared Euclidean cumulated distance of all dimensions, but DTWI is to consider the cumulative distance over the multiple dimensions. Feature-based methods transform the original time series into a low latent space that is easier to classify. There are two techniques that widely used for the transformation of time series: Shapelets models and Bag-of-Words (BOW). The bag-of-features framework (TSBF) [15] extracts the local and global features for each time series and feeds them to a random forest classifier. Bag-of-SFA-Symbols (BOSS) [8] introduces a combination of a distance-based classifier and histograms with symbolic Fourier approximation. Ensemble algorithms that use multiple feature-based algorithms such as the elastic ensemble (PROP) [16] and the flat collective of transform-based ensembles (COTE) [17] also achieve promising results. Recently effort has been made to exploit the deep learning approaches to overcome the limitation of feature-based methods. A hybrid model combines FCN and LSTM is proposed by [10] with the aims of better feature extraction. A novel model, integrating with random group permutation method, LSTM and multi-layer convolutional networks for MTSC is proposed [18]. The above researches target designing a network architecture by stacking CNN and LSTM for better performance. We focus on the attention mechanism for MTSC which is less addressed by most of the existing works.

B. ATTENTION MECHANISM

Attention has been recognized as an important role in human perception [19]. Attention mechanisms have been demonstrated in sequence learning [20] and image understanding [21] for its ability to focus on the informative salient parts of a signal. Attention mechanisms have been proven as an effective way to enhance CNN. Now the developments of attention mechanism can be roughly categorized into two directions: enhancement of feature aggregation and combination of channel and spatial attention. A compact attention module called Squeeze-and-Excitation (SE) is proposed to exploit the inter-channel relationship [22]. SE is the first attempt to learn channel attention and achieves promising performance. A Convolutional Block Attention Module (CBAM), which can integrate into any CNN architectures seamlessly, is proposed [11]. CBAM can infer the attention map along the channel and spatial dimension, and then the input feature map can be refined based on the element-wise multiplication of input feature map and attention map. A second-order attention module is proposed for effective feature aggregation by learning more discriminative representations [23]. Another attention module named gather-excite (GE) is introduced to aggregate spatial features in CNN [24]. A non-local (NL) attention module is presented to utilize the local relationship for capturing long-range dependencies in the task of computer vision [25]. On the basis of the NL module, a GCNet is developed to model long-range dependency [26]. Inspired by the promising results achieved in the domain of image processing, CNN has been gradually used in many MTSC tasks. The impact of attention mechanism on CNN has not been exploited extensively in MTSC. The SE module is first integrated into a CNN model and applied to MTSC [10]. The experimental results show that with the help of the SE module, the accuracy of the model has been greatly improved. Obviously, all of the above methods are dedicated to the development of sophisticated attention modules by learning more discriminative features. Different from them, our proposed attention module aims at learning effective channel attention as well as temporal attention simultaneously.

III. CHANNEL AND TEMPORAL CALIBRATION ATTENTION MODULE

As mentioned above, CNN has become a common framework for TSC tasks. This paper mainly studies the use of attention mechanism to improve the classification ability of CNN. In other words, the proposed attention module can be applied to all kinds of variants of CNN, such as FCN, ResNet, and DenseNet. As illustrated in FIGURE 1, the original multi-layer feature maps of FCN would be enhanced through CT_CAM module, which consists of CCAM and TCAM.

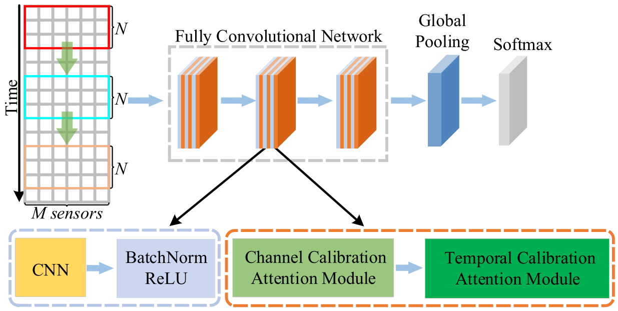
Time
N
N
N
M sensors
Fully Convolutional Network
Global Pooling
Softmax
CNN
BatchNorm ReLU
Channel Calibration Attention Module
Temporal Calibration Attention Module

FIGURE 1. Illustration of FCN with CT_CAM.

The extracted features would be processed by CT_CAM sequentially. Given an intermediate feature map at k-th layer $F_k \in \mathbb{R}^{T\times C}$, T stands for the timesteps and C is the channels of features. The CT_CAM modulates F_k using the attention weights in a recurrent and multi-layer fashion as:

$$\begin{aligned} F_k &= CNN(F_{k-1}), \\ \alpha &= \Phi(F_k), \\ F_{k+1} &= f(F_k, \alpha). \end{aligned} \tag{1}$$

where F_k is the feature map output from previous CNN layer which consists of a convolutional layer, a normalization layer, and a RELU layer. F_{k+1} and Φ are the modulated feature and the CT_CAM function, respectively, which will be detailed in Section III-A and Section III-B. $f(\cdot)$ is a weighting function that modulates CNN features and attention weights.

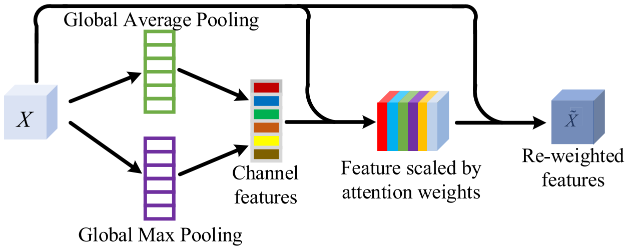

FIGURE 2. Illustration of CCAM.

A. CCAM

The whole process of CCAM is depicted in FIGURE 2. Assuming the shape of the raw multivariate time series data is $\mathbb{R}^{T\times N}$, T and N are the timesteps and dimension of time series data, respectively. Usually, 1D CNN would be utilized to extract feature from the raw time series data. The filter of 1D CNN is performed as the pattern detector, which can transform the raw time series data $\mathbb{R}^{T\times N}$ to the features $\mathbb{R}^{T\times C}$. C is the number of filters in 1D CNN, and also is the channel of feature. It is easy to know that each channel of feature represents the response activation of convolutional filter. In this paper, a channel attention module is proposed to overcome the conventional CNN treat feature channel equally. That is, employing an attention module in a channel manner can be regarded as "channel selection". In the domain of image processing, it is also called as semantic attribute selection [27].

