Computer Communications 182 (2022) 52–59

Contents lists available at ScienceDirect

Computer Communications

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

LSTM-MFCN: A time series classifier based on multi-scale spatial–temporal features

Liang Zhao [a,b], Chunyang Mo [a,b], Jiajun Ma [a,b], Zhikui Chen [a,b], Chenhui Yao [c,*]

[a] School of Software Technology, Dalian University of Technology, Dalian, 116600, China
[b] Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian, 116600, China
[c] The First Affiliated Hospital of Dalian Medical University, Dalian, 116000, China

ARTICLE INFO

Keywords:
Time series classification
Fully convolutional networks
LSTM
Multi-scale
Spatial–temporal features

ABSTRACT

Time series classification (TSC) task attracts huge interests, since they correspond to the real-world problems in a wide variety of fields, such as industry monitoring. Deep learning methods, especially CNN and FCN, shows competitive performance in TSC task by their virtue of good adaption for raw time series and self-adapting extraction of features. Then various variants of CNN are proposed so as to make further breakthrough by the better perception to characteristics of data. Among them, LSTM-FCN and GRU-FCN who learn spatial and temporal features simultaneously are the most remarkable ones, achieving state of the art results. Therefore, inspired by their success and in consideration of the discriminative features implied in time series are diverse in size, a multimodal network LSTM-MFCN composed of multi-scale FCN (MFCN) and LSTM are proposed in this work. The gate-based network LSTM naturally fits to various terms time dependencies, and FCN with multi-scale sets of filters are capable to perceive spatial features of different range from time series curves. Besides, dilation convolution is deployed to build multi-scale receptive fields in larger level without increasing the parameters to be trained. The full perception of large multi-scale spatial–temporal features lead LSTM-MFCN to possess comprehensive and thorough grasp to time series, thus achieve even better accuracies. Finally, two representative architectures are presented specifically and their experiments on UCR datasets reveals the effectiveness and superiority of proposed LSTM-MFCN.

1. Introduction

Time series are the real-world data in a wide variety of fields, including but not limited to healthcare records, industry monitoring signals and stock tendencies. Thus, Time series classification (TSC) task has been a significant topic of data mining, attracting huge interests.

Time series instances are the successive and sequential values harvested at equally spaced time steps. They are usually unaligned in time steps due to the different-extent delays and probably have overhigh dimensionalities. Traditional classification methods are overshadowed when comes to time series, for they have poor adaption to above two drawbacks. Distance-based methods are the representative. They commonly employ K Nearest Neighbour (KNN) classifier with Dynamic Time Wrapping [1] (DTW) as distance function [2]. However, DTW fails to correctly match the unaligned points if the peaks and valleys of series are not corresponding in amounts, thus the generating distance is not a reliable factor to distinguish instances. And the long-steps accumulation will weaken the global differences among instances. So distance-based methods achieve inferior performance. Facing this issue, two directions are developed. On the one hand, feature-based methods [3] which start with feature-engineering or data preprocessing are proposed to achieve the low dimensional and discrete representation of time series, thus traditional methods receiving them will make better judgements, such as Shapelet [4] and Symbolic Aggregate Approximation (SAX) [5]. On the other hand, deep learning methods [6] are employed since they have certain tolerance for the unaligned raw input and automatic extraction of the features benefiting for classification task. Typical ones like MLP, CNN, ResNet and LSTM, and some multimodal hybrid structures derived from them both involve in this task. In general, feature-based methods discard a lot of points from raw data, thus many subtle patterns of time series could be lost during the manual feature-extraction phase. While approaches based on deep learning obtain features via a full search and detection on the internal structure of raw data. Consequently, they outperform feature-based techniques, even the ensemble of them.

Among deep learning TSC methods, CNN is the most highly competitive, and its specific form with global pooling layer, the fully

* Corresponding author.
E-mail addresses: liangzhao@dlut.edu.cn (L. Zhao), 1798797049@qq.com (C. Mo), 2267572682@qq.com (J. Ma), zkchen@dlut.edu.cn (Z. Chen), gavinyaoch@126.com (C. Yao).

https://doi.org/10.1016/j.comcom.2021.10.036
Received 27 May 2021; Received in revised form 16 October 2021; Accepted 26 October 2021
Available online 3 November 2021

convolutional networks (FCN), has proven to be an appropriate architecture for TSC problem. The core competence of CNN lies in the capture to spatial features. During past years, different variants of CNN are presented aiming at earning higher accuracies by better perceiving spatial patterns. Res-FCN [7] are proposed to strengthen the advantage of convolution operation by cascading ResNet and FCN in depth and dilated convolution are leveraged in similar field of multivariate time series forecasting to reinforce extraction along breadth [8]. One of them focuses on high-level features and the other on the long-term ones. MFCN [9] and MCNN [10] are designed to extend CNN by capture multi-scale characteristics all at once. All of above variants is demonstrated to be effective but have limitations of providing single type information for classification task, as they merely depend on the contribution of convolution. Thus, other powerful deep learning models like LSTM and GRU which naturally match time-related tasks are taken into account. Multimodal networks GRU-FCN [11], LSTM-FCN [12] simultaneously absorb the spatial features along with time series curves and the time-dependent features which correspond with history value, producing higher accuracies than all above variants, reaching state of the art results in current. However, the remarkable achievement does not prevent us from scaling a new height, as it has not made exhaustive extraction of features yet. At least, they overlook that the discriminative features implied in time series are of different scale.

Therefore, inspired by above successes and imperfection, we design a multimodal network (LSTM-MFCN) which conjoin LSTM with multi-scale FCN parallelly, in this work. Wherein, a time series is regarded as a successive curve; MFCN with varisized convolution receptive fields are utilized to detect the spatial features in a variety size from curves and LSTM is adopted to learn different terms of time-dependencies related to history value. Now that most time series datasets are short of instances and thus have modest training capacities, dilation convolution is deployed to generate the large size receptive fields in MFCN without increasing the parameters to be trained. To evaluate the proposed method, architectures with double scales and triple scales convolution are presented as the representatives. We compare them with the related and typical TSC baselines on datasets from UCR Time Series Classification Archive [13]. The core contributions of this paper can be summarized as follows:

- This paper employed dilated convolution in TSC problem. In regression issues of time series, dilated convolution has contributed to capture long-distance dependencies of the observed variables. Similarly, for TSC problems, we take the large-scale spatial features into consideration.
- We present a novel implementation of ‘multi-scale’ via the combination of dilated and traditional continuous convolution. Compared with previous organization manners, like the homonymous MFCN [9] and the classical MCNN, it provides larger receptive fields under almost the same parameter amounts, without information loss.
- LSTM-MFCN focuses on both spatial and temporal features of large multi-scales, producing more comprehensive and thorough grasp to time series data than any existing multimodal TSC model which is mainly based on FCN and LSTM. The achieved promising result brought by above virtues pushes this kind of multimodal NNs a step further.

