Supplementary Material ShapeNet: 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

The organization of this supplementary materials is as follows. Section 1 presents some preliminaries and the problem statement. The details of the preprocess are given in Section 2. Section 3 reports the differentiation of the loss function. Section 4 explains the pseudo-code of multivariate shapelet transformation. Section 5 introduces the datasets tested in the full paper.

1 Preliminaries

In this section, we present some preliminaries and the problem statement. We summarize the notations and their meanings in Table 1.

Definition 1. **Distance between two time series** [3]. 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 $|T_q| \geq |T_p|$),

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

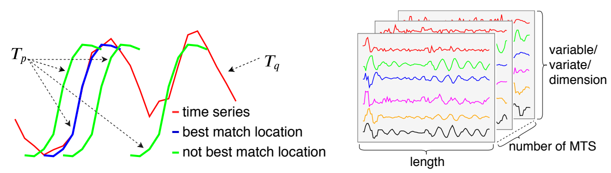
where tq_i and tp_i are the i-th value of T_p and T_q, respectively. □

Intuitively, dist is the distance of the shorter sequence T_p to the most similar subsequence in T_q, as illustrated with in Figure 1(a). Then we define **Shapelet** S as follows.

Definition 2. **Shapelet** S [9]. 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 discriminates C_j from other classes, *i.e.*, $\mathcal{C} \setminus \{C_j\}$. That is, for all T_j having the label C_j, $\mathsf{dist}(T_j, S)$ is smaller than $\mathsf{dist}(T_k, S)$, where T_k is time series having a label in $\mathcal{C} \setminus \{C_j\}$. □

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

Manuscript received XX XX, XXXX;



(a) The best match location

(b) A MTS dataset — extracted from Basicmotions [1]

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

The apparent difference between UTS and MTS is that there are multiple observations at each timestamp in MTS. An example of the MTS dataset, called Basicmotions from [1], is shown in Figure 1(b). While the definition of shapelet can be naturally extended for MTS, the candidates of shapelet of MTS are voluminous, which makes the classification problem more difficult.

Example 1. In Figure 1(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. □

Algorithm 1: Shapelet candidate 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 $\Omega_l = \emptyset$;
  for $m = \{1, 2, \cdots, M\}$ **do**
    for $v = \{1, 2, \cdots, V\}$ **do**
      for $i = \{1, 2, \cdots, N - l + 1\}$ **do**
        $e = T^v(i, i + l - 1)$;
        $\Omega_l = \Omega_l + e$;
return Ω

Algorithm 2: Selection of triplet (APN)

Input: Shapelet candidates Ω
Output: $(x, \boldsymbol{x}^+, \boldsymbol{x}^-)$
1 $\bigcup_{i=1}^{Y} \Omega^i \leftarrow$ kmeans(Ω) ;
2 **for** $i = \{1, 2, \cdots, Y\}$ **do** `// for each cluster`
3   {Anchor selection}
4   $x \leftarrow \Omega^i$.Random() ;
5   $\Omega^i = \Omega^i \setminus \{x\}$;
6   {Positive selection}
7   **for** $k = \{1, 2, \cdots, K^+\}$ **do**
8     $x^+ = \Omega^i$.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** $k = \{1, 2, \cdots, \lceil \frac{K^-}{Y-1} \rceil\}$ **do**
15       $x^- = \Omega^j$.Random() ;
16       $\Omega^i = \Omega^i \setminus \{x^-\}$;
17       $\boldsymbol{x}^- = \boldsymbol{x}^- \cup \{x^-\}$;
18   k++ ;
19 **return** $(x, \boldsymbol{x}^+, \boldsymbol{x}^-)$

2 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 unsupervised representation learning.

Shapelet candidate generation. We apply sliding windows of different sizes to generate abundant shapelet candidates from the original multivariate time series [7][8]. A variate label is then annotated to each candidate, for shapelet transformation in Section 4. Thus, there are two labels for each shapelet candidate: one for the variable and one for the class of the time series. The shapelet candidate 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.

Specifically, we construct the triplet tuple $(x, \boldsymbol{x}^+, \boldsymbol{x}^-)$, namely $\left(x, \bigcup_{i \in [1, K^+]} (x_i^+), \bigcup_{i \in [1, K^-]} (x_i^-)\right)$ from shapelet candidates at the beginning of each iteration as follows. **I.** We do the clustering on the candidates by kmeans [5] 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 $\boldsymbol{x}^+$. **III.** For the negative samples $\boldsymbol{x}^-$, 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 2. Algo. 2 presents the pseudo-code of triplet selection.

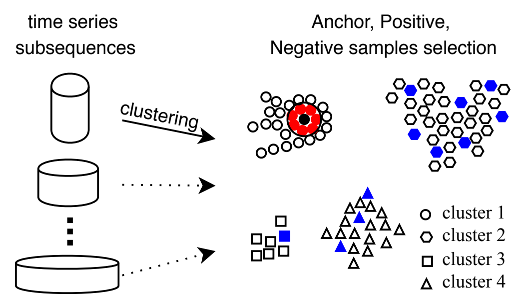


Fig. 2. An illustration of triplet sampling, black for Anchor, red for Positive samples, blue for Negative samples [best viewed in color]

Algo. 2 details the pseudo-code of triplet selection. In Line 1, the shapelet candidates are clustered. This avoids selecting positive (negative, respectively) samples that are highly similar to each other, but 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.