For the raw convolutional feature $X = [x_1, x_2, \ldots, x_C]$, where $X \in \mathbb{R}^{T\times C}$, $x_k \in \mathbb{R}^{T\times 1}$ represents the k-th channel of feature map X. Then the global average pooling and global max pooling are applied to each channel to obtain the average channel feature $X_{ac} \in \mathbb{R}^{C\times 1}$ and max channel feature $X_{mc} \in \mathbb{R}^{C\times 1}$.

To calculate the attention weights, the average channel feature X_{ac} and max channel feature X_{mc} are forwarded to two multilayer perceptron (MLP) with shared weights. After the MLP, the features is transformed by sigmoid. Finally, the raw features can be calibrated by the attention map.

$$\begin{aligned} \alpha &= \sigma(\boldsymbol{W_2}(\boldsymbol{W_1}(X_{ac})) + \boldsymbol{W_2}(\boldsymbol{W_1}(X_{mc}))), \\ \boldsymbol{X_{att}} &= \alpha \otimes \boldsymbol{X}, \end{aligned} \tag{2}$$

where X is the original input, α is the weights of attention module, X_{att} is the weighted features, $\otimes$ means the element-wise multiply. $W_1 \in \mathbb{R}^{C/r \times C}$ and $W_2 \in \mathbb{R}^{C \times C/r}$ represent the weights of the first and second MLP, respectively. σ denotes the sigmoid transformation.

Inspired by the success of residual blocks [9], the channel attention is integrated with residual connection, which is called residual channel attention. From FIGURE 2, we have $\widetilde{X} = X + X_{att}$, where $\widetilde{X}$ is the weighted features, X is the original input, X_{att} is the feature scaled by attention weights.

B. TACM

The channel attention is focusing on the informative features in different channels, but the information in time axis also should be emphasized. To achieve this goal, TACM is proposed. To compute the temporal attention efficiently, Gate Recurrent Unit (GRU) is utilized, as illustrated in FIGURE 3.

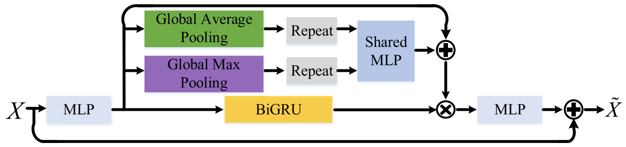

FIGURE 3. Illustration of TACM.

Suppose the feature processed by channel attention is $X \in \mathbb{R}^{T \times N}$. The features will first be mapped to $\widetilde{X} \in \mathbb{R}^{T \times K}$. As shown in FIGURE 3, the raw feature would be transformed using bidirectional Gate Recurrent Unit (BiGRU) to better capture the temporal memory information: $\mathbf{X}_{gru} = BiGRU(\widetilde{\mathbf{X}})$.

To calibrate the temporal information, the idea of channel attention and residual connection is adopted. The $\widetilde{\mathbf{X}}$ is first forwarded to average pooling and max pooling, as similar with channel attention. And then these two pooling features would be repeated along the time-axis. Finally, the attention weights can be computed as follows:

$$\begin{aligned}\alpha &= \sigma(\mathbf{Shared\_MLP}(F_{at}) + \mathbf{Shared\_MLP}(F_{mt})) + \widetilde{\mathbf{X}}, \\ &= \sigma(\mathbf{W_2}(\mathbf{W_1}(F_{at})) + \mathbf{W_2}(\mathbf{W_1}(F_{mt}))) + \widetilde{\mathbf{X}} \end{aligned} \tag{3}$$

where F_{at} and F_{mt} are the features after average pooling and max pooling, respectively. $\boldsymbol{W_1} \in \mathbb{R}^{C/r \times C}$ and $\boldsymbol{W_2} \in \mathbb{R}^{C \times C/r}$ represent the weights of the first and second shared MLP, respectively.

The refined feature can be computed as follows:

$$X_{att} = \alpha \otimes X_{gru} \tag{4}$$

where $\otimes$ means the element-wise multiply.

The final output $\mathbf{X}_T$ of temporal attention module can be computed based on the residual connection, as shown in the left branch of FIGURE 3:

$$\mathbf{X}_T = \mathbf{X} + \mathbf{MLP}(\mathbf{X}_{att}) \tag{5}$$

where **MLP** is used for the shape mapping from $\mathbb{R}^{T \times K}$ to $\mathbb{R}^{T \times N}$.

C. ARRANGEMENT OF ATTENTION MODULES

According to the different implementation order of CCAM and TACM, there are two types of models, which incorporates both two attention mechanisms. These two types are described as follows:

1) CHANNEL-TEMPORAL (CT)

The first type, denoted as Channel-Temporal (CT), applies CCAM before TACM. The flow chart of CT is represented in FIGURE 1. For the initial convolutional feature map X_r, the residual channel-wise attention Ψ_c is adopt to obtain the weights α for raw feature map. Then the weighted feature map can be obtained through the combination of X_r and α. After the channel attention, the weighted feature map is fed to the temporal attention Ψ_t and the temporal attention weights β is obtained in the same way with channel attention. The whole process can be summarized as follows:

$$\begin{aligned}\alpha &= \Psi_c(X_r), \\ \beta &= \Psi_t(f_c(X_r, \alpha)), \\ X_w &= f(X_r, \alpha, \beta, dp) \end{aligned} \tag{6}$$

where $f_c(\cdot)$ is the multiplication of feature map channels and corresponding weights. X_w is the modulated feature map, f represents the modulate function. dp is the dropout rate between the two attention modules, and dp is set to 0.3 in this paper.

2) TEMPORAL-CHANNEL (TC)

The second type is called as Temporal-Channel (TC), which implements the TACM first. For this type, given the raw feature map X_r, the TACM Ψ_t is first utilized to calculate the temporal attention weights β. The CCAM Ψ_c would employ the weighted channel feature map as the input, and the channel attention weight α can be calculated. The whole processes are summarized as follows:

$$\begin{aligned}\beta &= \Psi_t(X_r), \\ \alpha &= \Psi_c(f_t(X_r, \beta)), \\ X_w &= f(X_r, \alpha, \beta, dp) \end{aligned} \tag{7}$$

where $f_t(\cdot)$ is a element-wise multiplication for feature map time-steps and corresponding attention weights. f denotes the modulate function, and dp is the dropout rate between the two attention modules. X_w represents the weighted feature map through the two attention modules.

IV. EXPERIMENT

A. EXPERIMENTAL SETUP

1) DATASETS

We use 15 datasets from the latest MTSC archive [28]. This archive consists of real-world multivariate time series data with a wide range of cases, dimensions, and series lengths, as presented in TABLE 1. Its application mainly includes human activity recognition, motion classification, ECG/EEG signal classification, and audio spectra classification.