The rest of this work is organized as follows. Section 2 reviews previous related studies, Section 3 describes the components and structures of proposed method. Section 4 presents the extensive comparative evaluation between LSTM-MFCN and the benchmark approaches, as well as the ablative validation. Finally, in Section 5, conclusions of the paper are drawn.

2. Related work

Deep learning methods have brought revolutionized improvements to fields of computer vision and natural language processing, with the advent of novel deeper architectures such as ResNet, FCN and LSTM, which learn ordinal and distributional relations among the fed pixels or words. Time series issues, with the same form as sequential word vector, require temporal order and spatial distribution information as well. In this section, we briefly review some related feature extraction cases in TSC fields, including the co-learning of spatial–temporal features by multimodal networks.

Convolutional Neural Network shows an overwhelming advantage on TSC task. An intuitional example is the evaluation to strong baselines conducted by Z. Wang et al. [14], where the simple convolution-based deep learning structures ResNet and FCN obtained more competitive results than the notable non-deep-learning ensembles COTE [15], which was the SOTA result at that moment. It is all aware that the superiority of CNN lies in the detection of highly distinguishable spatial features. However, these geometric patterns exist in time series curves has different scales. Approaches are thus dedicated to give good TSC solutions via better accesses of them. MCNN [10] is a deep network with fixed-size filters but its inputs will undergo a down-sampling phase prior to training with different sampling rate, thus it has equivalent multi-range perceptions to the raw time series. W. Zhou et al. [9] organize a set of convolution blocks with multi-scale filters parallelly then connect them by a global pooling layer, composing an ensemble of FCN, in order to learn TSC task via these diverse spatial patterns. B. Qian et al. [16] notice that a same pattern will shows various sizes when appears in different time series instances, due to the unaligned time steps. To extract these instance-specific patterns flexibly, their proposed DMS-CNN use learnable soft masks to generate convolution filters with dynamic length. Judging from these achievements, multi-scale convolution seems to be an effective way to fully learn spatial feature. Then another problem is closer to view. Conventionally, a large-size receptive field relies on a wide one-dimensional convolution filter, Or, on a skipping sampling, like MCNN. But deep learning methods are still prone to overfitting [17] when the time series training set has too few instances and the model has great amounts of parameters to be trained. It is the same pity that a cut-down representation by sampling will lead the loss of subtle patterns. Previous works in similar domains offers the third solution. A. Borovykh et al. [8], inspired by the deep WaveNet architecture [18], use the stacks of dilated convolutions to detect a broad range of time series tendency. R. Xi et al. [19] utilize an end-to-end dilated convolutional neural network to the task of human activities recognition which based on the real-word time series. These successful applications on time series motivates our usage of dilated convolution.

In the normal sense, recurrent neural networks naturally fit time-related data. Their advanced variants LSTM and GRU, with self-adapting gating mechanism, can generate appropriate terms of time dependencies without intervention. But researches have proven that they are not capable to TSC task on their own [20]. LSTM and GRU mostly play the role of auto-encoder, to yield a good representation of time series prior to the learning phase of classifiers. P. Malhotra et al. [21] employ the RNNs with Gated Recurrent Units to build a generic off-the-shelf feature extractor for time series, and its validation by SVM classifiers yields significantly better results than corresponding end-to-end processes. N. Mehdiyev et al. [22] obtain compressed feature representations of time-series data in an unsupervised manner using stacked LSTM autoencoders. These sequence-to-sequence preprocesses are built on condition that different-terms temporal information can be well extracted from raw data during encoding phase.

Recently, further breakthroughs are achieved by join two strong hands together directly. C. Yang et al. [23] attach a RNN in the rear to CNN to learn the time-dependencies among their given spatial features. And G. Liu et al. [24] take varisized spatial features into account on

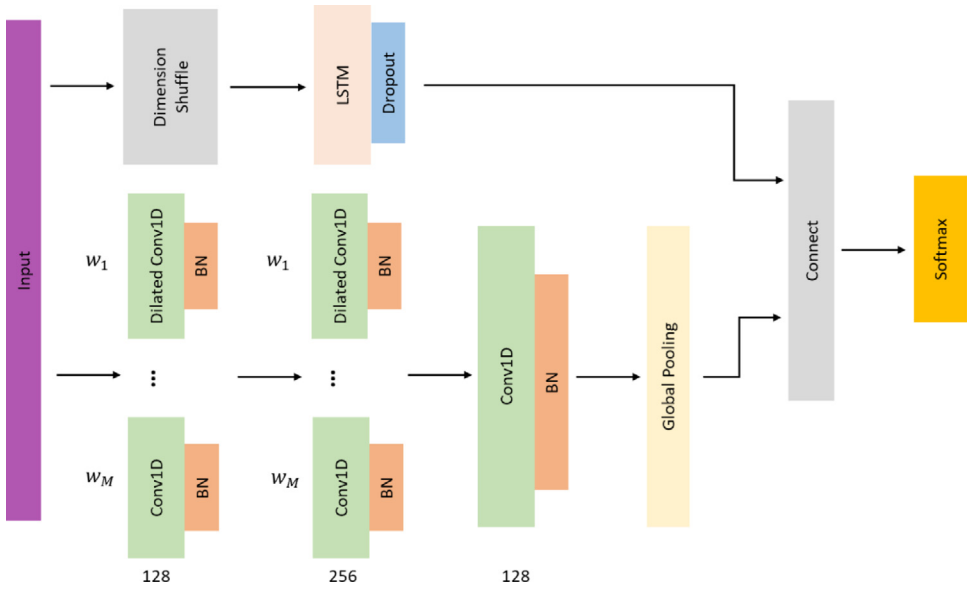

Fig. 1. The overall architecture of proposed LSTM-MFCN.

this basis. Where time series are down-sampling into different scales and in each scale branch, series are extracted by convolutional units, then the time dependency and the complementary across different scales are captured through RNN. These serial combinations of CNN and RNN are mainly presented for application scenario. Though it is hard to give a fair evaluation without seeing their work on common problems, we could infer the run-of-the-mill effectiveness from their ordinary advantage over traditional machine learning methods. While, the parallelly connection is indeed excellent, being one of the hot spots of current researches. LSTM-FCN and ALSTM-FCN [12] are the first proposed parallel spatial–temporal learning networks in TSC field, where times series go through independent learning phase in two modules and finally combined by connection layer. They outperform most state-of-the-art models. Coincidentally, at the same period, a parallel integration of attention-based LSTM and CNN has been exploited as well [25], since it is an easily available inspiration for combining the two significant features together. Then, F. Karim et al. [26] perform thousands of ablation tests on LSTM-FCN and ALSTM-FCN, showing that the LSTM and the FCN blocks perform better when applied in a conjoined manner. Which indicates the necessity of spatial–temporal conjunction, as well. N. Elsayed et al. [11] use GRU module to realize the extraction of time-related features, composing a same temporal and spatial learning structure with LSTM-FCN, and the GRU-FCN delivers competitively accurate results, as well. Above achievements have repeatedly demonstrated the significance of combining isolated spatial and temporal patterns.

