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

Guozhong Li[1], Byron Choi[1], Jianliang Xu[1], Sourav S Bhowmick[2], Kwok-Pan Chun[3], Grace L.H Wong[4]

[1]Department of Computer Science, Hong Kong Baptist University, Hong Kong
[2]School of Computing Engineering, Nanyang Technological University, Singapore
[3]Department of Geography, Hong Kong Baptist University, Hong Kong
[4]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 short discriminative subsequences that recently have been found not only to be *accurate* but also *interpretable* for the classification problem of univariate time series (UTS). However, existing work on shapelets selection cannot be applied to multivariate time series classification (MTSC) since the candidate shapelets of MTSC may come from different variables of different lengths and thus cannot be directly compared. To address this challenge, in this paper, we propose a novel model called ShapeNet, which embeds shapelet candidates of different lengths into a unified space for shapelet selection. The network is trained using cluster-wise triplet loss, which considers the distance between anchor and *multiple positive (negative) samples* and the distance between positive (negative) samples, which are important for convergence. We compute representative and diversified final shapelets rather than directly using all the embeddings for model building to avoid a large fraction of non-discriminative shapelet candidates. We have conducted experiments on ShapeNet with competitive state-of-the-art and benchmark methods using UEA MTS datasets. The results show that the accuracy of ShapeNet is the best of all the methods compared. Furthermore, we illustrate the shapelets' interpretability with two case studies.

1 Introduction

Multivariate time series (MTS), containing multiple observations at each timestamp, are ubiquitous in many applications, ranging from astronomy, biology, geoscience, and smart cities, to health care, human action recognition, marketing, and other scientific and social domains. For example, data from electroencephalography (EEG) and magnetoencephalography (MEG) are standard multivariate data that 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 the specific case of univariate time series classification (UTSC). Various methods (Bagnall et al. 2017) have been proposed for UTSC, and its accuracy has increased significantly when compared to some benchmark methods, such as 1 Nearest Neighbor (1-NN) with Euclidean distance (ED) or Dynamic Time Warping (DTW) (Berndt and Clifford 1994).

Some related studies on improving MTSC accuracy are presented in Section 2. In particular, *shapelets* (Ye and Keogh 2009) are short discriminative time series subsequences. The effectiveness of shapelet-based classifiers of UTSC has been proven by many related studies in the last decade, *e.g.*, logical shapelets (Mueen, Keogh, and Young 2011), fast shapelets (Rakthanmanon and Keogh 2013), learning shapelets (Grabocka et al. 2014) and dynamic shapelets (Ma et al. 2020). Their efficiency has improved significantly recently (Li et al. 2020a; Hou, Kwok, and Zurada 2016). Importantly, shapelets themselves are intuitive, and the distances between shapelets and time series from different classes indicate significant differences in the classes. To integrate shapelets with standard classifiers, such as SVM and the Naive Bayes classifier, shapelet transformation (Lines et al. 2012) has been proposed.

Challenges. A shapelet-based approach for MTSC is in its infancy, however. Few shapelet-based methods for MTSC have been introduced (Bostrom and Bagnall 2017)(Grabocka, Wistuba, and Schmidt-Thieme 2016). The challenges of a shapelet approach for MTSC can be listed as follows.

- First, multivariate time series, of course, have multiple variables. Shapelet candidates can be *voluminous* and *heterogeneous*. Exhaustive searches of shapelets (Bostrom and Bagnall 2017)(Grabocka, Wistuba, and Schmidt-Thieme 2016) can be inaccurate.
- Second, shapelet candidates of different variables can be of different lengths, and such shapelets are *hard to compare*. With excessive candidates, it is not clear how to select the discriminative ones for classification.
- Third, most existing studies take a black-box approach. Few methods provide interpretable results for understanding and explaining the classification. It is crucial that the MTSC approach maintains the interpretability of shapelets.

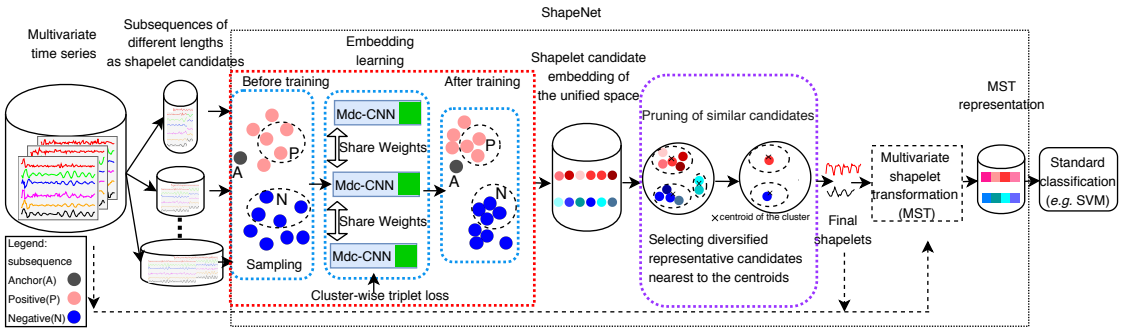


Figure 1: Overview of ShapeNet for multivariate time series classification (MTSC)