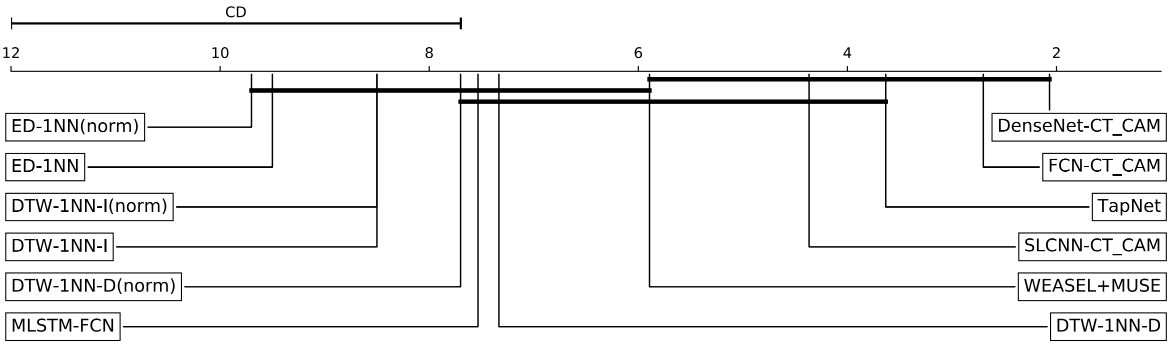

FIGURE 4. Critical difference diagram of the average ranks with the Nemenyi test ($p = 0.05$).

The number of the class ranges from 2 such as face detection to 39 in audio phoneme. The length of the time series ranges from 8 to 3,000 while the dimension ranges from 2 to 963. The size of datasets also has a range from 27 to 9,414. For each dataset, the classification accuracy is calculated as the evaluation metric. The average accuracy value, the number of Wins/Ties and the average rank are computed to compare different methods.

TABLE 1. Description of 15 MTSC datasets.

Dataset	#Train	#Test	#Variables	Length	#Classes
Articulary WordRecognition	275	300	9	144	25
AtrialFibrillation	15	15	2	640	3
BasicMotions	40	40	6	100	4
Character Trajectories	1422	1436	3	182	20
FaceDetection	5890	3524	144	62	2
HandMovement Direction	160	74	10	400	4
Heartbeat	204	205	61	405	2
MotorImagery	278	100	64	3000	2
NATOPS	180	180	24	51	6
PenDigits	7494	3498	2	8	10
PEMS-SF	267	173	963	144	7
Phoneme	3315	3353	11	217	39
SelfRegulationSCP2	200	180	7	1152	2
SpokenArabicDigits	6599	2199	13	93	10
StandWalkJump	12	15	4	2500	3

2) IMPLEMENTATION DETAILS

All the experiments are implemented on a server, which is equipped with Intel processors (64GB) and TITAN V (12GB). Pytorch is used for the implementation of the models [29]. During the whole training process, the learning rate is set to 1e-4; Adam is utilized as an optimizer [30]; For a fair comparison, the training epochs are set 3000, which is the same as [18].

B. BENCHMARK COMPARISON

We plug the CT_CAM module into the FCN [9], single layer CNN (SLCNN), and the latest proposed DenseNet [4] and then compared this DenseNet_CT_CAM, FCN_CT_CAM, and SLCNN_CT_CAM with nine different baseline approaches, including common distance-based classifiers, bag-of-patterns feature-based methods, and deep learning framework. The details of the baselines we use are provided as follows. **ED-1NN**, **ED-1NN(norm)**, **DTW-1NN-I**, **DTW-1NN-I (norm)**, **DTW-1NN-D** and **DTW-1NN-D (norm)**: One nearest neighbor classifier (1NN) with two different distance measurements, Euclidean distance (ED) and dynamic time warping (DTW). I and D denote that the DTW is computed by treating every dimension individually or together respectively. Data normalization is applied with annotation (norm) [31]. **WEASEL-MUSE** [8]: This framework builds a large feature space using multiple window lengths. Then Chi-squared test is used to identify the most relevant features and feed them to logistic regression. **MLSTM-FCN** [10]: This deep learning model consists of an LSTM layer and an FCN layer along with a SE module. **TapNet** [18]: This model is also a combination of an LSTM layer and stacked CNN layers. The random permutation was used before stacked CNN layers to reorganize the time series dimensions into different groups.

For a fair comparison, we duplicate the table shown in [18], and add the experimental results of our model, as listed in TABLE 2. The default settings are adopted for the DenseNet_CT_CAM, the number of filters for SLCNN and FCN are 128 and {128, 256, 128}. In the three cases, the number of hidden units for BiGRU is 8. However, for some large datasets, such as MotorImagery, Phoneme, the hyperparameter will be adjusted accordingly. The best accuracy for each dataset is denoted with boldface. In terms of average accuracy, our three models outperform all the baseline methods. The DenseNet_CT_CAM gets the best average accuracy of 0.746, which achieves a significant improvement compared with the existing state-of-the-art approach TapNet with the average accuracy of 0.691. In terms of the number of wins/ties, our model achieves 9 wins/ties which is the best among nine methods, while both TapNet and WEASEL+MUSE achieve 2 wins/ties. It can be observed that our model can achieve better performance in most datasets, especially in the datasets with small amounts of data such as Heartbeat and HandMovementDirection, which contains only hundreds of training samples.

FIGURE 4 shows a critical difference diagram [32] over the average ranks of the different MTSC methods.

TABLE 2. Accuracy comparison in UEA multivariate time series dataset.

