Shapelet-Net: A Shapelet-Neural Network Approach for Multivariate Time Series Classification

Guozhong Li*, Byron Choi*, Jianliang Xu*, Sourav S Bhowmick†, Kwok-Pan Chun‡ and Grace L.H.Wong§
*Department of Computer Science, Hong Kong Baptist University, Hong Kong
† School of Computing Engineering, Nanyang Technological University, Singapore
‡ Department of Geography, Hong Kong Baptist University, Hong Kong
§Faculty of Medicine, The Chinese University of Hong Kong, Hong Kong
*{csgzli,bchoi,xujl}@comp.hkbu.edu.hk, †assourav@ntu.edu.sg, ‡kpchun@hkbu.edu.hk, §wonglaihung@cuhk.edu.hk

***Abstract*—Time series shapelets are discriminative subsequences that have been recently found not only *accurate* but also *interpretable* for the classification problem of univariate time series (UTS). However, existing works on shaplets selection cannot applied to multivariate time series classification (MTSC) since the candidate shaplets of MTSC may come from different variables with different lengths and we can not compare them. To address this challenge, in this paper, we propose a novel model called Shapelet-net, which can embed shapelet candidates from different original spaces into the same common space for shapelets selection. Then we compute representative and diversified final shapelets rather than directly using all the embeddings for model building. We have conducted experiments on Shapelet-Net with competitive state-of-the-art and benchmark methods using UEA MTS datasets. The results validate that the accuracy of Shapelet-Net is often ranked the $1st$ among all the compared methods. Furthermore, we illustrate its interpretability with two case studies.**

***Index Terms*—Multivariate time-series classification, Cluster-wise triplet loss, Unsupervised representation learning, Multivariate shapelet transformation**

I. Introduction

Multivariate time series (MTS), containing multiple observations at each timestamp, are ubiquitous in many applications, ranging from astronomy, biology, geoscience, smart cities, to health care, human action recognition, marketing, and other scientific and social domains. For example, data from EEG and MEG[1] are some standard multivariate data, which have a wide range of applications in medicine, neurology, and psychology. Multivariate time series classification (MTSC) has been one of the most fundamental tasks of MTS. However, MTSC has received much less research attention than its specific case, namely univariate time series classification (UTSC). Various methods [2] for UTSC have been proposed, and the accuracy has been significantly increased when compared to some benchmark methods, *e.g.*, 1 Nearest Neighbor (1-NN) with Euclidean distance (ED) or Dynamic Time Warping (DTW) [5].

Some related works for improving the MTSC accuracy are presented in Section V. Shapelets are discriminative time-series subsequences. The effectiveness of shapelet-based classifiers of UTSC has been proved by many related works in the last decade, *e.g.*, learning shapelets [14], logical shapelets [24], and fast shapelets [25]. Its efficiency has been recently significantly improved [20]. Importantly, shapelets themselves are intuitive, and the distances between shapelets and time series from different classes indicate a significant difference in the classes. To integrate shapelets with standard classifiers, such as SVM and Naive Bayes classifier, shapelet transformation [22] has been proposed.

Challenges. A shapelet-based approach for MTSC is, however, in its infancy. Few shapelet-based methods for MTSC are introduced. Thus, the performance in terms of accuracy and interpretability of shapelets for MTSC has not been demonstrated.

- First, multivariate time series obviously contain multiple variables, instead of only one variable as in univariate ones. Candidates of shapelets are often voluminous and heterogeneous. Previous methods [2][15] that exhaustively search shapelets can be inaccurate.
- Second, shapelet candidates of different variables can be of different lengths, and the qualities of shapelets are hence hard to compare. With excessive candidates, it is not clear how to select the discriminative subsequences as the final shapelets for classification.
- Third, most existing works are black-box approaches. Few methods provide interpretable results for understanding and explaining the classification. Hence, it is crucial the MTSC approach maintains the interpretability of shapelets.

Contributions. In this paper, we propose a new shapelet-neural network approach for the MTSC problem, called Shapelet-Net, for addressing the challenges mentioned above. An overview of Shapelet-Net is presented in Figure 1. The benefits are twofold, namely, *accuracy improvement* and *interpretable classification results*.

First, we propose *Multi-length-input dilated causal Convolutional Neural Network (Mdc-CNN)*, which enhances Dc-CNN [3], to embed shapelet candidates of different lengths and different variables into the same space (shapelet embedding). We propose a *cluster-wise triplet loss function* for training Mdc-CNN that considers the intra/inter cluster metric learning for accelerating convergence and improving stability. Our cluster-wise triplet loss not only considers multiple positive

[1]EEG and MEG stand for Electroencephalography and Magnetoencephalography, respectively.

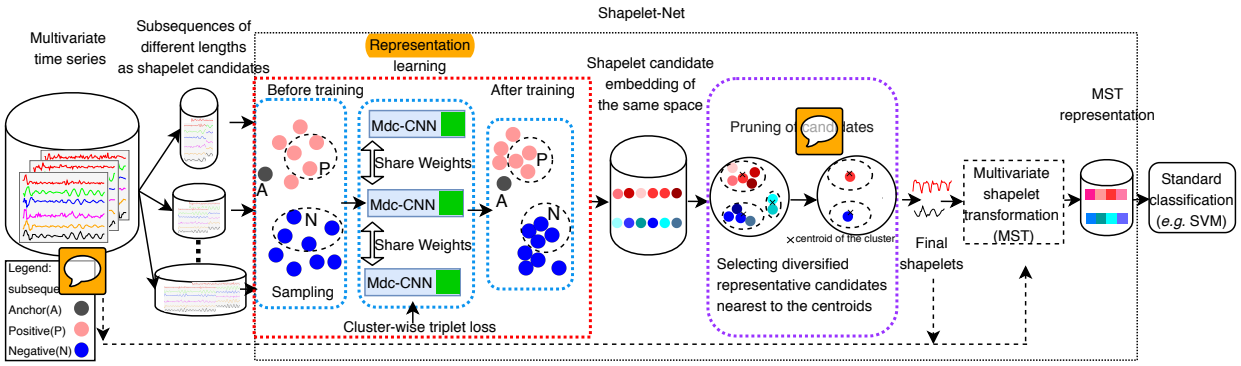


Fig. 1. Overview of Shapelet-Net for multivariate time series classification (MTSC)

samples and multiple negative samples, but also calculates the distance among the positives (negatives). The previous triplet loss [27] only proposes one positive sample and one negative sample. One positive and several negative samples without intra distance among negatives are introduced in word2vec [23] and time series representation learning [12]. We adopt *dilated convolution* that enables an exponentially large receptive field of the sequence for handling long-term dependencies without an explosion of model complexity. *Causal convolution* is adopted for convolving only the time before the current time, which ensures no future value impacts the current value.

Second, we avoid directly feeding numerous shapelet candidates (encoded by the representation learning part) for building a classifier. We cluster the shapelet candidates. We then propose a *utility function* for selecting top-k candidates that are close to the centroid of a large cluster and different from other cluster centroids, which give us representative and diversified final shapelets.

Then, we adopt *multivariate shapelet transformation* (MST) to determine the relationships among all variables. Specifically, we compute the distances from a multivariate time series to selected shapelet of the same variable. By multivariate shapelet transformation, each time series is embedded into the MST representation. Finally, we learn a classification model for the data in the MST representation. In this paper, we adopt linear SVM, which allows us to visualize how the shapelets separate the time series of different classes in the case studies.