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

First, we propose the *Multi-length-input dilated causal Convolutional Neural Network (Mdc-CNN)*, which enhances Dc-CNN (Bai, Kolter, and Koltun 2018), to embed shapelet candidates of different lengths and different variables into a unified space (shapelet embedding). We adopt *dilated convolution*, which 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 that no future value impacts the current value. In addition, we propose a *cluster-wise triplet loss function* for training Mdc-CNN that considers intra/inter cluster metric learning for accelerating convergence and improving stability. Our cluster-wise triplet loss not only takes multiple positive samples and multiple negative samples as input, but also calculates the distance between them. In comparison, the previous triplet loss (Schroff, Kalenichenko, and Philbin 2015) only involves one positive sample and one negative sample. Our loss function is more robust for faster determination of shapelet embedding and convergence (see Figure 4). To the best of our knowledge, this paper is the first to use a neural network to discover shapelets in MTS.

Second, we avoid directly feeding numerous shapelet candidates (encoded by the embedding learning using Mdc-CNN) to build a classifier. We first cluster the shapelet candidate embeddings. We then propose a *utility function* to select top-k candidates that are close to the centroid of a large cluster and different from other cluster centroids, which gives us *representative and diversified* final shapelets.

We then adopt *multivariate shapelet transformation* (MST), which is first formally defined. Specifically, given a multivariate time series, we compute its distance(s) to the selected shapelet(s) of the same variable to obtain a MST representation.

In all, ShapeNet learns the variable-length time series subsequences of different variables into the unified embeddings, where ShapeNet captures the interactions among different variables in MTS.

Finally, because of the MST representation, we can readily learn a classification model. In this paper, we adopt linear SVM, which allows us to visualize how the shapelets of different variables separate the time series of different classes in the case studies.

We conduct experiments on UEA MTS Archive (Bagnall et al. 2018). The results show that ShapeNet is the best of the baselines and the state-of-the-art methods in terms of accuracy. We note that ShapeNet gives the best performance in 14 datasets out of 30 datasets. We present two cases of human action recognition and ECG data, to illustrate how do the shapelets give insights into classification.

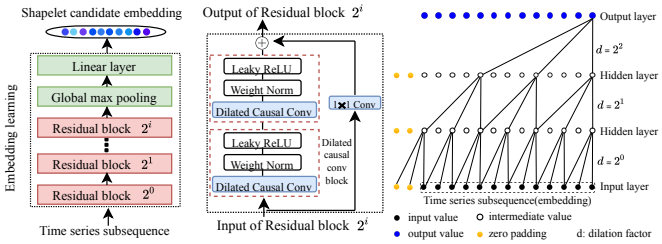
Organization. The rest of this paper is organized as follows. Section 2 reviews the related work. The details of our proposed method are given in Section 3. Section 4 reports the experimental results. Section 5 concludes the paper and presents avenues for future work.

2 Related Work

In this section, we give a brief introduction to 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 was proposed by (Baydogan and Runger 2015). An accurate and efficient classification method based on common principal components analysis (PCA) to reduce the dimensionality for MTS is proposed in (Li 2016). WEASEL-MUSE (Schäfer and Leser 2017) utilizes the bag of SFA (Symbolic Fourier Approximation) to classify MTS.

Neural network-based methods. Another type is based on neural networks. A nice review paper (Fawaz et al. 2019) summarizes many neural networks-based methods for time series classification. LSTM-FCN (Karim et al. 2019) employs an LSTM layer and stacked CNN layer to extract features for a softmax layer to predict the label for classification. (Franceschi, Dieuleveut, and Jaggi 2019) applies one positive sample and several negative samples when training their neural network, then SVM is utilized to do the final classification. TapNet (Zhang et al. 2020) is the latest model of this type. It utilizes an attentional prototype network to learn the latent features from MTS. All the methods men-



(a) Encoder architecture (b) Residual block of Fig. 2(a) (c) Dilated causal conv layer of Fig. 2(b)

Figure 2: An elaboration of the Multi-length-input dilated causal Convolutional Neural Network (Mdc-CNN)

tioned above learn an end-to-end classification model, providing little interpretability.