Dataset	DenseNet -CT_CAM	FCN -CT_CAM	SLCNN -CT_CAM	TapNet	MLSTM -FCN	WEASEL +MUSE	ED -1NN	DTW-1NN-I	DTW-1NN-D	ED-1NN (norm)	DTW-1NN -I(norm)	DTW-1NN -D(norm)
Articulary WordRecognition	0.9867	0.9867	0.9867	0.987	0.973	**0.99**	0.97	0.98	0.987	0.97	0.98	0.987
Atrial Fibrillation	**0.4667**	**0.4667**	**0.4667**	0.333	0.267	0.333	0.267	0.267	0.2	0.267	0.267	0.22
BasicMotions	**1**	**1**	**1**	**1**	0.95	**1**	0.675	**1**	0.975	0.676	**1**	0.975
Character Trajectories	0.9937	0.991	0.8196	**0.997**	0.985	0.99	0.964	0.969	0.99	0.964	0.969	0.989
FaceDetection	**0.5692**	0.559	**0.5692**	0.556	0.545	0.545	0.519	0.513	0.529	0.519	0.5	0.529
HandMovement Direction	0.473	**0.4865**	0.4459	0.378	0.365	0.365	0.279	0.306	0.231	0.278	0.306	0.231
Heartbeat	**0.8195**	0.8098	0.7756	0.751	0.663	0.727	0.62	0.659	0.717	0.619	0.658	0.717
MotorImagery	**0.66**	0.57	0.64	0.59	0.51	0.5	0.51	0.39	0.5	0.51	N/A	0.5
NATOPS	0.9833	**0.9889**	0.8833	0.939	0.889	0.87	0.86	0.85	0.883	0.85	0.85	0.883
PEMS-SF	**0.7977**	0.7746	0.7746	0.751	0.699	N/A	0.705	0.734	0.711	0.705	0.734	0.711
PenDigits	0.9877	**0.9886**	0.936	0.98	0.978	0.948	0.973	0.939	0.977	0.973	0.939	0.977
Phoneme	**0.2088**	0.1616	0.1816	0.175	0.11	0.19	0.104	0.151	0.151	0.104	0.151	0.151
SelfRegulation SCP2	**0.5944**	**0.5944**	0.5889	0.55	0.472	0.46	0.483	0.533	0.539	0.483	0.533	0.539
SpokenArabic Digits	0.9843	**0.9923**	0.9841	0.983	0.99	0.982	0.967	0.96	0.963	0.967	0.959	0.963
StandWalkJump	**0.6667**	0.6	0.6	0.4	0.067	0.333	0.2	0.333	0.2	0.2	0.333	0.2
Avg. Value	**0.746**	0.7313	0.7101	0.691	0.631	0.66	0.606	0.639	0.637	0.606	0.656	0.638
Wins&Ties	**9**	7	3	2	0	2	0	1	0	0	1	0
Avg. Rank	**2.067**	2.7	4.367	3.633	7.533	5.893	9.5	8.5	7.333	9.7	8.5	7.7

TABLE 3. Accuracy comparison of different attention modules with SLCNN as backbone.

Datasets	SLCNN				
	N/A	CBAM	GC	SE	CT_CAM
ArticularyWordRecognition	0.6967	0.9767	0.6667	0.55	0.9867
AtrialFibrillation	0.4467	0.3333	**0.5333**	0.4667	0.4667
BasicMotions	**1**	0.975	**1**	**1**	**1**
CharacterTrajectories	0.5857	0.8078	0.5919	0.555	0.8196
FaceDetection	0.569	0.561	0.5661	0.5661	**0.5692**
HandMovementDirection	0.4324	0.473	0.5	**0.5135**	0.4459
Heartbeat	0.7707	0.7659	**0.7805**	0.7659	0.7756
MotorImagery	0.61	0.57	0.54	0.61	**0.64**
NATOPS	0.7278	0.9611	**0.9722**	0.95	0.8833
PEMS-SF	0.7688	**0.7803**	0.7341	0.7341	0.7746
PenDigits	0.8602	0.9357	**0.9731**	0.8542	0.936
Phoneme	0.0734	0.176	0.1253	0.0641	**0.1816**
SelfRegulationSCP2	0.5944	**0.6**	0.5778	0.5778	0.5889
SpokenArabicDigits	0.8386	0.9791	0.9795	0.8859	**0.9841**
StandWalkJump	0.5333	0.4	**0.6**	0.5333	**0.6**
Average value	0.6338	0.6863	0.676	0.6418	**0.7101**
Wins&Ties	1	2	**6**	2	**6**
Avg. Rank	3.5	3.167	2.633	3.733	**1.967**

TABLE 4. Accuracy comparison of different attention modules with FCN as backbone.

Datasets	FCN				
	N/A	CBAM	GC	SE	CT_CAM
ArticularyWordRecognition	**0.9867**	0.8867	0.9733	**0.9867**	**0.9867**
AtrialFibrillation	0.4	0.4	0.4	0.4	**0.4667**
BasicMotions	**1**	**1**	**1**	**1**	**1**
CharacterTrajectories	0.9916	0.9784	**0.9951**	0.9923	0.991
FaceDetection	**0.569**	0.5006	0.5636	0.559	0.559
HandMovementDirection	0.4459	0.3378	0.4459	0.473	**0.4865**
Heartbeat	**0.8098**	0.7854	0.7707	0.8	**0.8098**
MotorImagery	0.59	**0.62**	**0.62**	0.59	0.57
NATOPS	0.9833	0.9667	0.9444	0.9722	**0.9889**
PEMS-SF	**0.7803**	0.763	0.7746	0.7572	0.7746
PenDigits	0.9889	0.984	0.988	**0.99**	0.9886
Phoneme	0.0871	**0.1843**	0.1599	0.1482	0.1616
SelfRegulationSCP2	0.5778	0.5899	0.5778	0.5889	**0.5944**
SpokenArabicDigits	0.9768	0.9463	0.9814	0.985	**0.9923**
StandWalkJump	**0.6667**	0.4	0.5333	0.6	0.6
Avg. Value	0.7236	0.6896	0.7152	0.7228	**0.73134**
Wins&Ties	6	3	3	3	**8**
Avg. Rank	2.7	3.867	3.3	2.867	**2.267**

Classifiers with the lowest (best) ranks are to the right. The group of classifiers that are not significantly different in their rankings are connected by the solid horizontal lines. The critical difference (CD) length at the top represents statistically significant differences.

C. COMPARISON WITH OTHER ATTENTION MECHANISMS

To illustrate the superiority of the proposed CT_CAM module for MTSC, three different attention modules are used for comparison in the benchmark datasets. Three different backbones, SLCNN, FCN and DenseNet, are used for these modules. SLCNN contains only one CNN block with the number of filter 128 while FCN consists of three CNN blocks with the number of filter {128, 256, 128}. The setting of DenseNet is the same with [4]. The attention module is stacked after each CNN block. The details of the attention modules we used are presented as follows. **CBAM** [11]: Convolutional block attention module (CBAM) consists of a channel and spatial attention block, where both the global average and max pooling are used to generate statistics. **GC** [26]: Global context (GC) adopted 1 × 1 convolution for both attention pooling and bottleneck transform. **SE** [22]: Squeeze-and-Excitation (SE) used global average pooling to generate channel-wise statistics and used bottleneck MLP for transform. Moreover, **N/A** means no attention modules are used. The comparison results are presented in TABLE 3, TABLE 4, and TABLE 5.