We conduct experiments on UEA MTS Archive [1]. The results show that Shapelet-Net is ranked $1st$ among the baselines and the state-of-the-art methods in terms of accuracy. We note that Shapelet-Net performs the best in 20 datasets out of 30 datasets. We present two cases about human action recognition and ECG data, to illustrate how do the shapelets offer insights to classification.

Organization. The rest of this paper is organized as follows. Section II presents some preliminaries and the problem statement. The details of our proposed method are given in Section III. Section IV reports the experimental results. Section V reviews the related work. Section VI concludes the paper and presents avenues for future work.

TABLE I
SUMMARY OF FREQUENTLY USED NOTATIONS

Notation	Meaning
T	a time series $(t_1, t_2, \cdots, t_i, \cdots, t_N)$, where t_i is the i-th value in T and N is the length of T
D	a time series dataset $(T_1, T_2, \cdots, T_M)$, where M is the number of time series in D
$T_{a,b}$	a subsequence $T_{a,b}$ of T, $(t_a, \cdots, t_b)$, where $1 \leq a \leq b \leq N$, a and b, the beginning and ending positions
$\mathcal{C}$	the label set
D_C	C is the label in $\mathcal{C}$; and for all $T \in D_C$, the label of T is C
V	the number of variables/observations
$\mathbb{T}$	a MTS $\mathbb{T} = (T^1, T^2, \cdots, T^v \cdots, T^V)$, where $T^v = (t_1^v, t_2^v, \cdots, t_i^v, \cdots, t_N^v)$
$\mathbb{D}$	a MTS dataset $(\mathbb{T}_1, \mathbb{T}_2, \cdots, \mathbb{T}_M)$, where M is the number of MTS in $\mathbb{D}$
$\mathcal{S}$	a set of shapelets

II. PRELIMINARIES

In this section, we present some preliminaries and problem statement. We summarize the notations and their meanings in Table I.

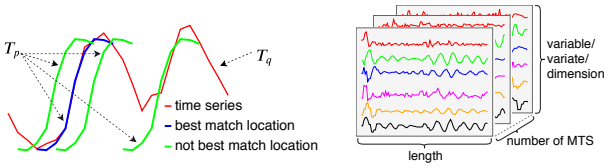
Definition 1: **Distance between two sequences** [10]. The distance of the sequence T_p of the length $|T_p|$ and T_q of the length $|T_q|$ is denoted as (w.l.o.g. assuming $|T_q| \geq |T_p|$),

$$\mathsf{dist}(T_p, T_q) = \min_{j=1,\cdots,|T_q|-|T_p|+1} \frac{1}{|T_p|} \sum_{l=1}^{|T_p|} (tq_{j+l-1} - tp_l)^2, \tag{1}$$

where tq_i and tp_i are the i-th value of T_p and T_q, respectively. □

Intuitively, dist is the distance of the shorter sequence T_p to the most similar subsequence in T_q, as shown in Figure 2(a).

Definition 2: **Shapelet** S [30]. A shapelet S of the length $|S|$ of class C_j, where $C_j \in \mathcal{C}$, is a time series subsequence, which represents class C_j and discriminate C_j from other classes, *i.e.*, $\mathcal{C} \setminus \{C_j\}$. That is, for all T_j having the label C_j, dist(T_j,



(a) The best match location (b) A MTS dataset — extracted from Basicmotions [1]

Fig. 2. (a) An illustration of the best match location of a subsequence in a time series; (b) An illustration of a multivariate time series dataset — extracted from Basicmotions [1]

S) is smaller than $\mathsf{dist}(T_k, S)$, where T_k is time series having a label in $\mathcal{C} \setminus \{C_j\}$. □

According to Eq. 1, the distance between the j-th time series T_j and a shapelet candidate S_i of the length $|S_i|$ is defined as follows:

$$d_{j,i} = \mathsf{dist}(T_j, S_i) \quad (2)$$

Shapelets transform a time series T_j into a representation in new data space $(d_{j,1}, \cdots, d_{j,|\mathcal{S}|})$ by calculating the distances with a set of shapelets $\mathcal{S}$, where $d_{j,i} = \mathsf{dist}(T_j, S_i)$ and $S_i \in \mathcal{S}$. After the transformation, the number of data values of the dataset D is reduced from $M \times N$ to $M \times |\mathcal{S}|$, where $|\mathcal{S}|$ is often much smaller than N.

The apparent difference between UTS and MTS is that there are multiple observations at each timestamp in MTS, which makes the classification problem more difficult. An example of the MTS dataset, called Basicmotions from [1], is shown in Figure 2(b).

Example 1: In Figure 2(b), six variables ($V = 6$) are recorded by an accelerometer and a gyroscope. In the dataset, there are four classes, *i.e.*, $|\mathcal{C}| = 4$, namely standing, walking, running, and playing badminton. Each class is accompanied with 10 training cases (instances) and 10 test cases. Thus, the overall number M of instances is 80. The length (N) of each time series is 100. It is ineffective and inefficient to compute all the distances between each time series and numerous subsequences in the dataset. □

Problem statement. Given a multivariate time series dataset $\mathbb{D}$, consisting of M multivariate time series instances $\mathbb{T}_1, \mathbb{T}_2, \cdots, \mathbb{T}_M$ with V variables, this paper investigates a shapelet-based classifier. □

III. SHAPELET-NET

In this section, we present the details of Shapelet-Net for MTSC. In particular, we present multi-length-input dialted casual CNN, the cluster-wise triple loss function, triplet sampling and multivariate shapelet transformation.

A. Multi-length-input Dilated Causal CNN (Mdc-CNN)

~~In this paper,~~ shapelet candidates of different lengths are generated by using different sliding windows (the data shown in the cylinders of Figure 1). Our target is to embed all the ~~time series subsequences of various lengths, which are~~ shapelet candidates, from the original space into a new common space.

Background. Shapelet-Net adopts a few existing works as the building blocks. First, the dilated causal convolutional neural network (Dc-CNN) [29] is employed to learn a new representation of time series subsequences. The effectiveness of the dilated causal network has been proved for sequence modeling tasks by Bai et al. [3]. The causal convolution is designed such that the future data do not impact the learning from the past data. The dilated convolution is utilized to modify the receptive field of the convolution.

Second, although the output of Dc-CNN can be of the same length as the input, it cannot handle inputs of various lengths. Thus, we propose to introduce a global max pooling layer and a linear layer, which are stacked on top of the last Dc-CNN layer, to embed all shapelet candidates into the same space (indicated by the green boxes in Figure 1). We call it *Multi-length-input Dilated Causal CNN (Mdc-CNN)*.