3 ShapeNet

In this section, we propose a shapelet-neural network approach, namely ShapeNet. Specifically, we present multi-length-input dialted casual CNN, the cluster-wise triplet loss function, and multivariate shapelet transformation.

3.1 Multi-length-input Dilated Causal CNN (Mdc-CNN)

Shapelet candidates are initially all time series subsequences of different lengths. We use sliding windows (the data shown in the cylinders of Figure 1) of discrete sizes to generate the candidates. Our target is to embed all the shapelet candidates from the original space into a new unified space.

Design rationale. ShapeNet adopts a few existing studies as its building blocks. First, the dilated causal convolutional neural network (Dc-CNN) (Van Den Oord and Dieleman 2016) 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, Kolter, and Koltun 2018). The dilated convolution is utilized to modify the receptive field of the convolution. The causal convolution is designed such that the future data do not impact the learning of the past data.

Second, although the output can be of the same length as the input, Dc-CNN 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 unified 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 illustrated in Figure 2. Figure 2(a) shows that the encoder has $i + 1$ layers of residual blocks, where 2^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 the output *shapelet candidate embedding*. Figure 2(b) presents the residual block with two identical subblocks, and a dilated causal convolution block. Figure 2(c) presents a dilated causal convolution example with dilation factor $d = 2^0, 2^1, 2^2$. Further details of Figure 2(b) and Figure 2(c) can be found in (Bai, Kolter, and Koltun 2018).

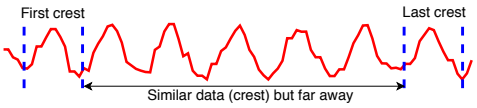


Figure 3: An example from Basicmotions of the violation of the second requirement of word2vec: subsequences that are far away but have a small distance between them

Following the standard practice (*e.g.*, (LeCun, Bengio et al. 1995)(Schroff, Kalenichenko, and Philbin 2015)), Mdc-CNNs use shared weights for training models of shapelet candidates of different lengths and variables, as noted on the left side of Figure 1. The settings of Mdc-CNN are presented in Section 4.2.

3.2 Unsupervised Representation Learning

We next explain how the Mdc-CNN networks are trained in an unsupervised manner. There have been several loss functions for unsupervised learning, such as word2vec (Mikolov et al. 2013), image similarity (Chechik et al. 2010), and face recognition (Schroff, Kalenichenko, and Philbin 2015). In (Chechik et al. 2010) and (Schroff, Kalenichenko, and Philbin 2015), only one positive sample and one negative sample are considered, whereas, in (Franceschi, Dieuleveut, and Jaggi 2019) and (Mikolov et al. 2013), one positive and several negative samples are considered. We recall that Franceschi et al. (Franceschi, Dieuleveut, and Jaggi 2019) followed the principle from word2vec (Mikolov et al. 2013), which makes the assumption that the representation of a word should meet two requirements: (i) the representation should be close to those near its context (Goldberg and Levy 2014), and (ii) it should be distant from those in a randomly chosen context, since they are probably different from the original word's context.

The objectives of learning/training (similar to word2vec) are 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 3. We can easily observe that some crests of the waveform are far away but not distant from each other. ② Only one positive sample is included in a batch to train the network, which is often unstable in the context of the representation learning of shapelets. ③ The distances between negative (positive) samples were not considered before. Figure 4 shows the loss in using the original triplet loss (Franceschi, Dieuleveut, and Jaggi 2019) to learn shapelet representation. It can be noted that while the loss has slightly declined, it is unstable and hardly converges.

Cluster-wise triplet loss function. In this paper, we propose a cluster-wise triplet loss function that takes *multiple positive and negative samples* and *the distance among positives (negatives)* as input. For simplicity, we take two clusters to

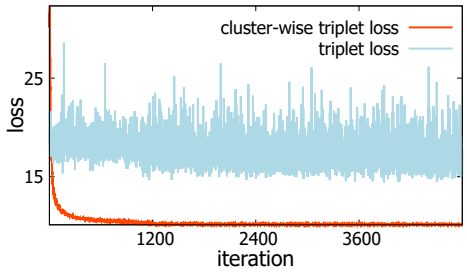


Figure 4: A comparison between our cluster-wise triplet loss (multiple positives and multiple negatives, both with intra distances) and original triplet loss (one positive and multiple negatives without intra distance) on ArticularyWordRecognition (Bagnall et al. 2018)

demonstrate our loss function. 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 of size K^+ and K^-, respectively.