From TABLE 3, TABLE 4, and TABLE 5, it is easy to see that the proposed CT_CAM outperforms other attention modules in terms of average accuracy, wins&ties and average rank in the three backbones. More specifically, the proposed CT_CAM in SLCNN shows 10.64%, 5.04%, and 3.47% improvement compared to the SE, GC, and CBAM, respectively, as depicted in Table Table 3. There are 6.05%,

TABLE 5. Accuracy comparison of different attention modules with DenseNet as backbone.

Datasets	DenseNet				
	N/A	CBAM	GC	SE	CT_CAM
ArticularyWordRecognition	**0.99**	0.9167	0.9133	0.9833	**0.99**
AtrialFibrillation	0.4	**0.4667**	**0.4667**	0.4	**0.4667**
BasicMotions	**1**	**1**	**1**	**1**	**1**
CharacterTrajectories	0.9854	0.9513	0.9645	0.993	**0.9965**
FaceDetection	0.5638	0.5599	0.5587	**0.5673**	0.5624
HandMovementDirection	0.4459	0.4189	0.4459	0.4459	**0.4595**
Heartbeat	**0.8**	0.7512	0.7561	0.7805	0.7902
MotorImagery	0.57	**0.66**	**0.66**	0.57	0.63
NATOPS	0.9833	0.9389	0.9556	0.9722	**0.9889**
PEMS-SF	**0.8671**	0.7399	0.7746	0.7746	0.8035
PenDigits	0.9826	0.9751	0.9771	0.9848	**0.9897**
Phoneme	0.2094	0.1342	0.0486	0.1321	**0.215**
SelfRegulationSCP2	0.5611	0.5944	**0.6**	0.5722	0.5833
SpokenArabicDigits	0.9823	0.9718	0.9659	0.9491	**0.9841**
StandWalkJump	0.6	0.4	0.4	**0.6667**	0.6
Average value	0.729	0.6986	0.6994	0.7194	**0.7373**
Wins&Ties	4	4	3	9	**8**
Avg. Rank	2.667	3.8	3.567	3.1	**1.867**

TABLE 6. Ablation Study with SLCNN as backbone.

Datasets	SLCNN				
	N/A	C	T	CT	TC
ArticularyWordRecognition	0.6967	0.9033	0.9633	**0.9867**	0.9633
AtrialFibrillation	0.4467	0.4	**0.4667**	**0.4667**	0.3333
BasicMotions	**1**	**1**	**1**	**1**	**1**
CharacterTrajectories	0.5857	0.6327	**0.9039**	0.8196	0.7695
FaceDetection	0.569	0.567	0.5678	**0.5692**	0.5664
HandMovementDirection	0.4324	**0.4865**	0.4324	0.4459	**0.4865**
Heartbeat	0.7707	0.7659	0.7707	**0.7756**	**0.7756**
MotorImagery	0.61	0.66	0.55	0.64	**0.68**
NATOPS	0.7278	0.7944	0.8389	**0.9833**	**0.9833**
PEMS-SF	0.7688	0.7399	0.7919	0.7746	**0.8092**
PenDigits	0.8602	0.8448	**0.9423**	0.936	0.9414
Phoneme	0.0734	0.1345	0.1766	**0.1816**	0.1789
SelfRegulationSCP2	0.5944	**0.6**	0.5889	0.5889	0.5944
SpokenArabicDigits	0.8386	0.9632	0.98	**0.9841**	0.9759
StandWalkJump	0.5333	0.4667	0.4667	**0.6**	**0.6**
Avg. Value	0.6338	0.6339	0.696	0.7101	**0.7105**
Wins&Ties	1	3	4	**9**	7
Avg. Rank	3.9	3.667	2.933	**2.133**	2.367

2.25%, and 1.18% improvement compared to CBAM, GC, and SE when the CT_CAM is applied to FCN presented in Table 4. From Table 5, we can know that the improvements of CT_CAM to CBAM, GC, and SE are 5.54%, 5.42%, and 2.49%, respectively. For SLCNN, the average accuracy can be improved dramatically by including attention module, especially for CBAM and CT_CAM, which consider both channel and temporal attention. However, when it comes to FCN and DenseNet, including CBAM, GC, SE provides even worse results than vanilla FCN. This suggests that the attention module can significantly enhance the performance of a simple network with relatively week representation power. The representation power of a deeper network might be suppressed due to the limit of data and the increase in complexity. Our CT_CAM module uses residual connection inside which allows the information flow explicitly into the next block and therefore the network's ability is not likely to be suppressed.

D. ABLATION STUDY

To validate the importance of the proposed attention module, four variants are compared. 1) **C**: It is a pure model of CCAM. In this case, the TCAM is removed. 2) **T**: It is a pure model of TCAM. In this case, the CCAM is removed. 3) **CT**: This is the proposed CT_CAM module. Detailed information is described in Section III-C. 4) **TC**: We exchanged the position of TCAM and CCAM. Detailed information is described in Section III-C. **N/A** means no attention modules are used. To fully illustrate the performance, SLCNN, FCN, and DenseNet are used as the backbone for these four variants in the 15 benchmark datasets. The results are presented in TABLE 6, TABLE 7, and TABLE 8.

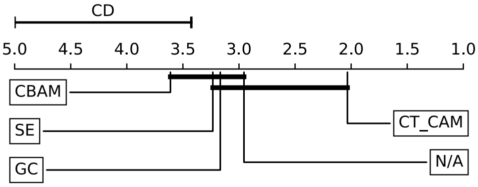

FIGURE 5. Critical difference diagram of the average ranks with the Nemenyi test ($p = 0.05$).

TABLE 7. Ablation study with FCN as backbone.

Datasets	FCN				
	N/A	C	T	CT	TC
ArticularyWordRecognition	0.9867	**0.99**	0.9833	0.9867	0.9833
AtrialFibrillation	0.4	0.4	0.4	**0.4667**	**0.4667**
BasicMotions	**1**	**1**	**1**	**1**	**1**
CharacterTrajectories	0.9916	**0.9951**	0.993	0.991	0.9944
FaceDetection	0.569	0.5638	**0.5721**	0.559	0.5633
HandMovementDirection	0.4459	0.4595	0.4459	**0.4865**	0.4459
Heartbeat	0.8098	**0.8146**	0.8049	0.8098	0.7854
MotorImagery	0.59	**0.63**	0.61	0.57	0.61
NATOPS	0.9833	0.9833	0.9833	**0.9889**	**0.9889**
PEMS-SF	0.7803	0.7688	**0.7977**	0.7746	0.7919
PenDigits	0.9889	0.99	**0.9909**	0.9886	0.9903
Phoneme	0.0871	0.1077	0.1387	0.1616	**0.1789**
SelfRegulationSCP2	0.5778	0.55	0.5889	**0.5944**	0.55
SpokenArabicDigits	0.9768	**0.9973**	0.9955	0.9923	0.9891
StandWalkJump	**0.6667**	0.5333	0.6	0.6	0.5333
Avg. Value	0.7236	0.7189	0.7269	**0.7313**	0.7248
Wins&Ties	2	**6**	4	5	4
Avg. Rank	3.4	2.8	**2.767**	2.967	3.067

TABLE 8. Ablation study with DenseNet as backbone.