In summary, the researches mentioned above make good use of time and space features. Therefore, we intend to investigate a new multimodal architecture on this basis, with the multi-scale extraction, constructs a full and deep acquisition to time series.

3. The proposed method

As stated earlier, we present a method called LSTM-MFCN, which combines multi-scale FCN with LSTM in parallel. Its overall architecture is shown in Fig. 1. Noteworthily, in our model, dilated convolution is employed as an alternative manner to the down-sampling or traditional continuous convolution, for large receptive fields. Therefore, LSTM-MFCN is able to generate accurate judgments by the fully learning of large varisized spatial and temporal characteristics. In this section, structures of two parallel components are introduced at first, then are the representatives with specific hyperparameters.

Multi-scale fully convolutional network block consists of three layers which have filter amounts of 128, 256 and 128, and filter size S of 8, 5 and 3, respectively. In fact, the overall hyperparameters will retain the same with that in LSTM-FCN, in order to perform a fair comparison. In the first two layers, filters are divided into M independent groups according to the given proportions, while the size and total number of filters keep constant. With different dilation rate d in each group, multi-scale receptive fields can be obtained, so as to extract the features of variable-length from input series IP. The convolution layer in each branched group is followed by a batch normalization and a RELU activation, and then their exporting features SF (short of spatial features) are concatenated together as a vector feeding into next layers, on account of the feature detection performed in deep neural networks is based on all the information from prior layer. So, our implement of multi-scale FCN is not identical to the previous homonymous classifier [9]. This progress can be described by follow formulas, where L is the $L-th$ layer and i is the ith of M scales .

$$SF_{Li} = BN\&RELU(CONV1D(IP_L, S_L, d_i)) \tag{1}$$

$$IP_{L+1} = SF_{L1} \cup SF_{L2} \cup, \ldots, SF_{Li} \cup, \ldots, \cup SF_{LM} \tag{2}$$

Further, the quantity ratio of different scale receptive fields should be a flexible hyperparameter, since the spatial features in each length have no fixed proportion. As shown in Fig. 1, they are denoted by $w_1, w_2, \ldots, w_M$. Thus, filter amounts can be calculated by formula (3). And in this formula, NF_{Li} and NF_L are amounts of the ith scale and the total of filters in this layer, respectively. At the end of MFCN block, outputs will experience a global pooling procedure.

$$NF_{Li} = NF_L * \frac{w_i}{\sum_i^M w_i} \tag{3}$$

In the other components Long Short-Term Memory module, time series data will first pass through a dimension conversion. A time series can be regarded as a list of numerical value, like $X_c = \{X_c^1, X_c^2, \ldots, X_c^T\}$, and a dataset as $X = \{X_1, X_2, \ldots, X_c, \ldots, X_N\}$. In MFCN module, Xc are inputted as a whole, and convolution filters slide on this one-dimensional structural data. But, LSTM which depends on the history states of previous input is not a point-to-point model. Consequently, a time series are converted into a serial input which feeds one value X_c^i at each time. LSTM has neuron number of 8, 64 or 128, which is up to specific dataset, as the time-dependencies to be learned are of different complexity. And a dropout is imported into this block

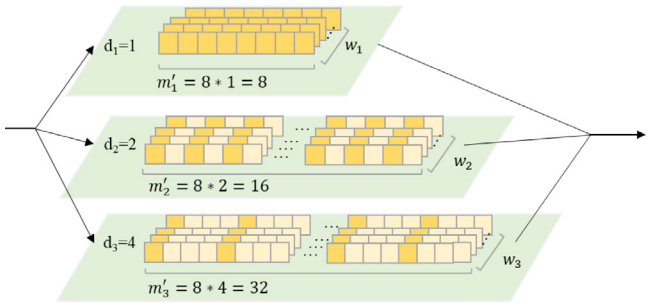

Fig. 2. An example of multi-scale convolution layers where M=3, dilation rate=1,2 and 4.

for a robust performance. Finally, a concatenation of the spatial and temporal features from multi-scale FCN and LSTM network can be obtained in a fully-connected layer, then it is passed onto *SoftMax* function to get prediction label.

Specifically, in MFCN, dilation convolution plays a vital role in building large and middle scale receptive fields, under circumstance of filter size is constant in each branch. In the regular convolution filter matrix, each element is close to each other, while dilated convolution expands its receptive range by inserting holes into those elements. The amounts of holes are up to the hyper-parameter dilation rate *d*. When the *d* has value of *n*, *n*-1 holes will be added between adjacent elements in the convolution kernel. The equivalent receptive fields of dilation convolution depends on the size of filter *m* and dilation rate *d*, and is given by :

$$m' = m * d \tag{4}$$

Wherein, *m'* is the receptive field. Dilated convolutional filter enlarges the range of receptive field without squeezing information or increasing the number of parameters. Meanwhile, small-scale ones can be implemented by traditional convolutions, or it can be regard as dilated convolutions with dilation rate $d = 1$. Fig. 2 displays an example of a layer which perform three scale convolutions with dilation rate of 1,2, and 4. In addition, though dilation is considered to be beneficial for learning long-term features from the high-dimension and successive time series, an over-high dilation rate corresponds to a wide skipping over series nodes, which may result in wrong perception of patterns. Hence, our model requires dilation rate to be less than or equal to 4.

Algorithm 1 The overall process to build a LSTM-MFCN classifier

Input:
D, T and Y_D,Y_T : time series train set, test set and their label
M: scales amount
R: the dilation rate in each scale, $R = d_1, d_2, ..., d_M$
W: a set of all possible scales proportions
Output: Prediction label Y_t of T
$NF_1 = 128, NF_2 = 256, NF_3 = 128, S_1 = 8, S_2 = 5, S_3 = 3$
// Constant structure hyperparameters, where S is the size of filter
for w in W **do** *// w represents for the whole of* w_1, w_2,...,w_M
  for $L = 1$, 2 **do** *// The first two layers in MFCN block*
    for i in range(1,M) **do**
      $NF_{Li} \leftarrow w, i, NF_L$ *// Obtain the number of filters in all scales by formula 1.*
    end for
  end for
  for N in { 8, 64, 128 } **do** *// Number of LSTM neurons.*
    Obtain a specific LSTM-FCN via N, R, all the NF_{Li} and above constant hyperparameters
    for $k = 1$, 2 **do** *// Perform the experiments twice.*
      Train LSTM-MFCN (Algorithm 2) with D, Y_D.
      Distinguish instances in test set T by the trained LSTM-DFCN with minimum loss.
      Save the classification results in this turn.
    end for
  end for
end for
Generate Y_t from the best predictions of each dataset.
return Y_t