The number of triplet $(x, \boldsymbol{x}^+, \boldsymbol{x}^-)$ in some real-world datasets is large, and it is computationally prohibitive and sub-optimal to use all the triplets for training. Instead, we conduct triplet sampling. The details of our triplet sampling are presented in the supplementary material (Li et al. 2020b).

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

$$\mathcal{D}_{AP} + \mu < \mathcal{D}_{AN}, \tag{1}$$

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.

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

and

$$\mathcal{D}_{AN} = \frac{1}{K^-} \sum_{i=1}^{K^-} ||f(x) - f(x_i^-)||_2^2, \tag{3}$$

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 (negative) samples, the distances among the positive (negative) samples are included and should be small (large). The maximum distance among all positive (negative) samples is presented in Eq. 4 (Eq. 5).

$$\mathcal{D}_{pos} = \max_{i,j \in (1,K^+) \wedge i<j} \{||f(x_i^+) - f(x_j^+)||_2^2\} \tag{4}$$

and

$$\mathcal{D}_{neg} = \max_{i,j \in (1,K^-) \wedge i<j} \{||f(x_i^-) - f(x_j^-)||_2^2\} \tag{5}$$

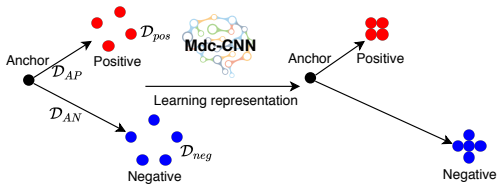


Figure 5: Illustration of the effect of training a model using the cluster-wise triplet loss function, positives are closer to each other and the anchor, negatives are closer to each other but farther from the anchor

The intra-sample loss is defined as follows:

$$\mathcal{D}_{intra} = \mathcal{D}_{pos} + \mathcal{D}_{neg} \tag{6}$$

Putting these together, we propose the *cluster-wise triplet loss function for the triplets* for our model in Eq. 7, to train the network under an unsupervised fashion.

$$\mathcal{L}\left(f(x), f(\boldsymbol{x}^+), f(\boldsymbol{x}^-)\right) = \log \frac{\mathcal{D}_{AP} + \mu}{\mathcal{D}_{AN}} + \lambda \mathcal{D}_{intra} \tag{7}$$

where λ is a hyperparameter.

Example 1 *Eq. 7 is illustrated in Figure 5. Two clusters of our cluster-wise triplet loss are illustrated in this example. 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.* □

3.3 Multivariate Shapelet Transformation

After determining the unified representation of shapelet candidates, we propose to select high-quality and diversified candidates as final shapelets. Finally, we adopt the procedure of shapelet transformation for MTS, then apply a classic classifier to solve the MTSC problem.

Determining final shapelets. By following previous subsections, all the candidates are embedded into a unified space. It allows us to simply employ a clustering method (*e.g.*, kmeans) to obtain Y clusters of the shapelet candidates. We propose a utility (Eq. 8) to rank the candidates that are nearest to the cluster centroids. The first component of Eq. 8 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 shows that the candidate is different from others:

$$\mathcal{U}(f(x_i)) = \beta \cdot \frac{\log(\text{size}(f(x_i)))}{\log(\max_{i=1}^{Y}(\text{size}(f(x_i))))} + (1-\beta) \frac{\log \sum_{j=1}^{Y} ||f(x_i) - f(x_j)||_2^2}{\log(\max_{i=1}^{Y}(\sum_{j=1}^{Y} ||f(x_i) - f(x_j)||_2^2))} \tag{8}$$

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

We select the top-k candidates among all Y clusters according to Eq. 8 and retrieve the original time series subsequences as the *final shapelets*, denoted as $\mathcal{S}_k$.

Multivariate Shapelet Transformation. MST is first mentioned in (Bostrom and Bagnall 2017) and the following is our formal definition of it.

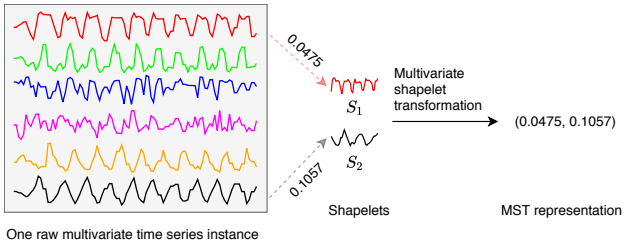


Figure 6: Illustration of transforming an MTS instance into the MST representation

Definition 1 ***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 $S_j \in \mathcal{S}_k$, $T_m^v \in \mathbb{T}_m$, and the variable of S_j and T_m^v is the same.* □

Example 2 *An example of MST is shown in Figure 6. The leftmost plot exhibits an instance with six variables from the Basicmotions dataset. Two shapelets, S_1 and S_2, are in the middle. For MST, we calculate the distance between the time series subsequence with the same variable (e.g., the distance between the first variable (red time series on top) and S_1). Thus, the MST representation of a time series instance is a vector, as shown in the rightmost part.* □

After MST, 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 $V \times N$.