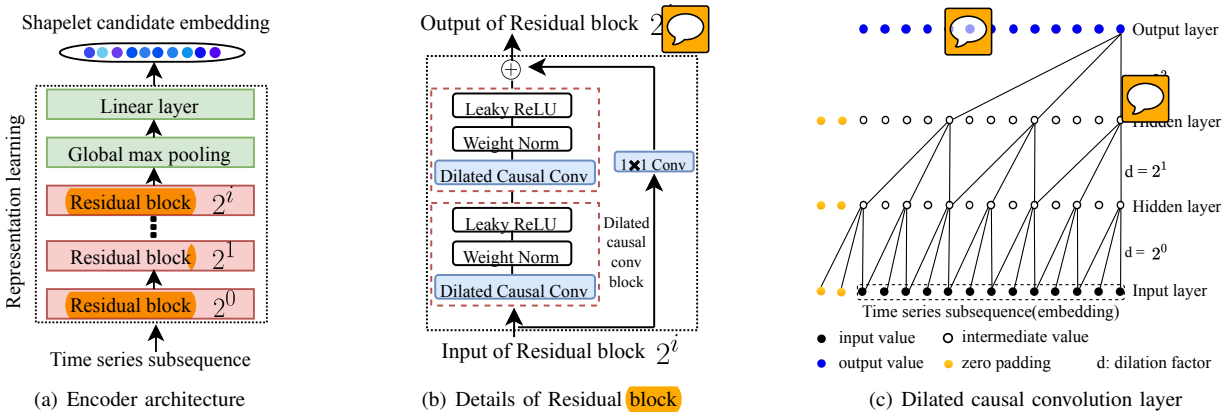
Mdc-CNN architecture. Mdc-CNN is further elaborated with Figure 3. Figure 3(a) shows that the encoder has $i+1$ layers of residual blocks, where i is the dilation factor, and the global max pooling layer and linear layer are stacked on top of the residual blocks. The input of the encoder is the time series subsequences of various lengths and variables, and the output is their unified representation. We call them shapelet candidate embedding. Figure 3(b) presents the residual block having two identical subblocks, dilated causal convolution block. Figure 3(c) presents a dilated causal convolution example with dilation factor $d = 2^0, 2^1, 2^2$. Further details of Figure 3(b) and Figure 3(c) can be found in [3].

Following the standard practice, Mdc-CNNs use shared weights for training models of shapelet candidates of different lengths and variables, as shown in Figure 1. The learned embedding of the time series in the new space (denoted as $f(\cdot)$) is expected to distance-preserving.

B. Unsupervised Representation Learning

We next explain how the Mdc-CNN networks are trained in an unsupervised manner. There have been several existing loss functions for unsupervised learning, such as word2vec [23], image similarity [8], and face recognition [27]. In [8] and [27], only one positive sample and one negative sample are considered, whereas, in [12] and [23], one positive and several negative samples are considered. We recall that Franceschi et al. [12] follow the principle from word2vec [23], which makes the assumption that the representation of a word should meet two requirements. In particular, (i) the representation should be close to the ones near its context [13], and (ii) it should be distant from the ones in a randomly chosen context, since they are probably different from the original word's context.

The objectives in our paper are analogous to word2vec, which is likewise to ensure that similar time series obtain similar representations and vice versa. However, ① the second requirement of the word2vec's assumption does not always hold in the context of time series. For example, one variable of the walking class in the Basicmotions dataset is shown in Figure 4. We can easily observe that some crests of the waveform are far away but not distant from each other. ②



(a) Encoder architecture (b) Details of Residual block (c) Dilated causal convolution layer

Fig. 3. An elaboration of the Multi-length-input dilated causal Convolutional Neural Network (Mdc-CNN)

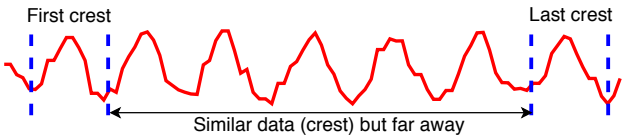


Fig. 4. An example from Basicmotions of violating the second requirement of word2vec: subsequences that are far away but have a small distance between them

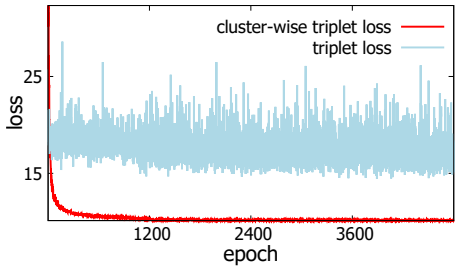


Fig. 5. A loss comparison between our cluster-wise triplet loss (multiple positives and multiple negatives, both with intra distance) and original triplet loss (one positive and multiple negatives without intra distance) on ArticularyWordRecognition [1]

Only one positive sample is utilized to train the network one batch, which increase the unstability of network training. ③ The distance among the negative (positive) samples is not considered. Figure 5 shows the loss comparison between our cluster-wise triplet loss and the original triplet loss [12]. Although, the original loss has a slight decline, we can precisely find that the network converges faster and stabler under our loss.

Cluster-wise triplet loss function. In this work, we propose our loss function takes *multiple positive and negative samples* and *the distance among positives (negatives)* into account. Specifically, the set of all possible triplets in the training set $\mathcal{T}$ is defined as follows:

$$\left(x, \boldsymbol{x}^+, \boldsymbol{x}^-\right) \in \mathcal{T},$$

where x is the anchor shapelet candidate, $\boldsymbol{x}^+$ and $\boldsymbol{x}^-$ denote the set of positive and negative samples with the sizes K^+ and K^-, respectively. The procedure of selecting triplets is presented in Section III-C.

First, denote the normalized distance of the positive (respectively, negative) samples from the anchor as $\mathcal{D}_{AP}$ (respectively, $\mathcal{D}_{AN}$). Hence, we have the following formula:

$$\mathcal{D}_{AP} + \mu < \mathcal{D}_{AN}, \tag{3}$$

where μ is a margin that is enforced between positive and negative samples. Suppose squared Euclidean distance is adopted. $\mathcal{D}_{AP}$ and $\mathcal{D}_{AN}$ can then be defined as follows.

$$\mathcal{D}_{AP} = \frac{1}{K^+} \sum_{i=1}^{K^+} ||f(x) - f(x_i^+)||_2^2 \tag{4}$$

and

$$\mathcal{D}_{AN} = \frac{1}{K^-} \sum_{i=1}^{K^-} ||f(x) - f(x_i^-)||_2^2, \tag{5}$$

where $f(\cdot) \in \mathbb{R}^z$ is the representation embedded by Mdc-CNN, and z is the length of the embedding.

In addition to the distances between the anchor and the positive samples (respectively, negative samples), the distances among the positive samples (respectively, negative samples) are included and should be small (respectively, large). The maximum distance among all positive (respectively, negative) samples is presented in Eq. 6 (respectively, Eq. 7).

$$\mathcal{D}_{pos} = \max_{i,j \in (1,K^+) \wedge i<j} \{||f(x_i^+) - f(x_j^+)||_2^2\} \tag{6}$$

and

$$\mathcal{D}_{neg} = \max_{i,j \in (1,K^-) \wedge i<j} \{||f(x_i^-) - f(x_j^-)||_2^2\} \tag{7}$$

The intra-sample loss is defined as follows:

$$\mathcal{D}_{intra} = \mathcal{D}_{pos} + \mathcal{D}_{neg} \tag{8}$$

Putting these together, we propose the *cluster-wise triplet loss function for the triplets* for our model in Eq. 9, to train the network under an unsupervised fashion.

$$\mathcal{L}\left(f(x), f(\boldsymbol{x}^+), f(\boldsymbol{x}^-)\right) = \log \frac{\mathcal{D}_{AP} + \mu}{\mathcal{D}_{AN}} + \lambda \mathcal{D}_{intra} \tag{9}$$

where λ is a hyper-parameter.