With above descriptions, the overall process and data flow of LSTM-MFCN are listed in Algorithm 1 and 2, respectively. At first, we should be clear about the constant structural parameters of LSTM-MFCN (line 3 of Algorithm 1). Then, the other parameters like the number of multi-scale M, dilation rate $d_1, d_2, \ldots, d_M$ and all possible multi-scale proportion combinations W need to be settled down. When starting our method, a possible combination of multi-scale ratio w is selected from W, and the undetermined hyperparameters of the MFCN module under this ratio combination can be calculated (lines 4-9). After that, the neurons amount N in the LSTM module should be to explored restrictively, and each value of N determines a specific model structure. Under this structure, training set D are used to train LSTM-MFCN and predict test set T (lines 11–17). Each input of LSTM-MFCN (Algorithm 2), will undergo several spatial-features extractions at convolution branches and then a concatenation, in the first two layers of MFCN module (lines 4–10), followed by a single scale convolution in the third layer. Simultaneously, LSTM are able to learn the temporal-dependencies from the conversed time series (lines 13–15). Finally, the two Kinds of discriminative features are mapping to a predictive result (lines 16–17). The best results given by LSTM-MFCN during the above structural exploration process are retained as its performance at M scales (line 19, Algorithm 1).

Algorithm 2 Data flow in LSTM-MFCN

Input: Time series X_c and hyper-parameters mentioned in Algorithm 1
Output: Prediction label Y_{pred} of X_c
$IP = X_c$
for $L = 1$, 2 **do**
  for i in range (1,M) **do**
    SF_{Li} = Conv1D (IP, NF_{Li}, S_L, d_i) *// SF:Spatial features*
    SF_{Li} = BN&RELU (SF_{Li})
  end for
  IP= $SF_{L1} \cup SF_{L2} \cup, ..., SF_{Li} \cup, ..., \cup SF_{LM}$
end for
SF = BN&RELU (Conv1D (IP, NF_3, S_3)) *// The Single-scale convolution performed in layer three*
SF = GlobalPooling (S_F)
$X = X_c.shuffle()$
TF= LSTM (X) *// TF:Temporal features*
TF= dropout (TF)
ST= SF ∪ TF *// ST:Spatial-Temporal features*
Y_{pred}=connect&softmax(ST) *// The final fully-connected layer.*
return Y_{pred}

We come up with two specific architectures as the representatives of proposed LSTM-MFCN, they are models with double-scale (M=2) and triple-scale (M=3) convolution, respectively. For the sake of description, we denote them by LSTM-DFCN (Double) and LSTM-TFCN (Triple), and use LSTM-MFCN as the whole of them when necessary. The major hyper-parameters of the two are shown in Table 1. Wherein, we intend to construct pyramidal hierarchy receptive fields in MFCN, so it requires the dilation rates in rear layer to be less than or equal to that in prior layer. Particularly, we explore two dilation rates at the first layer of LSTM-DFCN, since it is not clear which size will yield bigger gains, but only one of their results will be used at final. In addition, the small receptive field in DFCN and TFCN is exactly the same, thus we give a common description about this part, in this table.

Table 1
Structural hyperparameters of LSTM-DFCN and LSTM-TFCN.

	Hyperparameters	First layer	Second layer	Third layer	Connection layer
Common part	Filter amounts	128	256	128	
	Small receptive field	8*1	5*1	3*1	
DFCN block	Dilation rate d2	4/2	2	1	
	Large receptive field	32/16*1	10*1	3*1	
	Scales proportion	3:1/2:2/1:3	-		Neuron numbers = Class numbers
TFCN block	Dilation rate d2	2	2	1	
	Middle receptive field	16*1	10*1	3*1	
	Dilation rate d3	4	4	1	
	Large receptive field	32*1	20*1	3*1	
	Scales proportion	2:1:1/1:2:1/1:1:2	-		
LSTM block	Neuron number	8/64/128			
	Dropout rate	0.8			

Table 2
Hyperparameters of LSTM-MFCN evaluation experiment.

Hyper-parameters	Setting
Batch size	128
Optimizer	Adam
Learning rate	10^{-3} to 10^{-4}
Loss function	Cross-entropy

4. Experiments and results

In this section, we will perform two sets of experiments on the UCR datasets to verify the proposed method. The UCR archive is a collection of time-series data with category labels, and has been the standard validation means for time series classification problem. It got name for deriving from the University of California, Riverside. At present, it contains 128 specific time series data sets, including motion, ECG, sensor and other real-world data, as well as some simulated series. For convenience and fairness, we just evaluate LSTM-MFCN on the datasets adopted by Wang's work [14]. At first, a group of experiments are carried out to evaluate its performance. Then, by ablation experiments, the contributions of our innovative modification are revealed. All experiments are implemented on keras framework based on TensorFlow.

4.1. Evaluation experiments

On each given architecture determined by M, proportion of scales and amount of LSTM neuron can be tuned restrictively. For LSTM-DFCN, LSTM-TFCN and LSTM-FCN, we retain the trained model with minimum loss, and use the best test accuracy of each dataset as results. It is to be noted that the experiment of LSTM-DFCN and LSTM-TFCN is performed twice and LSTM-FCN is repeated for six times, to guarantee the fairness of comparisons. The other hyperparameters of these experiments are given in Table 2. If the loss does not decrease during 100 epochs of training, the learning rate will reduce by a rate of $1/\sqrt[3]{2}$ until reach the minimum.

Extensive baselines are adopted to prove the effectiveness of LSTM-MFCN, including: single modal deep learning methods [14] FCN and ResNet; multi-modal networks MCNN [10], MFCN [9], LSTM-FCN [12] and GRU-FCN [11]; typical and competitive feature-based methods BOSS [27], COTE [15], HIVE-COTE [28] and distance-based method LWDTW [29], PROP [2]. Among them, the four multimodal ones are closely related to our proposed method, hence convictive comparisons can be constructed. Metrics of this evaluation involves average accuracy, average rank, winning times and the mean error per class (MPEC) which is proposed in the work of Z. Wang et al. [14]. Further, the rank of all methods together with a critical distance given by hypothesis testing [30] could measure the achieved improvements. Critical distance (CD) is an alternative manner to pairwise comparison between methods. Only when the ranking gap is larger than CD, the performance difference can be considered to be significant. Suppose k algorithms are evaluated on N datasets, the critical distance can be calculated by the following formula, where q_α is a coefficient of probability.

$$CD = q_\alpha \times \sqrt{\frac{k(k+1)}{6N}} \tag{5}$$

Table 3 summarizes the experimental results of all approaches. Limited by space, methods names are further abbreviated, L, G and H are the short for LSTM, GRU and Hive. The given results of LSTM-DFCN are under the case where the first dilation rate is 4. Wherein, 'Max' represents the optimal case between LSTM-DFCN and LSTM-TFCN, which also means a superior state of LSTM-MFCN that can be achieved after sufficient tune. But it does not participate in ranking competitions. The bold font highlights the top results. Fig. 3 presents critical difference diagram with significance level α=0.05.