When the transformation of all the MTS instances is completed, some standard classifiers (*e.g.*, SVM) can be exploited to learn a classification model from the transformed representation. In this paper, we adopt SVM with a linear kernel so that we can observe the weights of the shapelets for classification.

4 Experiments

4.1 Environment

We have implemented the proposed method[1] 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).

4.2 Datasets and parameters

A well-known benchmark of MTS datasets, the UEA ARCHIVE, was tested. Detailed information regarding the datasets can be obtained from (Bagnall et al. 2018).

The following are some parameters used in our experiment. We follow the default hyperparameters of the network from (Bai, Kolter, and Koltun 2018). The batch size, the number of channels, the kernel size of the convolutional network, and the network depth are set to 10, 40, 3, and 10, respectively. The learning rate is kept fixed at the low value of $\eta = 0.001$, while the number of epochs for network training is 400. μ in Eq. 1 is set to 0.2, $\lambda = 1$ for the triplet loss function. The β in Eq. 8 is 0.5.

[1]To promote reproducibility, our source code is made public at http://alturl.com/d26bo.

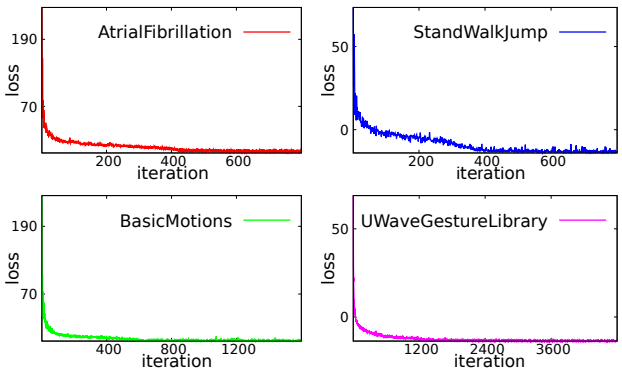


Figure 7: Convergences of the learning algorithm on some MTS datasets

4.3 Convergence of Mdc-CNN

We verify the convergence of Mdc-CNN, which depends on the parameters from Section 4.2. For instance, the convergences of the learning algorithm on four datasets, AtrialFibrillation, Basicmotions, StandWalkJump, and UWaveGestureLibrary, are illustrated in Figure 7.

All the losses converge very smoothly as the training proceeds on all four datasets. We can also observe that the loss converges quickly at the beginning, and then stabilizes. Similar trends can be observed from the rest of the datasets. This verifies the effectiveness of our cluster-wise triplet loss.

4.4 Baselines

We compared ShapeNet with seven different methods. Due to space restrictions, we provide only brief details of each. Interested readers may refer to the original paper for further information.

- **Three benchmarks (Bagnall et al. 2018).** 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 ($DTWD$) (Shokoohi-Yekta, Wang, and Keogh 2015).
- **MLSTM-FCNs (Karim et al. 2019).** MLSTM-FCNs is a deep learning framework transforming the LSTM-FCN models of UTS into MTS by augmenting it with squeeze-and-excitation block.
- **WEASEL-MUSE (Schäfer and Leser 2017).** WEASEL-MUSE is a bag-of-pattern based approach with statistical feature selection, variable window lengths and SAX for MTSC.
- **Negative samples (NS) (Franceschi, Dieuleveut, and Jaggi 2019).** This method applies several negative samples when training their neural network, then SVM is utilized to do the final classification.

- **TapNet (Zhang et al. 2020).** TapNet is a novel MTSC model with an attentional prototype network to harness the strengths of both traditional and deep learning based approaches.

4.5 Experiments on accuracy

Comparison with other methods The experimental accuracies of the baseline results are all taken from the original papers (Bagnall et al. 2018), (Franceschi, Dieuleveut, and Jaggi 2019) and (Zhang et al. 2020), respectively. We only consider the normalized datasets for the experiment. The overall classification accuracy results for the datasets are presented in Table 1. The accuracy results of ShapeNet are the mean values of 10 runs and the standard deviations of all the datasets are less than 0.01.