Datasets	DenseNet				
	N/A	C	T	CT	TC
ArticularyWordRecognition	**0.99**	0.9567	0.94	0.9867	**0.99**
AtrialFibrillation	0.4	0.4	0.4	**0.4667**	**0.4667**
BasicMotions	**1**	**1**	**1**	**1**	**1**
CharacterTrajectories	0.9854	0.9916	0.9805	0.9937	**0.9965**
FaceDetection	0.5638	0.5656	**0.5721**	0.5692	0.5624
HandMovementDirection	0.4459	0.4054	**0.4865**	0.473	0.4595
Heartbeat	0.8	0.8049	0.8049	**0.8195**	0.7902
MotorImagery	0.57	0.62	**0.67**	0.66	0.63
NATOPS	0.9833	0.9833	0.9722	0.9833	**0.9889**
PEMS-SF	**0.8671**	0.763	0.8092	0.7977	0.8035
PenDigits	0.9826	0.9877	0.9871	0.9877	**0.9897**
Phoneme	0.2094	0.1891	0.1933	0.2088	**0.215**
SelfRegulationSCP2	0.5611	0.5778	0.5778	**0.5944**	0.5833
SpokenArabicDigits	0.9823	**0.9877**	0.9871	0.9843	0.9841
StandWalkJump	0.6	0.5333	0.6	**0.6667**	0.6
Avg. Value	0.729	0.718	0.732	**0.746**	0.7373
Wins&Ties	3	2	4	5	**7**
Avg. Rank	3.567	3.567	3.067	**2.267**	2.533

As illustrated in TABLE 6, the best average accuracy happens when the **TC** model is added. The **CT** model shows

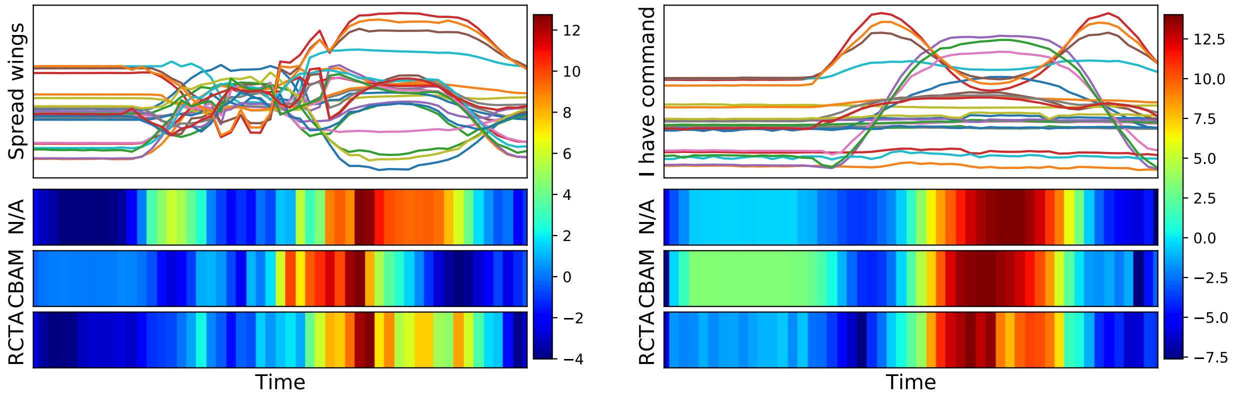

FIGURE 6. Visualizing high attention area with CAM in dataset 'NATOPS'.

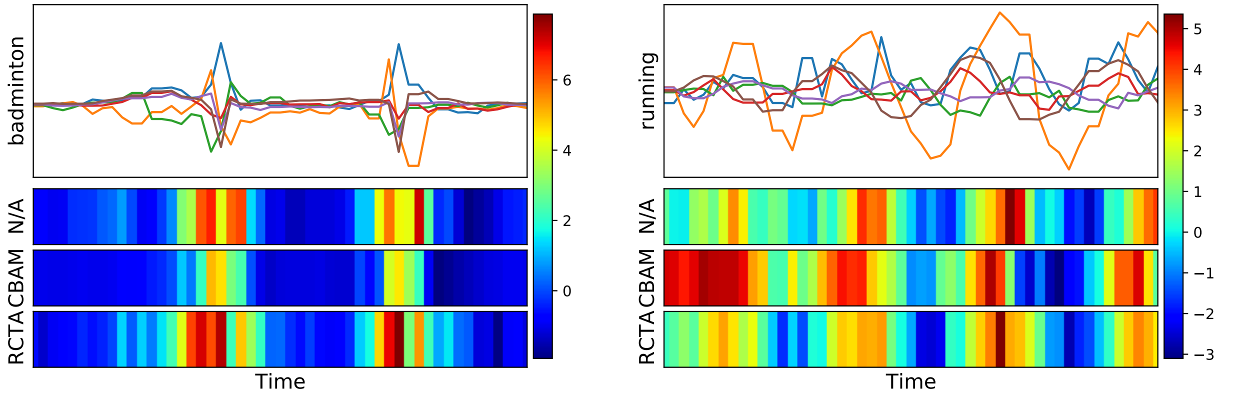

FIGURE 7. Visualizing high attention area with CAM in dataset 'BasicMotions'.

a slight lower accuracy than **TC** but with more numbers of Wins&Ties and better average rank than **TC**. Compared the **N/A** module in SLCNN, the performance of **TC** and **CT** has relatively improved 12.10% and 12.03%, respectively. From TABLE 7, we also can know that the **CT** module shows a higher average accuracy than other modules. However, the **C** and **T** achieve higher Wins&Ties and average rank, respectively. TABLE 8 shows similar results with TABLE 7 where the FCN is used as the backbone. It is easy to know that the **CT** achieves highest average accuracy and average rank.