From these results, it can be found that LSTM-MFCN delivers state of the art performance on accuracy of 90.24%, winning times of 12 out the 44 datasets and rank of 3.295 among the 13 classifiers. Not surprisingly, the distance-based method LWDTW and ensemble PROP have the most disappointing results, for their limitations in handling unaligned raw data. Single modal deep classifiers FCN and ResNet produce the same-level performance with superior feature-based individual method BOSS and ensemble method COTE. Even if the new breakthrough in feature-based models, Hive-COTE, has caught up with MFCN and LSTM-FCN by its complicated preprocessing and learning phase, it still shows inferiority when compared with LSTM-MFCN. These cross-kinds comparisons place deep learning methods in the first position.

Then, moving eyes to the superior multimodal networks, the LSTM or GRU augmented methods all outperform MFCN and MCNN, which confirms time-dependencies plays a vital important role in TSC task, as well. The proposed LSTM-MFCN achieves even better performance in accuracy, rank and MPEC than LSTM-FCN and GRU-FCN. And our networks are proven to have a significant advantage over LSTM-FCN by Fig. 3. This can be attributed to that multi-scale and larger receptive fields by dilated convolution yield more comprehensive spatial–temporal features than constant-length and narrow-range ones. Thus, our LSTM-MFCN could grasp much more discriminative information from this thorough learning, giving more correct distinguishments. Specially, GRU-FCN wins most times but has a poor average accuracy and rank. It can be interpreted as lower robustness than LSTM-FCN and LSTM-MFCN, outstanding on some datasets but much inferior on others. Similarly, the excellent feature-based ensemble method HIVE-COTE also numerically weak in that two metrics.

Furthermore, the two representatives LSTM-DFCN and LSTM-TFCN both achieve superior results on basis of LSTM-FCN. But it should be noted that double-scale lags behind triple-scale. The reason is that TFCN module has respective fields of small, middle and large scale, thus a more comprehensive coverage to varisized features. Which proves that in our method, respective fields of each size attend to their own duties. In addition, 'Max' produces an overwhelming advantage over the other approaches, and indicates a great potential to the more promising results under sufficient hyperparameter-exploring.

Table 3
Experiment results of our method and baselines on UCR datasets.

Dataset	FCN	ResNet	MCNN	MFCN	COTE	LWDTW	PROP	BOSS	H-COTE	L-FCN	G-FCN	L-DFCN	L-TFCN	Max
Adaic	0.857	0.826	0.769	0.829	0.767	0.614	0.647	0.780	0.815	0.834	**0.893**	0.844	0.852	0.852
Beef	0.750	0.767	0.633	0.767	0.867	0.633	0.633	0.800	0.723	**0.900**	0.867	0.833	0.833	0.833
CBF	0.992	0.994	0.998	**1.000**	0.999	0.998	0.998	**1.000**	0.999	0.996	**1.000**	0.999	0.999	0.999
ChloCon	0.843	0.828	0.797	0.803	686	0.644	0.640	0.666	0.725	0.819	**0.998**	0.830	0.840	0.840
Cin_ECG	0.813	0.828	0.942	0.810	0.936	0.935	0.938	0.875	**0.988**	0.907	0.876	0.917	0.918	0.918
Coffee	**1.000**	**1.000**	0.964	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.988	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
CricKX	0.815	0.821	0.818	0.828	**0.846**	0.774	0.797	0.741	0.830	0.795	0.802	0.836	0.836	0.836
CricKY	0.792	0.805	0.846	0.836	0.833	0.759	0.844	0.792	0.837	0.797	0.812	0.851	**0.862**	**0.862**
CricKZ	0.813	0.831	0.858	0.856	**0.872**	0.749	0.844	0.754	0.848	0.818	0.817	0.859	0.862	0862
DiaSRed	0.931	0.931	0.977	**1.000**	0.918	0.977	0.941	0.954	0.942	0.951	0.964	0.944	0.944	0.944
ECG5Day	0.990	0.955	**1.000**	0.999	**1.000**	0.835	0.822	**1.000**	0.990	0.991	0.993	0.999	0.999	0.999
FaceAll	0.929	0.834	0.765	0.876	0.895	0.836	0.885	0.790	**0.996**	0.944	0.915	0.884	0.900	0.900
FaceFour	0.932	0.932	**1.000**	0.955	0.909	0.886	0.909	**1.000**	0.950	0.955	0.864	0.955	0.955	0.955
FacesUC	0.948	0.958	0.937	0.967	0.943	0.908	0.937	0.958	**0.984**	0.937	0.947	0.963	0.964	0.964
50Words	0.679	0.727	0.810	0.749	0.809	0.769	0.820	0.699	0.807	0.807	0.749	0.824	**0.835**	**0.835**
Fish	0.971	0.989	0.949	**0.994**	0.971	0.846	0.966	0.989	0.976	0.966	0.977	0.989	0.989	0.989
GunPnt	**1.000**	0.993	**1.000**	**1.000**	0.993	0.933	0.993	**1.000**	0.997	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Haptics	0.551	0.505	0.470	0.513	0.512	0.409	0.416	0.464	0.530	0.510	0.529	**0.581**	0.575	**0.581**
InlSkat	0.411	0.365	0.382	0.338	0.449	0.398	0.433	0.489	**0.526**	0.475	0.481	0.475	0.493	0.493
ItPwD	0.970	0.960	0.970	0.965	0.964	0.956	0.961	0.947	0.968	0.963	**0.973**	0.964	0.969	0.969
Light2	0.803	0.754	0.836	0.770	0.836	**0.902**	0.885	0.852	0.797	0.787	0.743	0.820	0.836	0.836
Light7	0.863	0.836	0.781	0.808	0.753	0.753	0.767	0.658	0.811	0.836	0.863	0.863	**0.877**	**0.877**
Mallat	**0.980**	0.979	0.943	0.973	0.964	0.921	0.950	0.942	0.976	0.972	0.962	0.973	0.973	0.973
MedImgs	0.792	0.772	0.740	0.767	0.742	0.741	0.755	0.712	**0.815**	0.779	0.801	0.763	0.768	0.768
MotStra	**0.950**	0.895	0.921	0.938	0.915	0.870	0.886	0.927	0.947	0.931	0.924	0.919	0.913	0.919
NonIThA	0.961	0.948	0.936	0.961	0.907	0.805	0.822	0.839	0.932	**0.967**	0.963	0.966	0.965	0.966
NonIThB	0.955	0.951	0.940	0.961	0.927	0.873	0.888	0.899	0.952	0.961	0.958	0.961	**0.965**	**0.965**
OliveO	0.833	0.867	0.867	0.767	0.900	0.867	0.867	0.900	0.898	0.733	**0.933**	0.900	**0.933**	**0.933**
OSLeaf	0.988	0.979	0.729	0.955	0.855	0.612	0.806	0.988	0.971	0.979	**1.000**	0.971	0.975	0.975
Sony_A	0.968	**0.985**	0.770	0.952	0.854	0.759	0.707	0.679	0.887	0.975	0.983	0.968	0.965	0.968
Sony_B	0.962	0.962	0.930	0.975	0.924	0.871	0.876	0.902	0.945	0.976	**0.982**	0.966	0.966	0.966
StarLCs	0.967	0.971	0.977	0.975	0.969	0.900	0.921	0.979	**0.982**	0.971	0.975	0.978	0.979	0.979
SwedL	0.966	0.958	0.934	0.973	0.954	0.848	0.915	0.728	0.969	0.976	**0.984**	0.982	0.973	0.982
Symbols	0.962	0.872	0.951	0.945	0.954	0.941	0.951	0.968	0.966	0.983	**0.986**	0.985	0.985	0.985
SynCntr	0.990	**1.000**	0.997	**1.000**	**1.000**	0.983	0.990	0.970	0.999	0.993	0.996	**1.000**	**1.000**	**1.000**
Trace	**1.000**	**1.000**	**1.000**	**1.000**	0.990	0.990	0.990	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
TwoLECG	0.897	**1.000**	0.999	0.999	0.985	0.885	**1.000**	0.996	0.994	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
TwoPats	**1.000**	**1.000**	0.998	**1.000**	**1.000**	0.999	0.933	0.984	**1.000**	0.995	0.991	**1.000**	0.999	**1.000**
uWavGX	0.754	0.787	0.820	0.790	0.804	0.778	0.801	0.759	**0.838**	0.837	0.825	0.837	0.837	0.837
uWavGY	0.725	0.668	0.732	0.713	0.733	0.702	0.717	0.687	**0.776**	0.752	0.760	0.761	0.765	0.765
uWavGZ	0.729	0.755	0.768	0.735	0.735	0.679	0.710	0.688	0.778	**0.783**	0.763	0.780	**0.783**	**0.783**
wafer	0.997	0.997	0.998	0.998	0.999	0.996	0.997	0.999	**1.000**	0.998	0.999	**1.000**	**1.000**	**1.000**
WordSy	0.580	0.632	0.724	0.679	0.734	0.749	**0.774**	0.655	0.748	0.671	0.738	0.702	0.718	0.718
yoga	0.902	0.858	0.888	0.871	0.887	0.847	0.879	**0.919**	0.917	0.911	0.910	0.913	0.906	0.913
Avg_acc	0.8753	0.8699	0.8651	0.8770	0.8748	0.8211	0.8457	0.8484	0.8935	0.8891	0.8976	0.8990	0.9024	0.9032
MPEC	0.0255	0.0267	0.0271	0.0247	0.0258	0.0348	0.0320	0.0297	0.0224	0.0227	0.0201	0.0206	0.0200	0.0197
Best	6	6	4	8	6	2	3	7	11	7	13	8	12	14
Rank	6.841	7.591	7.273	5.977	7.023	10.636	9.023	7.818	5.227	5.909	4.705	3.659	3.295	–