From Table 1, we can observe that the overall accuracy of ShapeNet is the best of all the methods compared. Moreover, ShapeNet performs best in 14 datasets, more than the other three benchmarked methods. The total best accuracy of ShapeNet is almost two times better than those of NS, TapNet, WEASEL-MUSE, and MLSTM-FCNs, and clearly even better than those of other methods. ShapeNet 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 ShapeNet can discover them for classification. Our accuracies on 1-to-1-Losses datasets are only slightly lower than those of WEASEL-MUSE (*e.g.*, , Cricket, Epilepsy), NS (*e.g.*, JapaneseVowels, Libras) and TapNet (*e.g.*, PenDigits, SpokenArabicDigits).

Friedman test and Wilcoxon test We follow the process described in (Demšar 2006) to conduct the Friedman test and Wilcoxon-signed rank test with Holm's α (5%) (Holm 1979) for all the methods.

The Friedman test is a non-parametric statistical test to detect the differences in 30 datasets across eight 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 eight methods.

We note that ShapeNet ranks the $1st$ on average among all the compared methods. We further conducted the Wilcoxon test against all baselines and found out that all results are statistically significant at $p < 0.05$, except WEASEL-MUSE, NS from the last row in Table 1.

Triplet sampling vs. random sampling To study the performance of our triplet sampling, we compare with random triplet sampling to train the network. Due to limitations of space here, we present only the results from four MTS datasets, namely ArticularyWordRecognition, Epilepsy, RacketSports and UWaveGestureLibrary, in Figure 8. Figure 8 shows the results of final accuracy: our triplet sampling is evidently the best of the four datasets.

Utility-based vs. random selection To study the effectiveness of the utility function for selecting final shapelets in Section 3.3, we conduct an experiment to compare it with random selection. The clustering number is 200 and the value of k is 50. The random selection number is 50.

Table 1: Accuracy of our method and related methods on UEA ARCHIVE

Dataset	EDI	DTWI	DTWD	MLSTM -FCNs	WEASEL +MUSE	NS	TapNet	ShapeNet
ArticularyWordRecognition	0.97	0.98	0.987	0.973	0.99	0.987	0.987	0.987
AtrialFibrillation	0.267	0.267	0.22	0.267	0.333	0.133	0.333	0.4
BasicMotions	0.676	1	0.975	0.95	1	1	1	1
CharacterTrajectories	0.964	0.969	0.989	0.985	0.99	0.994	0.997	0.98
Cricket	0.944	0.986	1	0.917	1	0.986	0.958	0.986
DuckDuckGeese	0.275	0.55	0.6	0.675	0.575	0.675	0.575	0.725
EigenWorms	0.549	N/A	0.618	0.504	0.89	0.878	0.489	0.878
Epilepsy	0.666	0.978	0.964	0.761	1	0.957	0.971	0.987
ERing	0.133	0.133	0.133	0.133	0.133	0.133	0.133	0.133
EthanolConcentration	0.293	0.304	0.323	0.373	0.43	0.236	0.323	0.312
FaceDetection	0.519	N/A	0.529	0.545	0.545	0.528	0.556	0.602
FingerMovements	0.55	0.52	0.53	0.58	0.49	0.54	0.53	0.58
HandMovementDirection	0.278	0.306	0.231	0.365	0.365	0.27	0.378	0.338
Handwriting	0.2	0.316	0.286	0.286	0.605	0.533	0.357	0.451
Heartbeat	0.619	0.658	0.717	0.663	0.727	0.737	0.751	0.756
InsectWingbeat	0.128	N/A	N/A	0.167	N/A	0.16	0.208	0.25
JapaneseVowels	0.924	0.959	0.949	0.976	0.973	0.989	0.965	0.984
Libras	0.833	0.894	0.87	0.856	0.878	0.867	0.85	0.856
LSST	0.456	0.575	0.551	0.373	0.59	0.558	0.568	0.59
MotorImagery	0.51	N/A	0.5	0.51	0.5	0.54	0.59	0.61
NATOPS	0.85	0.85	0.883	0.889	0.87	0.944	0.939	0.883
PEMS-SF	0.705	0.734	0.711	0.699	N/A	0.688	0.751	0.751
PenDigits	0.973	0.939	0.977	0.978	0.948	0.983	0.98	0.977
Phoneme	0.104	0.151	0.151	0.11	0.19	0.246	0.175	0.298
RacketSports	0.868	0.842	0.803	0.803	0.934	0.862	0.868	0.882
SelfRegulationSCP1	0.771	0.765	0.775	0.874	0.71	0.846	0.652	0.782
SelfRegulationSCP2	0.483	0.533	0.539	0.472	0.46	0.556	0.55	0.578
SpokenArabicDigits	0.967	0.959	0.963	0.99	0.982	0.956	0.983	0.975
StandWalkJump	0.2	0.333	0.2	0.067	0.333	0.4	0.4	0.533
UWaveGestureLibrary	0.881	0.868	0.903	0.891	0.916	0.884	0.894	0.906
Total best acc	1	2	2	4	12	5	5	14
Ours 1-to-1-Wins	29	26	22	21	15	18	20	-
Ours 1-to-1-Draws	1	3	5	3	3	5	5	-
Ours 1-to-1-Losses	0	1	3	6	12	7	5	-
Rank Mean	6.2	5.43	4.77	4.6	3.47	3.67	3.23	2.23
Wilcoxon Test p-vlaue	0.000	0.000	0.000	0.001	0.183	0.819	0.002	-