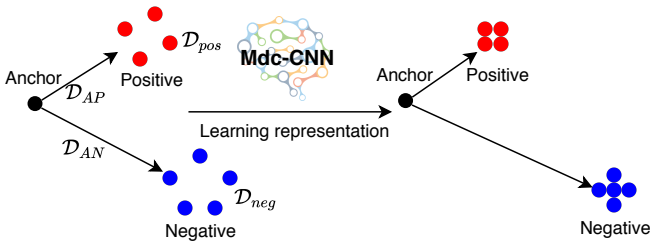


Fig. 6. Illustration of the effect of training a model using the cluster-wise triplet loss function

Example 2: An illustration of Eq. 9 is shown in Figure 6. The triplet loss function both minimizes the distance between the anchor and all positive samples, and the distance among all positive (negative) samples, and maximizes the distance between the anchor (positive) and all negatives. □

Differentiation of the loss function. In order to compute the derivative of Eq. 9, all the involved functions of the model should be differentiable. Unfortunately, the maximum function of Eq. 6 and Eq. 7 are not continuous and differentiable. We then introduce a differentiable approximation to the maximum function [7].

For the sake of organizational clarity, we use $\mathcal{D}^+_{i,j}$ and $\mathcal{D}^-_{i,j}$ to represent $||f(x^+_i) - f(x^+_j)||^2_2$ and $||f(x^-_i) - f(x^-_j)||^2_2$, respectively.

$$\mathcal{D}_{pos} \approx \tilde{\mathcal{D}}_{pos} = \frac{\sum_{i=1}^{K^+}\sum_{j=1}^{K^+} \mathcal{D}^+_{i,j} \cdot e^{\alpha \cdot \mathcal{D}^+_{i,j}}}{\sum_{i=1}^{K^+}\sum_{j=1}^{K^+} e^{\alpha \cdot \mathcal{D}^+_{i,j}}} \tag{10}$$

and

$$\mathcal{D}_{neg} \approx \tilde{\mathcal{D}}_{neg} = \frac{\sum_{i=1}^{K^-}\sum_{j=1}^{K^-} \mathcal{D}^-_{i,j} \cdot e^{\alpha \cdot \mathcal{D}^-_{i,j}}}{\sum_{i=1}^{K^-}\sum_{j=1}^{K^-} e^{\alpha \cdot \mathcal{D}^-_{i,j}}}, \tag{11}$$

where $\alpha > 0$ in Eq. 10 and Eq. 11 yields a smooth maximum approximation.

The gradients of overall maximum distance are presented in Eq. 12 and Eq. 13.

$$\frac{\partial \tilde{\mathcal{D}}_{pos}}{\partial \mathcal{D}^+_{i,j}} = \frac{e^{\alpha \cdot \mathcal{D}^+_{i,j}}(1 + \alpha(\mathcal{D}^+_{i,j} - \tilde{\mathcal{D}}_{pos}))}{\sum_{i=1}^{K^+}\sum_{j=1}^{K^+} e^{\alpha \cdot \mathcal{D}^+_{i,j}}} \tag{12}$$

$$\frac{\partial \tilde{\mathcal{D}}_{neg}}{\partial \mathcal{D}^-_{i,j}} = \frac{e^{\alpha \cdot \mathcal{D}^-_{i,j}}(1 + \alpha(\mathcal{D}^-_{i,j} - \tilde{\mathcal{D}}_{neg}))}{\sum_{i=1}^{K^-}\sum_{j=1}^{K^-} e^{\alpha \cdot \mathcal{D}^-_{i,j}}} \tag{13}$$

Thus, the gradients of Eq. 9 with respect to $f(x), f(x_i{}^+), f(x_i{}^-)$ are as follows:

$$\frac{\partial \mathcal{L}}{\partial f(x)} = \frac{2\sum_{i=1}^{K^+} ||f(x) - f(x^+_i)||_2}{\sum_{i=1}^{K^+} ||f(x) - f(x^+_i)||^2_2} - \frac{2\sum_{i=1}^{K^-} ||f(x) - f(x^-_i)||_2}{\sum_{i=1}^{K^-} ||f(x) - f(x^-_i)||^2_2} \tag{14}$$

Algorithm 1: Shapelet candidates generation

Input: MTS dataset $\mathbb{D} = \mathbb{T}^{M \times V \times N}$, sliding window size ϕ
Output: Shapelet candidates Ω

```
for l in φ do
    Initialize Ω_l = ∅ ;
    for m = {1, 2, ··· , M} do
        for v = {1, 2, ··· , V} do
            for i = {1, 2, ··· , N − l + 1 } do
                e = T^v(i, i + l − 1) ;
                Ω_l = Ω_l + e ;
return Ω
```

$$\frac{\partial \mathcal{L}}{\partial f(x_i{}^+)} = \frac{2\sum_{i=1}^{K^+} ||f(x^+_i) - f(x)||_2}{\sum_{i=1}^{K^+} ||f(x) - f(x^+_i)||^2_2} + \sum_{j=1}^{K^+} \frac{\partial \tilde{\mathcal{D}}_{pos}}{\partial \mathcal{D}^+_{i,j}} \cdot 4||f(x^+_i) - f(x^+_j)||_2 \tag{15}$$

$$\frac{\partial \mathcal{L}}{\partial f(x_i{}^-)} = \frac{2\sum_{i=1}^{K^-} ||f(x) - f(x^-_i)||_2}{\sum_{i=1}^{K^-} ||f(x) - f(x^-_i)||^2_2} + \sum_{j=1}^{K^-} \frac{\partial \tilde{\mathcal{D}}_{neg}}{\partial \mathcal{D}^-_{i,j}} \cdot 4||f(x^-_i) - f(x^-_j)||_2 \tag{16}$$

Since Eq. 9 is differentiable, we use back propagation over the entire neural network based upon minibatch stochastic gradient descent together with Adam optimizer [19] to optimize our Shapelet-Net parameters.

Discussions. There are at least two advantages of our cluster-wise triplet loss function over the previous ones [12]. On the one hand, it accelerates convergence and improves stability through not only selecting both multiple positive and negative shapelet candidates but also minimizing the distance among positive/negative shapelet candidates, which are not considered before. On the other hand, an important property of shapelets is depicted by Eq. 3. That is, the shapelet is a subsequence of a time series (the anchor) that most of the time series in one class (positives) are close to it, while most of the time series from other classes (negatives) are far away from it.

C. Data preprocessing

In this subsection, we present the details for preparing the multivariate time series data for Mdc-CNN. Specifically, we present the shapelet candidate generation and the triplets selection for the training.

Shapelet candidates generation. We apply sliding windows of different sizes to generate abundant shapelet candidates from the original multivariate time series. A variate label is then annotated to each candidate, for shapelet transformation in Section III-D. Thus, there are two labels for each shapelet candidate, one for the variable and the other for the class of the time series. The shapelet candidates generation procedure is summarized in Algo. 1.

Triplet selection. The numbers of triplets of some real-world datasets are large, and it is computationally prohibitive and sub-optimal to use all the triplets for training. Instead, we conduct triplet sampling.