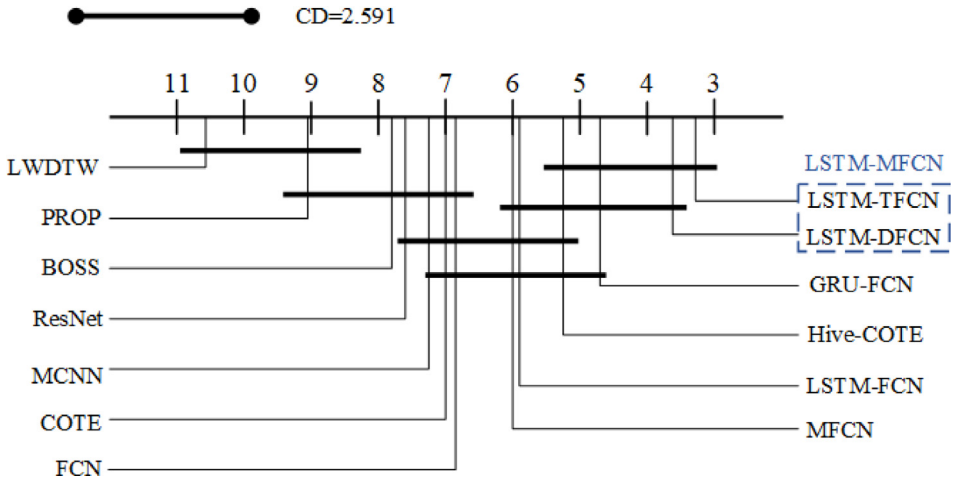

Fig. 3. Critical difference diagram of models involved in comparison.

Table 4
Structures and experiment results of the models involved in ablation experiment.

	Structure Hyperparameters			Experiment results
	Filter size	Respective fields	Scale proportions	
LSTM-FCN	8-5-3	8-5-3	-	0.8891
LSTM-DFCN(1)	(10,6)- (6,4)- 3	(10,6)- (6,4)- 3	1:3/2:2/3:1	0.8954
LSTM-DFCN(2)	(32,8)- (10,5)- 3	(32,8)- (10,5)- 3	1:3/2:2/3:1	0.8957
LSTM-DFCN	8-5-3	(32,8)- (10,5)- 3	1:3/2:2/3:1	0.8990

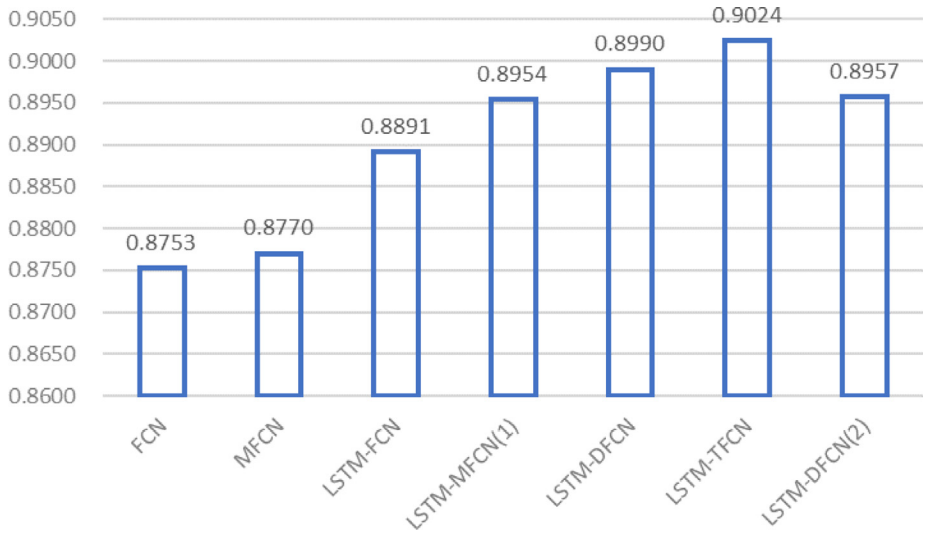

Fig. 4. Accuracies comparison of FCN related models or structures.