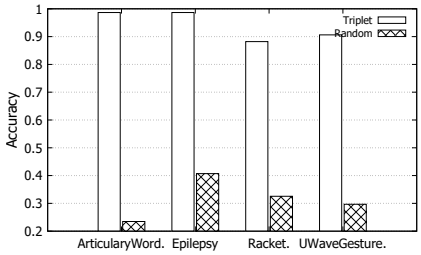


Figure 8: Triplet vs random sampling of final shapelets

Due to space restrictions, we report the final classification accuracy on four MTS datasets, ArticularyWordRecognition, Epilepsy, RacketSports and UWaveGestureLibrary as examples. They are shown in Figure 9. The same trend can be found in other datasets. Among all four datasets, the accuracy of our utility-based method is clearly better than those of random selection, which shows its superiority ability to discover high-quality shapelets.

Varying shapelet numbers We compare the impact of different number of top-k shapelets from 200 clusters on the final accuracy of ShapeNet on four MTS datasets: ArticularyWord., Epilepsy, RacketSports, and UWaveGestureLibrary.

Figure 10 shows accuracy by varying shapelet numbers. The accuracy increases rapidly as the number of shapelets increases from 5 to 50 in all four datasets, and then decreases slightly. This tendency is more evident in the ArticularyWordRecognition dataset than the 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 4.5.

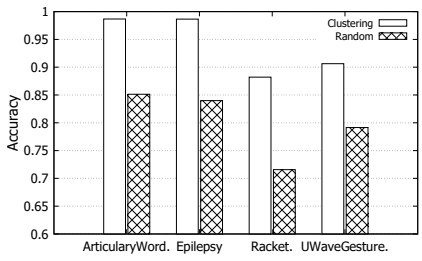


Figure 9: Utility-based vs random selection of final shapelets

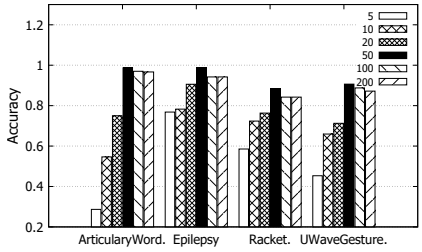


Figure 10: MTSC accuracy by varying 6 shapelet numbers

4.6 Experiments on interpretability

We further investigate the shapelets' interpretability, which is a strength of shapelet-based methods. We report two shapelets (*i.e.*, $k = 2$) generated by ShapeNet from two datasets. These datasets are chosen simply because they can be presented without much domain knowledge.

Interpreting Basicmotions' shapelets Two interesting shapelets, S_1 and S_2, are discovered from the Basicmotions dataset (leftmost plots) in Figure 11. S_1 describes the acceleration of the x-axis and S_2 depicts the angular velocity of the z-axis. The shapelets selected by ShapeNet are from the first and fifth variables, which shows the differing importance of the variables. 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 two-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 badminton motion from others. S_1 can distinguish walking and running from others. Finally,

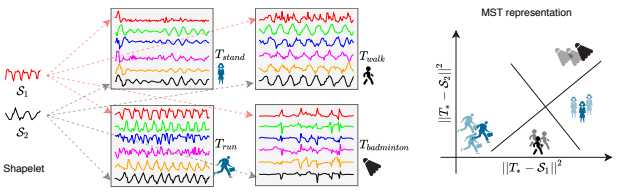


Figure 11: An example of multivariate shapelet transformation on Basicmotions

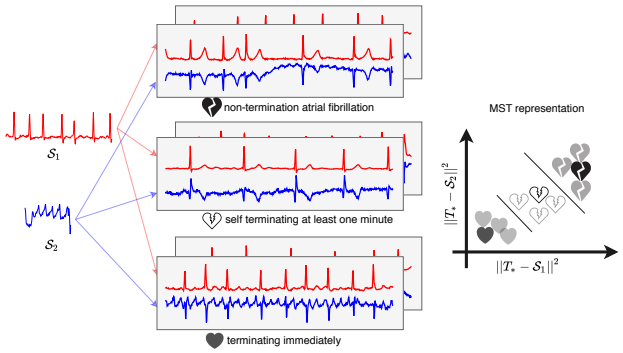


Figure 12: An example of multivariate shapelet transformation on Atrialfibrillation

both S_1 and S_2 can identify standing from others.