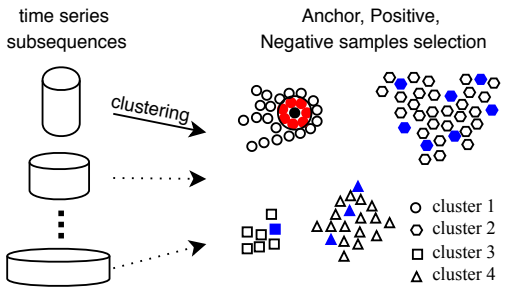


Fig. 7. An illustration of triplet sampling, black for Anchor, red for Positive samples, blue for Negative samples [best viewed in color]

Specifically, we construct the triplet tuple $\left(x,\ (x_i^+)_{i\in[1,K^+]},\ (x_i^-)_{i\in[1,K^-]}\right)$ from shapelet candidates at the beginning of each iteration as follows. **I.** We do the clustering on the candidates by kmeans [17] to obtain Y clusters. We randomly select one candidate as the anchor sample x. **II.** Then, from the same cluster, top K^+ other shapelet candidates nearest to the anchor are chosen as positive samples $(x_i^+)_{i\in[1,K^+]}$. **III.** For the negative samples $(x_i^-)_{i\in[1,K^-]}$, we randomly pick candidates from other clusters in proportion. The generalization of this procedure to mini-batch training is straightforward, and thus we omit the details. An example of the triplet selection process is shown in Figure 7. Algo. 2 shows the pseudo-code of triplet selection.

Example 3: After the candidates' generation, we have some candidates with different lengths (the leftmost part of Figure 7). For each length, we use a clustering technique to divide them into several groups (4 groups in this example). The following processes are the same for each cluster, and we take cluster 1 as an illustration in Figure 7. We arbitrarily choose one time series subsequence (shapelet candidate) as the anchor (the solid black circle). We obtain 8 closest candidates as positive samples (the solid red circle). Then, negative samples are selected from other clusters in proportion, specifically, $7, 1, 3$ negative samples (the blue solid hexagon, square, triangle) in these three clusters, respectively. □

Algo. 2 shows the pseudo-code of triplet selection. In Line 1, the shapelet candidates are clustered. This avoids selecting positive (respectively negative) samples that are highly similar to each other, which gives little information for training. The three parts of Algo. 2 are corresponding to the selection of anchor (Lines 4−5), and positive and negative samples (Lines 7−11 and Lines 13−18, respectively).

After we train the network with the triplet samples into new representations using the cluster-wise triplet loss function, we use the network to embed all the other shapelet candidates.

Analysis. We can find that all elements, namely the anchor, positives, and negatives in the tuple are selected at each batch. As the training goes on, our network minimizes the loss of a batch of triplet tuples at each epoch, which can be approximately regarded as minimizing all the triplets. Thus, although we do not consider the loss of all triplets, the effectiveness of our network can still be improved through sampling.

Algorithm 2: Selection of APN

Input: Shapelet candidates Ω
Output: $(x, \boldsymbol{x}^+, \boldsymbol{x}^-)$

1 $\bigcup_{i=1}^{Y} \Omega^i \leftarrow \mathsf{kmeans}(\Omega)$;
2 **for** $i = \{1, 2, \cdots, Y\}$ **do** `// for each cluster`
3 {Anchor selection}
4 $x \leftarrow \Omega^i.\mathsf{Random}()$;
5 $\Omega^i = \Omega^i \setminus \{x\}$;
6 {Positive selection}
7 **for** $k = \{1, 2, \cdots, K^+\}$ **do**
8 $x^+ = \Omega^i.\mathsf{Top}(x)$;
9 $\Omega^i = \Omega^i \setminus \{x^+\}$;
10 $\boldsymbol{x}^+ = \boldsymbol{x}^+ \cup \{x^+\}$;
11 k++ ;
12 {Negative selection in proportion}
13 **for** $j = \{1, 2, \cdots, Y\} \setminus \{i\}$ **do**
14 **for** $k = \{1, 2, \cdots, \left\lceil \frac{K^-}{Y-1} \right\rceil\}$ **do**
15 $x^- = \Omega^j.\mathsf{Random}()$;
16 $\Omega^i = \Omega^i \setminus \{x^-\}$;
17 $\boldsymbol{x}^- = \boldsymbol{x}^- \cup \{x^-\}$;
18 k++ ;
19 **return** $(x, \boldsymbol{x}^+, \boldsymbol{x}^-)$

D. Multivariate Shapelet Transformation

After determining the new representation of shapelet candidates, we propose to select high-quality, diversified candidates as final shapelets. Finally, we adopt the procedure of shapelet transformation for MTS, then adopt a standard classifier.

Determining final shapelets. By following previous subsections, all the candidates are embedded into the same space. Next, a clustering method (*e.g.*, kmeans) is employed to yield Y clusters of the shapelet candidates. Then, we propose a utility (Eq. 17) to rank the candidates that are nearest to the cluster centroids. The first component of Eq. 17 is the size of the candidate's cluster. A large cluster means that it represents many candidates. The second component is the candidate's distance to other candidates in other clusters. A large distance indicates that the candidate is different from others.

$$\mathcal{U}(f(x_i)) = \beta \cdot \frac{\log(\mathsf{size}(f(x_i)))}{\log(\max_{i=1}^{Y}(\mathsf{size}(f(x_i))))} + (1-\beta)\frac{\log \sum_{j=1}^{Y} ||f(x_i) - f(x_j)||_2^2}{\log(\max_{i=1}^{Y}(\sum_{j=1}^{Y} ||f(x_i) - f(x_j)||_2^2))} \tag{17}$$

where $\beta \in [0, 1]$.

We select the top-k candidates according to Eq. 17 and retrieve the original time series subsequences which represent as the *final shapelets*, denoted as $\mathcal{S}_k$.

Multivariate Shapelet Transformation (MST). The multivariate shapelet transformation is first mentioned in [6] and the following is our formal definition.

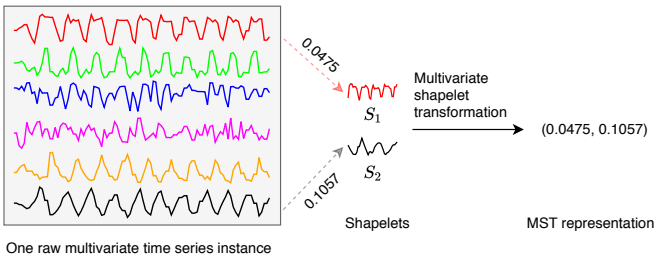


Fig. 8. Illustration of transforming an MTS instance into the MST representation

Definition 3: **Multivariate shapelet transformation**. Multivariate shapelet transformation is a method to transform a multivariate time series $\mathbb{T}_m$ into a new data space ($d_{m,1}$, $d_{m,2}$, $\cdots$, $d_{m,k}$) by calculating the distances with a set of final shapelets $\mathcal{S}_k$, denoted as $d_{m,j} = \mathsf{dist}(T_m^v, S_j)$, where $S_j \in \mathcal{S}_k$, and the variable of S_j and T_m^v is the same. □

Example 4: An example of multivariate shapelet transformation is shown in Figure 8. The leftmost plot exhibits an instance with 6 variables from the Basicmotions dataset. Two shapelets S_1 and S_2 are in the middle. For multivariate shapelet transformation, we only calculate the distance between the time series subsequence with the same variable (*e.g.*, the distance between the first variable (red) and S_1). Thus, the MST representation of the raw time series instance is displayed in the rightmost part. □

Algo. 3 is the pseudo-code of multivariate shapelet transformation from final shapelet discovery (Lines 3−15), and shapelet transformation for the original dataset (Lines 17−23).

First, we do the clustering among all the new representations of the shapelet candidates $f(\Omega)$ using a standard clustering algorithm (Line 3), We use $(f(\Omega))^i$ to denote the ith cluster. Then, the nearest one to the centroid in each cluster is added to the set $f(\mathcal{S})$. In Lines 13−14, the utility of each candidate (Eq. 17) is used to select the top-k candidates. We trace $f(\mathcal{S})$ to the original dataset to retrieve their corresponding original subsequences (Line 15). They form the final shapelets $\mathcal{S}_k$ for the dataset, where $|\mathcal{S}_k| = k$.