4.2. Ablation experiments

In this subsection, we launch ablation study around LSTM-DFCN architecture, so as to provide an interpretation about how our innovative contributions affect the performance. We set the same experimental hyperparameters and environment with above evaluation experiments and employ two similar variant structures (Table 4) of LSTM-DFCN implemented by traditional continuous convolution as contrasts. Performance of them will derive from the best results of twice experiments. They are:

- Structures with same amount of parameters with LSTM-DFCN, marked by LSTM-DFCN(1),
- Structures with same receptive fields with LSTM-DFCN, marked by LSTM-DFCN(2).

Table 4 reports the average accuracies of four networks involved in this ablative comparison. Observation reveals the two variant structures are both less effective than our LSTM-DFCN. For further analysis, we build three pair-to-pair comparisons. Firstly, with same parameter amounts, LSTM-DFCN(1) which utilizes multi-scale traditional convolution obtains a high accuracy than LSTM-FCN. A filter is able to detect a feature with breadth within its size, so the multi-scale filters make LSTM-DFCN(1) possible to perceive broad range spatial features without ignoring subtle ones. In a word, multi-scale can be regard as a reasonable reallocation to training attention, and thus benefit for learning the diverse features. Secondly, LSTM-DFCN outperforms LSTM-DFCN(1), for it has a dilated multi-scale of (32,8)-(10,5), while that in LSTM-DFCN(1) is only (10,6)-(6,4). It is no wonder that a big receptive field provides a potential to comprehensively exploring to features and then an access to promising performance. And the second comparison also illustrates that dilation convolution indeed fits to learn the successive time series data. Thirdly, LSTM-DFCN(2) is compared with LSTM-DFCN, its failure indicates that magnifying receptive fields merely by successive convolution cannot train the same effective model with that by dilation. A big filter implemented by successive convolution is accompanied with huge amounts of parameters, while dilation convolution is not. Meanwhile, a lot of datasets in UCR archive have small training set, so that a restricted tuning capacity. Thus, LSTM-DFCN(2) via a rough training yields disappointing results.

Above analyses interpret and confirm our innovative designs on the multi-scale perception of spatial features and the dilated amplification to filters. Then, we will give an overall comparison of FCN-based methods, contributions of these various evolvements can be revealed by the accuracy improvements shown in Fig. 4.

As we can see, FCN earns the worst accuracy because of its insufficient learning of spatial features; LSTM-FCN and MFCN amend FCN from different aspects, which join FCN with time dependencies or conduct a full perception to spatial patterns, both obtaining more accurate judgements. Then LSTM-DFCN(1) and LSTM-DFCN(2) are methods which combine the strengths of MFCN and LSTM-FCN together, delivering even better results. Here, our proposed representatives LSTM-DFCN and LSTM-TFCN import dilation convolution on this basis, providing a thorough and comprehensive learning to the spatial–temporal features in a variety of scales, leading to a state-of-the-art accuracies of 90.24% in average.

5. Conclusion

In TSC field, lots of approaches are dedicated to make further breakthrough on basis of Fully convolution networks, and the methods who learning spatial and temporal features simultaneously have shown the most significant improvements. In consideration of time series features exist in different size, a multi-modal neural network composed of multi-scale FCN and LSTM is proposed in this paper, where the large receptive fields are implemented by dilation convolution. Compared with the previous, our method offers a full and thorough learning to time series by virtue of large multi-scale spatial patterns and time-dependencies, so that instances can be better distinguished.

With two representative architectures, LSTM-MFCN is evaluated on 44 UCR datasets. Comparisons of experimental results illustrate that

it outperforms all the other baselines, including typical single-modal, multi-modal networks and notable non-deep-learning methods, which confirms the significance of fully grasping to spatial–temporal features. Ablation tests indicate that perceiving time series in multi-scale produces better acquisition to features than in fixed length and dilation generates a better set of multi-scale receptive fields than traditional successive convolution.

CRediT authorship contribution statement

Liang Zhao: Contributed to the idea and methodology. **Chunyang Mo:** Contributed to the methodology, Writing and experiments. **Jiajun Ma:** Contributed to the experiments, Writing. **Zhikui Chen:** Contributed to the proofing. **Chenhui Yao:** Contributed to the idea and proofing.

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 is supported by the National Natural Science Foundation of China (61906030), the Natural Science Foundation of Liaoning Province (2020-BS-063), the Fundamental Research Funds for the Central Universities (DUT20RC(4)009) and the Equipment Adavance Research Fund (80904010301).

References

[1] T. Rakthanmanon, B.J.L. Campana, A. Mueen, G.E.A.P.A. Batista, M.B. Westover, Q. Zhu, J. Zakaria, E.J. Keogh, Searching and mining trillions of time series subsequences under dynamic time warping, in: Q. Yang, D. Agarwal, J. Pei (Eds.), The 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '12, Beijing, China, August 12-16, 2012, ACM, 2012, pp. 262–270, http://dx.doi.org/10.1145/2339530.2339576.

[2] J. Lines, A.J. Bagnall, Time series classification with ensembles of elastic distance measures, Data Min. Knowl. Discov. 29 (3) (2015) 565–592, http://dx.doi.org/10.1007/s10618-014-0361-2.

[3] M.E. Karasikov, V.V. Strijov, Feature-based time-series classification, A.A.Dorodnicyn Comput. Center (2016) 121–131, http://dx.doi.org/10.14357/19922264160413.

[4] L. Ye, E.J. Keogh, Time series shapelets: a new primitive for data mining, in: J.F.E. IV, F. Fogelman-Soulié, P.A. Flach, M.J. Zaki (Eds.), Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009, ACM, 2009, pp. 947–956, http://dx.doi.org/10.1145/1557019.1557122.

[5] J. Lin, E.J. Keogh, L. Wei, S. Lonardi, Experiencing SAX: a novel symbolic representation of time series, Data Min. Knowl. Discov. 15 (2) (2007) 107–144, http://dx.doi.org/10.1007/s10618-007-0064-z.

[6] H.I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P. Muller, Deep learning for time series classification: a review, Data Min. Knowl. Discov. 33 (4) (2019) 917–963, http://dx.doi.org/10.1007/s10618-019-00619-1.

[7] X. Zou, Z. Wang, Q. Li, W. Sheng, Integration of residual network and convolutional neural network along with various activation functions and global pooling for time series classification, Neurocomputing 367 (2019) 39–45, http://dx.doi.org/10.1016/j.neucom.2019.08.023.