It is shown in TABLE 6, TABLE 7, and TABLE 8 that adding **C** exhibits a small decrease in average accuracy while adding **T** alone displays a small average accuracy increase. But sequentially adding the CCAM and TCAM shows relatively large improvement. The reason might be that temporal attention can compensate for the channel features. This phenomenon is much obvious in the shallow CNN architecture but we empirically show that the CT_CAM block can enhance the performance of shallow and deep network architecture. TABLE 6, TABLE 7, and TABLE 8 also summarize the experimental results on different attention arrangement. From the results, it can be found that the channel-first order performs slightly better than the temporal-first order but they can be considered as almost equal. All the arranging methods outperform using only the channel or temporal attention independently, showing that utilizing both attention is crucial.

E. VISUALIZATION

We visualize the attention maps using CAM [33]. FIGURE 6, FIGURE 7, FIGURE 8, and FIGURE 9 show the attention map for N/A, CBAM, and CT_CAM in dataset 'NATOPS', 'BasicMotions', 'AtrialFibrillation', and 'StandWalkJump', respectively. SLCNN is used as the backend in this section. Only two samples of two classes in each dataset are randomly selected for visualization.

It is shown in FIGURE 6 that these three models highlights a similar region. These models highlight the plateau are for "I have a command" while focus on the transition area for "Spread wings". The N/A model clearly have a more wide spread attention region than CBAM and CT_CAM. The CBAM and CT_CAM modules help the network to focus on the informative area and related region.

From FIGURE 7, it is also can know CT_CAM, CBAM, and N/A model are focusing on the transition in both classes. However, the CT_CAM and CBAM have a more wide attention area. FIGURE 8 and FIGURE 9 present the similar scenarios where the changes of signal only occurs in a small area,

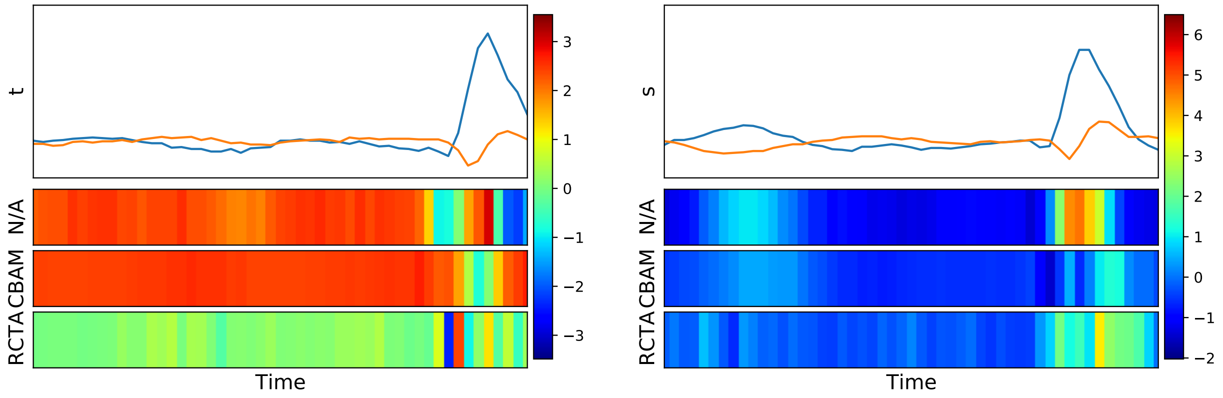

FIGURE 8. Visualizing high attention area with CAM in dataset 'AtrialFibrillation'.

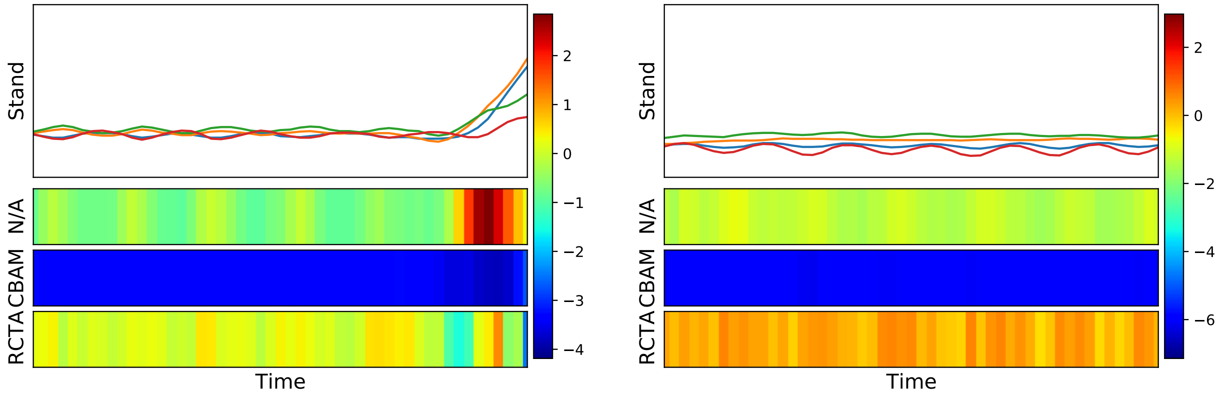

FIGURE 9. Visualizing high attention area with CAM in dataset 'StandWalkJump'.

and the signal remains stable in other areas. From these two figures, the proposed CT_CAM can not only obtain useful information in the sharply changing area of the signal, but also identify subtle changes of these signals. The performance of CBAM is even worse than N/A model in the two cases as CBAM cannot obtain such wider informative area as well as cannot observe the signal's subtle changes.

V. CONCLUSION

In this paper, the CT_CAM module is presented to improve the representation power of CNN networks for MTSC problem. This module consists of a channel and a temporal block, which focus on refining the feature from the two dimension, i.e. spatial and temporal, in multivariate time series. The experimental results in the public UEA archive demonstrate that the recent proposed DenseNet, FCN, and SLCNN combined with the proposed CT_CAM module achieve the state-of-the-art results compared to nine baseline methods. Compared with other attention modules, the proposed CT_CAM provides a better performance whether it is combined with SLCNN or FCN. The sensitivity analysis studies the impact of the number of hidden unit in GRU. From the experimental results, the proposed CT_CAM can enhance the performance of various CNN networks and CT_CAM can be an important component of CNN networks.

The focus of this work is to improve the performance of feature extraction ability of CNN by utilizing attention mechanism. According to the characteristics of time series data and drawing on the design ideas of attention mechanism in the direction of computer vision, we propose a sequential attention structure, which can learn temporal and spatial information simultaneously. This novel attention module can improve the accuracy of the model, but it will inevitably lead to the model being too cumbersome and not lightweight enough.

REFERENCES

[1] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias, "Attend and diagnose: Clinical time series analysis using attention models," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 4091–4098.