Finally, the transformation computes the distance when the shapelet and the variate of the time series have the same variable (Line 19). The distance between them is calculated by Eq. 1 (Line 20). After the calculation between one instance in the original time series dataset and all the shapelets, the MST representation of the instance is denoted as $\tilde{\mathbb{T}}_m$ (Line 22). Then, the MST representation of the dataset is $\tilde{\mathbb{D}} = \tilde{\mathbb{T}}^{M\times k}$.

After multivariate shapelet transformation, the dataset $\mathbb{D}$ is reduced from $M \times V \times N$ to $M \times k$, where $|\mathcal{S}_k| = k$ and k is significantly smaller than $V \times N$. In all, we summarize the final shapelet discovery and multivariate shapelet transformation in Algo. 3.

When the transformation of all the MTS instances are done, some standard classifiers (*e.g.*, SVM) are exploited to learn a classification model from the transformed representation. In this paper, we adopt SVM with a linear kernel.

Algorithm 3: Shapelet Transformation of MTS

Input: MTS dataset $\mathbb{D} = \mathbb{T}^{M\times V\times N}$, $f(\Omega)$, k
Output: Shapelets $\mathcal{S}_k$

1 Initialize the priority queue $f(\mathcal{S}) = \emptyset, \mathcal{S} = \emptyset$;
2 {Shapelet discovery}
3 $\bigcup_{i=1}^{Y} (f(\Omega))^i \leftarrow \mathsf{kmeans}(f(\Omega))$;
4 **for** $i = \{1, 2, \cdots, Y\}$ **do** // each cluster
5 $min = +\infty$;
6 $f(\mathcal{S}).\mathsf{push}(f(min))$;
7 **foreach** $f(e) \in (f(\Omega))^i$ **do**
8 $tmp = ||(f(\Omega))^i.\mathsf{centroid} - f(e)||_2^2$;
9 **if** $tmp < min$ **then**
10 $min = tmp$;
11 $f(\mathcal{S}).\mathsf{pop}()$;
12 $f(\mathcal{S}).\mathsf{push}(f(e))$;
13 Calculate $\mathcal{U}$ (Eq. 17) for each candidate in $f(\mathcal{S})$;
14 Sort each candidate based on $\mathcal{U}$ and select top-k candidates, denoted as $f(\mathcal{S}_k)$;
15 Retrieve $\mathcal{S}_k$ of $f(\mathcal{S}_k)$ from $\mathbb{D}$;
16 {Shapelet transformation}
17 **for** $m = \{1, 2, \cdots, M\}$ **do**
18 **for** $j = \{1, 2, \cdots, k\}$ **do**
19 $v = S_j.variable$;
20 $d_{m,j} = \mathsf{dist}(\mathbb{T}_m^v, S_j)$;
21 $\tilde{\mathbb{T}}_m.\mathsf{append}(d_{m,j})$;
22 $\tilde{\mathbb{T}}_m =< d_{m,1}, d_{m,2}, \cdots, d_{m,k} >$;
23 $\tilde{\mathbb{D}} = \tilde{\mathbb{T}}^{M\times k}$;
24 **return** $\mathcal{S}_k$

IV. Experiments

A. Environment

We implemented the proposed method [2] in Python. All the experiments were conducted on a machine with two Xeon E5-2630v3 @ 2.4GHz (2S/8C) / 128GB RAM / 64 GB SWAP and two NVIDIA Tesla K80, running on CentOS 7.3 (64-bit).

B. Datasets and parameters

A well-known benchmark of MTS datasets, namely UEA Archive, was tested. The detailed information of the datasets can be obtained from [1].

The following are some parameters used in our experiment. μ in Eq. 3 is set to 0.2, $\lambda = 1$ for the triplet loss function. The learning rate is kept fixed at a small value of $\eta = 0.001$, while the number of iterations for network training is 400. The α in Eq. 10 and Eq. 11 is set to 100. The β in Eq. 17 is set to 0.5.

C. Convergence of Mdc-CNN

We verify the convergence of Mdc-CNN depending on the parameters from Section IV-B. For instance, the convergence of the learning algorithm on four datasets, AtrialFibrillation, Basicmotions, StandWalkJump, UWaveGestureLibrary, is illustrated in Figure 9.

All the losses converge very smoothly for the training process among all four datasets. We can also observe that the loss converges fast at the beginning, then stabilizes. Similar

[2]To promote reproducibility, our source code will be made public.

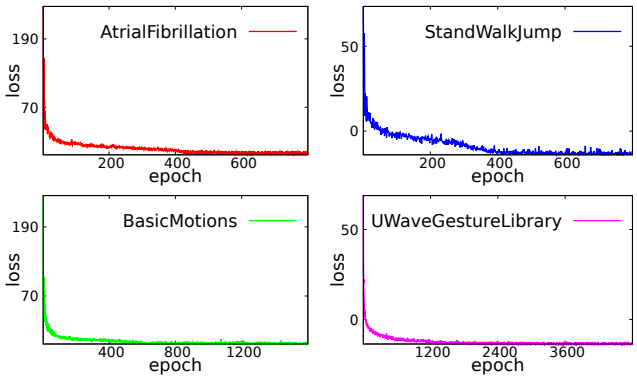


Fig. 9. Convergence of the learning algorithm on some MTS datasets

trends can be observed from the rest of the datasets. This demonstrates the effectiveness of our cluster-wise triplet loss.

D. Baselines

We compared Shapelet-Net with five different methods. Due to space restrictions, we provide brief details of each method. Interested readers may refer to the original paper for details. TapNet [31] can significantly outperform some existing methods like SMTS [4], LSTM-FCN [18], WEASEL-MUSE [26]. Thus, we only compare TapNet with our Shapelet-Net in the following part.

- **Three benchmarks [1].** Three benchmark classifiers (EDI, $DTWI$, and $DTWD$) are based on Euclidean Distance (EDI), dimension-independent dynamic time warping ($DTWI$), and dimension-dependent dynamic time warping ($DTWD$) [28].
- **Negative samples (NS) [12].** This method applies several negative samples when training their neural network, then SVM is utilized to do the final classification.
- **TapNet [31].** TapNet is a novel MTSC model with an attentional prototype network to take the strengths of both traditional and deep learning based approaches.

E. Experiments on accuracy

1) Comparison with other methods: The experiment accuracy results of the baselines are all taken from the original papers [1], [12] and [31], respectively. We only consider the normalized datasets for the experiment. That is, we did not take the accuracy of unnormalized datasets from [1]. The overall classification accuracy results for the datasets are presented in Table II.

From Table II, we can observe that the overall accuracies of Shapelet-Net is ranked $1st$ among all the compared methods. Moreover, Shapelet-Net performs the best in 20 datasets, which is more than the other three benchmarked methods. The total best accuracy number of Shapelet-Net is almost two times larger than that of NS and clearly more than those of other methods. Shapelet-Net is clearly more accurate in some datasets, such as AtrialFibrillation and StandWalkJump. A probable reason is that high-quality shapelets do exist in those datasets and Shapelet-Net can discover them for classification.