We note that the MST representation w.r.t shapelets 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. It turns out that when waiting for the badminton, many players just stand.

Interpreting AtrialFibrillation's shapelets We use AtrialFibrillation, which is an ECG dataset with two variables, as an 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, and T, respectively.

From the brief description of AtrialFibrillation, we can know that the terminating time of the three classes is T $<$ S $<$ N. However, the raw data are hard to understand even in a plot form. In Figure 12, our shapelets, S_1 and S_2, transform all the original time series into two-dimensional space. In the MST representation, readers can easily follow the terminating time of each class. The larger the magnitude in the new space, the more time for terminating on the original time series.

5 Conclusion

This paper has proposed a novel shapelet-neural network approach for MTSC, ShapeNet. We propose Mdc-CNN to learn time series subsequences of various lengths into unified space and propose a cluster-wise triplet loss to train the network in an unsupervised fashion. We adopt MST to obtain the MST representation of time series. After the transformation, we employ SVM with a linear kernel to do the classification. The experiment's results show that the classification accuracy of ShapeNet is superior to seven compared methods. The learning algorithm converges quickly, and the utility function is effective. The number of shapelets can be set to 50 (by default) 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.

Acknowledgments. We thank the anonymous reviewers for their helpful feedbacks. This work has been supported by the Hong Kong Research Grant Council (HKRGC), HKBU12232716, 12201518 and 12201119, C6030-18GF, GDNSF/2019B1515130001, and IRCMS/19-20/H01, and National Science Foundation of China (NSFC) 61602395. This work was partially supported by the Health and Medical Research Fund (HMRF) of the Food and Health Bureau (Ref. no: 07180216) awarded to Grace Wong.

References

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

Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; and Keogh, E. 2017. 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.

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

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

Berndt, D. J.; and Clifford, J. 1994. Using dynamic time warping to find patterns in time series. In *SIGKDD workshop*, volume 10, 359–370.

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

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

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

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

Franceschi, J.-Y.; Dieuleveut, A.; and Jaggi, M. 2019. Unsupervised Scalable Representation Learning for Multivariate Time Series. In *NeurIPS*, 4652–4663.

Goldberg, Y.; and Levy, O. 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722* .

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

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

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

Hou, L.; Kwok, J. T.; and Zurada, J. M. 2016. Efficient learning of timeseries shapelets. In *AAAI*, 1209–1215.

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

LeCun, Y.; Bengio, Y.; et al. 1995. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks* 3361(10): 1995.

Li, G.; Choi, B.; Xu, J.; Bhowmick, S. S.; Chun, K. P.; and Wong, G. 2020a. Efficient Shapelet Discovery for Time Series Classification. *IEEE Transactions on Knowledge and Data Engineering* doi:10.1109/TKDE.2020.2995870.

Li, G.; Choi, B.; Xu, J.; Bhowmick, S. S.; Chun, K. P.; and Wong, G. 2020b. Supplementary Material of ShapeNet. http://alturl.com/wtpe8.

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

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

Ma, Q.; Zhuang, W.; Li, S.; Huang, D.; and Cottrell, G. W. 2020. Adversarial Dynamic Shapelet Networks. In *AAAI*, 5069–5076.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *NeurIPS*, 3111–3119.

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

Rakthanmanon, T.; and Keogh, E. 2013. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *SDM*, 668–676.

Schäfer, P.; and Leser, U. 2017. Multivariate time series classification with WEASEL+ MUSE. *arXiv preprint arXiv:1711.11343* .

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 815–823.

Shokoohi-Yekta, M.; Wang, J.; and Keogh, E. 2015. On the non-trivial generalization of dynamic time warping to the multi-dimensional case. In *SDM*, 289–297. SIAM.

Van Den Oord, A.; and Dieleman, S. e. 2016. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* 125.

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

Zhang, X.; Gao, Y.; Lin, J.; and Lu, C.-T. 2020. TapNet: Multivariate Time Series Classification with Attentional Prototypical Network. In *AAAI*.