[2] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 3995–4001.

[3] A. Gumaei, M. M. Hassan, A. Alelaiwi, and H. Alsalman, "A hybrid deep learning model for human activity recognition using multimodal body sensing data," *IEEE Access*, vol. 7, pp. 99152–99160, 2019.

[4] X. Cheng, G. Li, A. L. Ellefsen, S. Chen, H. P. Hildre, and H. Zhang, "A novel densely connected convolutional neural network for sea-state estimation using ship motion data," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 9, pp. 5984–5993, Sep. 2020.

[5] X. Cheng, G. Li, R. Skulstad, S. Chen, H. P. Hildre, and H. Zhang, "Modeling and analysis of motion data from dynamically positioned vessels for sea state estimation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6644–6650.

[6] A. L. Ellefsen, P. Han, X. Cheng, F. T. Holmeset, V. Aesoy, and H. Zhang, "Online fault detection in autonomous ferries: Using fault-type independent spectral anomaly detection," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8216–8225, Oct. 2020.

[7] J. Korbicz, J. M. Koscielny, Z. Kowalczuk, and W. Cholewa, *Fault Diagnosis: Models, Artificial Intelligence, Applications*. Cham, Switzerland: Springer, 2012.

[8] P. Schäfer and U. Leser, "Fast and accurate time series classification with WEASEL," in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 637–646.

[9] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, May 2017, pp. 1578–1585.

[10] F. Karim, S. Majumdar, H. Darabi, and S. Harford, "Multivariate LSTM-FCNs for time series classification," *Neural Netw.*, vol. 116, pp. 237–245, Aug. 2019.

[11] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.

[12] H. I. Fawaz, B. Lucas, G. Forestier, C. Pelletier, D. F. Schmidt, J. Weber, G. I. Webb, L. Idoumghar, P.-A. Muller, and F. Petitjean, "Inceptiontime: Finding alexnet for time series classification," *Data Mining Knowl. Discovery*, vol. 34, no. 6, pp. 1936–1962, Nov. 2020,

[13] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[14] H. Ma, W. Li, X. Zhang, S. Gao, and S. Lu, "AttnSense: Multi-level attention mechanism for multimodal human activity recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, Aug. 2019, pp. 3109–3115.

[15] M. G. Baydogan, G. Runger, and E. Tuv, "A Bag-of-Features framework to classify time series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2796–2802, Nov. 2013.

[16] J. Lines and A. Bagnall, "Time series classification with ensembles of elastic distance measures," *Data Mining Knowl. Discovery*, vol. 29, no. 3, pp. 565–592, May 2015.

[17] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with COTE: The collective of transformation-based ensembles," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2522–2535, Sep. 2015.

[18] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu, "Tapnet: Multivariate time series classification with attentional prototypical network," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 6845–6852.

[19] R. A. Rensink, "The dynamic representation of scenes," *Vis. Cognition*, vol. 7, nos. 1–3, pp. 17–42, Jan. 2000.

[20] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.

[21] M. Jaderberg, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[22] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[23] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3024–3033.

[24] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi, "Gather-excite: Exploiting feature context in convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 9401–9411.

[25] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[26] Y. Cao, J. Xu, S. Lin, F. Wei, and H. Hu, "GCNet: Non-local networks meet squeeze-excitation networks and beyond," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1971–1980.

[27] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao, W. Liu, and T.-S. Chua, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5659–5667.

[28] A. Bagnall, H. Anh Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh, "The UEA multivariate time series classification archive, 2018," 2018, *arXiv:1811.00075*. [Online]. Available: http://arxiv.org/abs/1811.00075

[29] A. Paszke, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 8026–8037.

[30] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980

[31] M. Shokoohi-Yekta, J. Wang, and E. Keogh, "On the non-trivial generalization of dynamic time warping to the multi-dimensional case," in *Proc. SIAM Int. Conf. Data Mining*, Jun. 2015, pp. 289–297.

[32] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[33] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

XU CHENG (Member, IEEE) received the master's degree in computer science and technology from the Zhejiang University of Technology, Hangzhou, China, in 2015. He is currently pursuing the Ph.D. degree. His current research interests include sea state estimation, data analysis, neural networks, and ship motion modeling.

PEIHUA HAN received the bachelor's and master's degrees from the Department of Architecture and Civil Engineering, Zhejiang University, China, in 2019. He is currently pursuing the Ph.D. degree with the Norwegian University of Science and Technology (NTNU), Aalesund, Norway, as part of the Mechatronics Laboratory, Department of Ocean Operations and Civil Engineering. His current research interests include fault diagnosis and prognostics, predictive maintenance, machine learning, and uncertainty qualification.

GUOYUAN LI (Senior Member, IEEE) received the Ph.D. degree from the Department of Informatics, Institute of Technical Aspects of Multimodal Systems, University of Hamburg, Hamburg, Germany, in 2013. Since 2014, he has been with the Mechatronics Laboratory, Department of Ocean Operations and Civil Engineering, Norwegian University of Science and Technology, Aalesund, Norway. In 2018, he become an Associate Professor in ship intelligence. His research interests include path planning, ship motion prediction, maneuvering control, artificial intelligence, optimization algorithms, and locomotion control of bioinspired robots. In these areas, he has authored or coauthored more than 60 articles.

SHENGYONG CHEN (Senior Member, IEEE) received the Ph.D. degree in computer vision from the City University of Hong Kong, Hong Kong, in 2003. He is currently a Professor with the Tianjin University of Technology. He received a fellowship from the Alexander von Humboldt Foundation of Germany and worked at the University of Hamburg, from 2006 to 2007. His research interests include computer vision, robotics, and image analysis. He is a Fellow of IET and a Senior Member of CCF. He has published over 100 scientific articles in international journals. He received the National Outstanding Youth Foundation Award of China, in 2013.

HOUXIANG ZHANG (Senior Member, IEEE) received the Ph.D. degree in mechanical and electronic engineering, in 2003, and the Habilitation degree in informatics from the University of Hamburg, in February 2011. Since 2004, he worked as a Postdoctoral Fellow with the Department of Informatics, Faculty of Mathematics, Informatics and Natural Sciences, Institute of Technical Aspects of Multimodal Systems (TAMS), University of Hamburg, Germany. He joined the NTNU (before 2016, Aalesund University College), Norway, in April 2011, where he is a Professor on robotics and cybernetics. The focus of his research lies on two areas. One is on biological robots and modular robotics. The second focus is on virtual prototyping and maritime mechatronics. In these areas, he has published over 130 journal and conference papers and book chapters as author or coauthor.