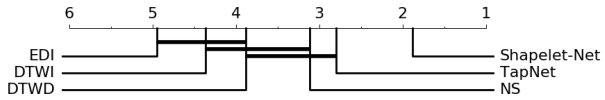


Fig. 10. Critical difference digram showing pairwise statistical difference comparison of 6 methods on UEA archive

TABLE II
ACCURACY OF OUR METHOD AND RELATED METHODS ON UEA ARCHIVE

Dataset	EDI	DTWI	DTWD	NS	TapNet	Shapelet-Net
ArticularyWordRecognition	0.97	0.98	**0.987**	**0.987**	**0.987**	**0.987**
AtrialFibrillation	0.267	0.267	0.22	0.133	0.333	**0.4**
BasicMotions	0.676	**1**	0.975	**1**	**1**	**1**
CharacterTrajectories	0.964	0.969	0.989	0.994	**0.997**	0.98
Cricket	0.944	0.986	**1**	0.986	0.958	0.986
DuckDuckGeese	0.275	0.55	0.6	0.675	0.575	**0.725**
EigenWorms	0.549		0.618	**0.878**	0.489	**0.878**
Epilepsy	0.666	0.978	0.964	0.957	0.971	**0.987**
ERing	**0.133**	**0.133**	**0.133**	**0.133**	**0.133**	**0.133**
EthanolConcentration	0.293	0.304	**0.323**	0.236	**0.323**	**0.323**
FaceDetection	0.519		0.529	0.528	0.556	**0.602**
FingerMovements	0.55	0.52	0.53	0.54	0.53	**0.58**
HandMovementDirection	0.278	0.306	0.231	0.27	**0.378**	0.338
Handwriting	0.2	0.316	0.286	**0.533**	0.357	0.451
Heartbeat	0.619	0.658	0.717	0.737	0.751	**0.756**
InsectWingbeat	0.128			0.16	0.208	**0.25**
JapaneseVowels	0.924	0.959	0.949	**0.989**	0.965	0.984
Libras	0.833	**0.894**	0.87	0.867	0.85	0.856
LSST	0.456	0.575	0.551	0.558	0.568	**0.59**
MotorImagery	0.51		0.5	0.54	0.59	**0.61**
NATOPS	0.85	0.85	0.883	**0.944**	0.939	0.883
PEMS-SF	0.705	0.734	0.711	0.688	**0.751**	**0.751**
PenDigits	0.973	0.939	0.977	**0.983**	0.98	0.977
Phoneme	0.104	0.151	0.151	0.246	0.175	**0.298**
RacketSports	0.868	0.842	0.803	0.862	0.868	**0.882**
SelfRegulationSCP1	0.771	0.765	0.775	**0.846**	0.652	0.782
SelfRegulationSCP2	0.483	0.533	0.539	0.556	0.55	**0.578**
SpokenArabicDigits	0.967	0.959	0.963	0.956	**0.983**	0.975
StandWalkJump	0.2	0.333	0.2	0.4	0.4	**0.533**
UWaveGestureLibrary	0.881	0.868	0.903	0.884	0.894	**0.906**
Total best acc	1	3	3	9	8	20
Ours 1-to-1-Wins	29	26	22	18	20	-
Ours 1-to-1-Draws	1	3	5	5	5	-
Ours 1-to-1-Losses	0	1	3	7	5	-

Our accuracies on 1-to-1-Losses datasets are only slightly lower than those of NS (*e.g.*, JapaneseVowels, Libras) and TapNet (*e.g.*, PenDigits, SpokenArabicDigits).

2) Friedman test and Wilcoxon test: We follow the process described in [9] to do the Friedman test and Wilcoxon-signed rank test with Holm's α (5%) [16] for all the methods.

The Friedman test is a non-parametric statistical test to detect the differences in 30 datasets across 6 methods. Our statistical significance is $p = 0.00$, which is smaller than $\alpha = 0.05$. Thus, we reject the null hypothesis, and there is a significant difference among these 6 methods.

We then conduct the post-hoc analysis among all six methods. The results are visualized by a critical difference diagram in Figure 10 and we note that Shapelet-Net clearly outperforms other approaches. A thick horizontal line groups a set of classifiers that are not significantly different.

3) Utility-based vs random selection: To investigate the effectiveness of the utility function for selecting final shapelets in Section III-D, we conduct another experiment to compare it with random selection. The clustering number is 200 and the value of k in top-k is 50. The random selection number is also 50.

Due to space restrictions, we report the final classification

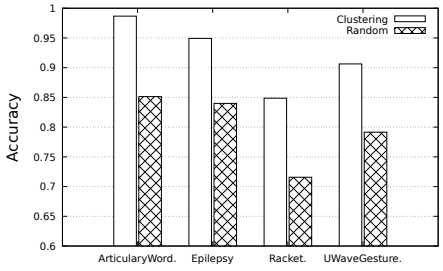


Fig. 11. Utility-based vs random selection of final shapelets

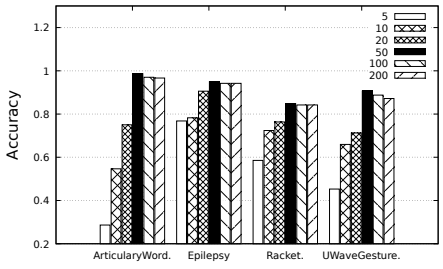


Fig. 12. MTSC accuracy by varying 6 shapelet numbers

accuracies on four MTS datasets, ArticularyWordRecognition, Epilepsy, RacketSports and UWaveGestureLibrary as examples. They are shown in Figure 11. The same trend can be found in other datasets. Among all four datasets, the accuracies of our utility-based method are clearly higher than those of random selection, which presents the superiority of the utility to discover the high-quality shapelets.

4) Varying shapelet numbers: We compare the impact of different top-k shapelet number from 200 clusters on the final accuracy of Shapelet-Net on four MTS datasets, ArticularyWordRecognition, Epilepsy, RacketSports, and UWaveGestureLibrary.

Figure 12 shows the accuracies by varying the shapelet numbers. The accuracies increase rapidly with the increase of the shapelet number from 5 to 50 on all four datasets, and then decrease slightly. This tendency is more evident in the ArticularyWordRecognition dataset than other datasets since ArticularyWordRecognition has 25 classes. Thus, it is much harder to do the classification when the shapelet number is small (*e.g.*, 5). Based on this observation, the default shapelet number of all the datasets is set to 50 in Section IV-E1.

F. Experiments on interpretability

We further investigate the interpretability of shapelets. We report two shapelets (*i.e.*, k = 2) generated by Shapelet-Net from two datasets. These datasets are chosen simply because they do not require much domain knowledge.

1) Interpretability on Basicmotions: Two interesting shapelets S_1 and S_2 are discovered from the Basicmotions dataset (leftmost plots) in Figure 13. S_1 describes the acceleration of x-axis and S_2 depicts the angular velocity of z-axis. The middle plots show four multivariate time series from four classes of the dataset. Different colors show different variables. The distance can only be calculated between the time series of the same variable (visually of the same color). The distances to two shapelets project the multivariate time series into a 2-dimensional space (rightmost plot). Then, the transformed representations are classified by a linear classifier. The result shows that S_2 is effective in distinguishing the motion badminton from others. S_1 can distinguish walking and running from others. Finally, *both* S_1 and S_2 can identify standing from others.

We note that the MST representation is easier to interpret than the raw data and some knowledge can be observed. For example, standing and badminton are similar w.r.t S_1, which is counter-intuitive. When waiting for the badminton, many players just stand.

2) Interpretability on Atrialfibrillation: We use AtrialFibrillation, which is an ECG dataset with two variables, as the second example, to show the interpretability of the discovered multivariate shapelets. There are three classes in the dataset, namely "non-termination atrial fibrillation", "self-terminating at least one minute", and "terminating immediately". They are labeled as N, S, T respectively.