[8] A. Borovykh, S. Bohte, C.W. Oosterlee, Dilated convolutional neural networks for time series forecasting, J. Comput. Finance 22 (10) (2018) 73–101, http://dx.doi.org/10.21314/JCF.2019.358.

[9] W. Zhou, K. Hao, X. Tang, Y. Xiao, T. Wang, Time series classification based on FCN multi-scale feature eensemble learning, in: 2019 IEEE 8th Data Driven Control and Learning Systems Conference, DDCLS, 2019, pp. 901–906, http://dx.doi.org/10.1109/DDCLS.2019.8909079.

[10] Z. Cui, W. Chen, Y. Chen, Multi-scale convolutional neural networks for time series classification, 2016, arXiv:1603.06995, CoRR abs/1812.07683, URL http://arxiv.org/abs/1603.06995.

[11] N. Elsayed, A.S. Maida, M.A. Bayoumi, Deep gated recurrent and convolutional network hybrid model for univariate time series classification, 2018, arXiv:1812.07683, CoRR abs/1812.07683, URL http://arxiv.org/abs/1812.07683.

[12] F. Karim, S. Majumdar, H. Darabi, S. Chen, LSTM fully convolutional networks for time series classification, IEEE Access 6 (2018) 1662–1669, http://dx.doi.org/10.1109/ACCESS.2017.2779939.

[13] H.A. Dau, A.J. Bagnall, K. Kamgar, C.M. Yeh, Y. Zhu, S. Gharghabi, C.A. Ratanamahatana, E.J. Keogh, The UCR time series archive, 2018, arXiv:1810.07758, CoRR abs/1810.07758, URL http://arxiv.org/abs/1810.07758.

[14] Z. Wang, W. Yan, T. Oates, Time series classification from scratch with deep neural networks: A strong baseline, in: 2017 International Joint Conference on Neural Networks, IJCNN 2017, Anchorage, AK, USA, May 14–19, 2017, IEEE, 2017, pp. 1578–1585, http://dx.doi.org/10.1109/IJCNN.2017.7966039.

[15] A.J. Bagnall, J. Lines, J. Hills, A. Bostrom, Time-series classification with COTE: the collective of transformation-based ensembles, in: 32nd IEEE International Conference on Data Engineering, ICDE 2016, Helsinki, Finland, May 16–20, 2016, IEEE Computer Society, 2016, pp. 1548–1549, http://dx.doi.org/10.1109/ICDE.2016.7498418.

[16] B. Qian, Y. Xiao, Z. Zheng, M. Zhou, W. Zhuang, S. Li, Q. Ma, Dynamic multi-scale convolutional neural network for time series classification, IEEE Access 8 (2020) 109732–109746, http://dx.doi.org/10.1109/ACCESS.2020.3002095.

[17] H.I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P. Muller, Transfer learning for time series classification, in: IEEE International Conference on Big Data, Big Data 2018, Seattle, WA, USA, December 10–13, 2018, IEEE, 2018, pp. 1367–1376, http://dx.doi.org/10.1109/BigData.2018.8621990.

[18] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A.W. Senior, K. Kavukcuoglu, Wavenet: A generative model for raw audio, in: The 9th ISCA Speech Synthesis Workshop, Sunnyvale, CA, USA, 13-15 September 2016, ISCA, 2016, p. 125, URL http://www.isca-speech.org/archive/SSW_2016/abstracts/ssw9_DS-4_van_den_Oord.html.

[19] R. Xi, M. Li, M. Hou, M. Fu, H. Qu, D. Liu, C.R. Haruna, Deep dilation on multimodality time series for human activity recognition, IEEE Access 6 (2018) 53381–53396, http://dx.doi.org/10.1109/ACCESS.2018.2870841.

[20] M. Längkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognit. Lett. 42 (2014) 11–24, http://dx.doi.org/10.1016/j.patrec.2014.01.008.

[21] P. Malhotra, V. TV, L. Vig, P. Agarwal, G. Shroff, Timenet: Pre-trained deep recurrent neural network for time series classification, in: 25th European Symposium on Artificial Neural Networks, ESANN 2017,Bruges, Belgium, April 26-28, 2017, ESANN, 2017, pp. 1–12, URL http://www.elen.ucl.ac.be/Proceedings/esann/esannpdf/es2017-100.pdf.

[22] N. Mehdiyev, J. Lahann, A. Emrich, D. Enke, P. Fettke, P. Loos, Time series classification using deep learning for process planning: A case from the process industry, Procedia Comput. Sci. 114 (2017) 242–249, http://dx.doi.org/10.1016/j.procs.2017.09.066, URL https://www.sciencedirect.com/science/article/pii/S1877050917318707, Complex Adaptive Systems Conference with Theme: Engineering Cyber Physical Systems, CAS October 30 – November 1, 2017, Chicago, Illinois, USA.

[23] C. Yang, W. Jiang, Z. Guo, Time series data classification based on dual path CNN-RNN cascade network, IEEE Access 7 (2019) 155304–155312, http://dx.doi.org/10.1109/ACCESS.2019.2949287.

[24] G. Liu, X. Wang, R. Li, Multi-scale RCNN model for financial time-series classification, 2019, arXiv:1911.09359, CoRR abs/1911.09359, URL http://arxiv.org/abs/1911.09359.

[25] Q. Du, W. Gu, L. Zhang, S. Huang, Attention-based LSTM-CNNs for time-series classification, in: G.S. Ramachandran, B. Krishnamachari (Eds.), Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems, SenSys 2018, Shenzhen, China, November 4-7, 2018, ACM, 2018, pp. 410–411, http://dx.doi.org/10.1145/3274783.3275208.

[26] F. Karim, S. Majumdar, H. Darabi, Insights into LSTM fully convolutional networks for time series classification, IEEE Access 7 (2019) 67718–67725, http://dx.doi.org/10.1109/ACCESS.2019.2916828.

[27] P. Schäfer, The BOSS is concerned with time series classification in the presence of noise, Data Min. Knowl. Discov. 29 (6) (2015) 1505–1530, http://dx.doi.org/10.1007/s10618-014-0377-7.

[28] J. Lines, S. Taylor, A.J. Bagnall, Time series classification with HIVE-COTE: the hierarchical vote collective of transformation-based ensembles, ACM Trans. Knowl. Discov. Data 12 (5) (2018) 52:1–52:35, http://dx.doi.org/10.1145/3182382.

[29] J. Yuan, A.D. Chouakria, S.V. Yazdi, Z. Wang, A large margin time series nearest neighbour classification under locally weighted time warps, Knowl. Inf. Syst. 59 (1) (2019) 117–135, http://dx.doi.org/10.1007/s10115-018-1184-z.

[30] J. Demsar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30, URL http://jmlr.org/papers/v7/demsar06a.html.