From the brief description of AtrialFibrillation, we can know that the urgency of three classes is T $>$ S $>$ N. However, the raw data even in a plot form is hard to understand. In Figure 14, our shapelets, S_1 and S_2, transform all the original time series into 2-dimensional space. In the MST representation, readers can easily follow the urgency of each class. The smaller the magnitude is in the new space, the more urgent the original time series is.

V. Related Work

In this section, we give some brief introductions about the existing methods of MTSC. We classify them into two main types, namely model-based, and neural network-based.

Model-based methods. A tree classifier based on a new symbolic representation to extract information contained in the relationships for MTS is proposed by Baydogan et al. [4]. An accurate and efficient classification method based on common principal components analysis (PCA) to reduce the dimensionality for MTS is proposed in [21]. WEASEL-MUSE [26] is introduced to utilize the bag of SFA (Symbolic Fourier Approximation) to classify MTS.

Neural network-based methods. Another type is based on neural network. A nice review paper [11] summarizes many neural networks-based methods for time series classification. LSTM-FCN [18] employs a LSTM layer and stacked CNN layer to extract features for a softmax layer to predict the label for classification. [12] applies one positive samples and several negative samples when training their neural network, then SVM is utilized to do the final classification. TapNet [31] is the latest work of this type. It utilizes an attentional prototype network to learn the latent features from MTS. All the methods mentioned above learn an end-to-end classification model, providing little interpretability.

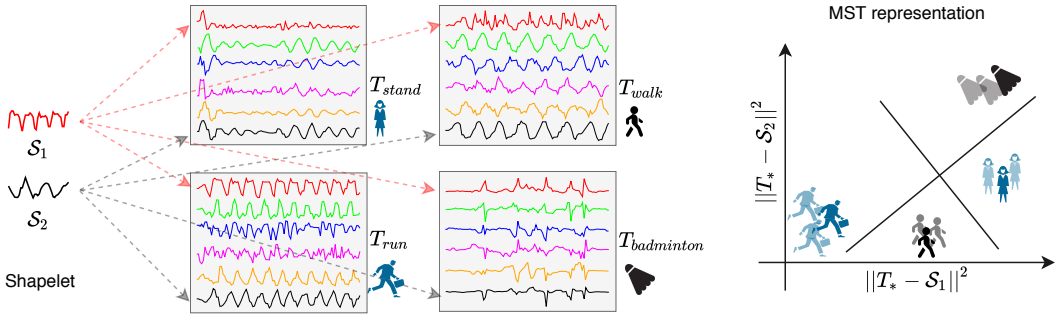


Fig. 13. An example of multivariate shapelet transformation on Basicmotions

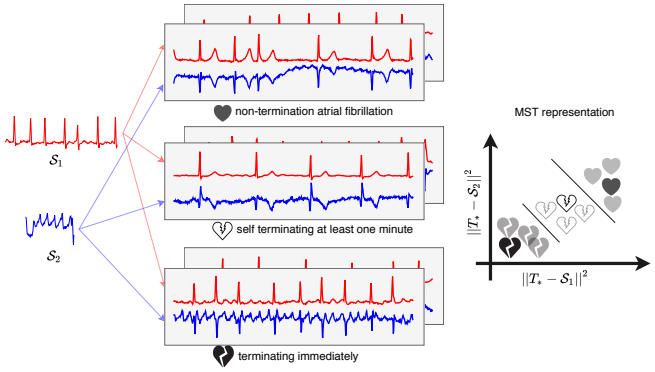


Fig. 14. An example of multivariate shapelet transformation on Atrialfibrillation

VI. Conclusion

This paper has proposed a novel shapelet-neural network approach for MTSC, Shapelet-Net. We propose Mdc-CNN to learn time series subsequences of various lengths into same space and propose a cluster-wise triplet loss to train the network in an unsupervised fashion. We adopt multivariate shapelet transformation to obtain the MST representation of time series. After the transformation, we employ SVM with a linear kernel to do the classification. The experiment results show that the classification accuracy of Shapelet-Net is superior to those of the five compared methods. The learning algorithm converges fast, and the utility function is effective. The number of shapelets can be set to 50 for the highest accuracy. The interpretability of shapelets is illustrated with two case studies. As for future work, we plan to study the MTS with missing values, which is challenging for real-world datasets.

References

[1] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bostrom, P. Southam, and E. Keogh. The uea multivariate time series classification archive, 2018. *arXiv preprint arXiv:1811.00075*, 2018.

[2] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

[3] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[4] M. G. Baydogan and G. Runger. Learning a symbolic representation for multivariate time series classification. *Data Mining and Knowledge Discovery*, 29(2):400–422, 2015.

[5] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[6] A. Bostrom and A. Bagnall. A shapelet transform for multivariate time series classification. *arXiv preprint arXiv:1712.06428*, 2017.

[7] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[8] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(Mar):1109–1135, 2010.

[9] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

[10] Z. Fang, P. Wang, and W. Wang. Efficient learning interpretable shapelets for accurate time series classification. In *ICDE*, pages 497–508, 2018.

[11] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, pages 1–47, 2019.

[12] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi. Unsupervised scalable representation learning for multivariate time series. In *Advances in Neural Information Processing Systems 32*, pages 4652–4663, 2019.

[13] Y. Goldberg and O. Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[14] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme. Learning time-series shapelets. In *SIGKDD*, pages 392–401, 2014.

[15] J. Grabocka, M. Wistuba, and L. Schmidt-Thieme. Fast classification of univariate and multivariate time series through shapelet discovery. *Knowledge and Information Systems*, 49(2):429–454, 2016.

[16] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

[17] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[18] F. Karim, S. Majumdar, H. Darabi, and S. Harford. Multivariate lstm-fcns for time series classification. *Neural Networks*, 116:237–245, 2019.

[19] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[20] G. Li, B. Choi, J. Xu, S. S. Bhowmick, K. P. Chun, and G. Wong. Efficient shapelet discovery for time series classification. *Technical Report*, 2019.

[21] H. Li. Accurate and efficient classification based on common principal components analysis for multivariate time series. *Neurocomputing*, 171:744–753, 2016.

[22] J. Lines, L. M. Davis, J. Hills, and A. Bagnall. A shapelet transform for time series classification. In *SIGKDD*, pages 289–297, 2012.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[24] A. Mueen, E. Keogh, and N. Young. Logical-shapelets: an expressive primitive for time series classification. In *SIGKDD*, pages 1154–1162, 2011.

[25] T. Rakthanmanon and E. Keogh. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *SIAM*, pages 668–676, 2013.

[26] P. Schäfer and U. Leser. Multivariate time series classification with weasel+ muse. *arXiv preprint arXiv:1711.11343*, 2017.

[27] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[28] M. Shokoohi-Yekta, J. Wang, and E. Keogh. On the non-trivial generalization of dynamic time warping to the multi-dimensional case. In *Proceedings of the 2015 SIAM international conference on data mining*, pages 289–297. SIAM, 2015.

[29] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *SSW*, 125, 2016.

[30] L. Ye and E. Keogh. Time series shapelets: a new primitive for data mining. In *SIGKDD*, pages 947–956, 2009.

[31] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu. Tapnet: Multivariate time series classification with attentional prototypical network. In *Thirty-Fourth AAAI conference on artificial intelligence*, 2020.