Example 2. After shapelet candidate generation, we have some candidates with different lengths (the leftmost part of Figure 2). 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 2. 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,

TABLE 1
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 $1 \le a \le b \le N$, a and b, the beginning and ending positions
$\mathcal{C}$	the label set
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

7, 1, 3 negative samples (the blue solid hexagon, square, triangle) in these three clusters, respectively. □

After we train the network with the triplets 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 iteration, 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 be improved through sampling in practice.

3 Differentiation of the loss function

In order to compute the derivative of Eq. 7 in the full paper, all the involved functions of the model should be differentiable. Unfortunately, the maximum function of Eq. 4 and Eq. 5 in the full paper are not continuous and differentiable. We therefore introduce a differentiable approximation to the maximum function [2].

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.

$$\mathcal{D}_{pos} \approx \tilde{\mathcal{D}}_{pos} = \frac{\sum\limits_{i=1}^{K^+}\sum\limits_{j=1}^{K^+} \mathcal{D}_{i,j}^{+} \cdot e^{\alpha \cdot \mathcal{D}_{i,j}^{+}}}{\sum\limits_{i=1}^{K^+}\sum\limits_{j=1}^{K^+} e^{\alpha \cdot \mathcal{D}_{i,j}^{+}}} \tag{2}$$

and

$$\mathcal{D}_{neg} \approx \tilde{\mathcal{D}}_{neg} = \frac{\sum\limits_{i=1}^{K^-}\sum\limits_{j=1}^{K^-} \mathcal{D}_{i,j}^{-} \cdot e^{\alpha \cdot \mathcal{D}_{i,j}^{-}}}{\sum\limits_{i=1}^{K^-}\sum\limits_{j=1}^{K^-} e^{\alpha \cdot \mathcal{D}_{i,j}^{-}}}, \tag{3}$$

where $\alpha > 0$ in Eq. 2 and Eq. 3 yields a smooth maximum approximation.

The gradients of overall maximum distance are presented in Eq. 4 and Eq. 5.

$$\frac{\partial \tilde{\mathcal{D}}_{pos}}{\partial \mathcal{D}_{i,j}^{+}} = \frac{e^{\alpha \cdot \mathcal{D}_{i,j}^{+}}(1 + \alpha(\mathcal{D}_{i,j}^{+} - \tilde{\mathcal{D}}_{pos}))}{\sum\limits_{i=1}^{K^+}\sum\limits_{j=1}^{K^+} e^{\alpha \cdot \mathcal{D}_{i,j}^{+}}} \tag{4}$$

$$\frac{\partial \tilde{\mathcal{D}}_{neg}}{\partial \mathcal{D}_{i,j}^{-}} = \frac{e^{\alpha \cdot \mathcal{D}_{i,j}^{-}}(1 + \alpha(\mathcal{D}_{i,j}^{-} - \tilde{\mathcal{D}}_{neg}))}{\sum\limits_{i=1}^{K^-}\sum\limits_{j=1}^{K^-} e^{\alpha \cdot \mathcal{D}_{i,j}^{-}}} \tag{5}$$

Thus, the gradients of Eq. 7 in the full paper with respect to $f(x), f(x_i{}^+), f(x_i{}^-)$ are as follows:

$$\frac{\partial \mathcal{L}}{\partial f(x)} = \frac{2\sum\limits_{i=1}^{K^+} ||f(x) - f(x_i^+)||_2}{\sum\limits_{i=1}^{K^+} ||f(x) - f(x_i^+)||_2^2} - \frac{2\sum\limits_{i=1}^{K^-} ||f(x) - f(x_i^-)||_2}{\sum\limits_{i=1}^{K^-} ||f(x) - f(x_i^-)||_2^2} \tag{6}$$

$$\frac{\partial \mathcal{L}}{\partial f(x_i{}^+)} = \frac{2\sum\limits_{i=1}^{K^+} ||f(x_i^+) - f(x)||_2}{\sum\limits_{i=1}^{K^+} ||f(x) - f(x_i^+)||_2^2} + \sum_{j=1}^{K^+} \frac{\partial \tilde{\mathcal{D}}_{pos}}{\partial \mathcal{D}_{i,j}^{+}} \cdot 4||f(x_i^+) - f(x_j^+)||_2 \tag{7}$$

$$\frac{\partial \mathcal{L}}{\partial f(x_i{}^-)} = \frac{2\sum\limits_{i=1}^{K^-} ||f(x) - f(x_i^-)||_2}{\sum\limits_{i=1}^{K^-} ||f(x) - f(x_i^-)||_2^2} + \sum_{j=1}^{K^-} \frac{\partial \tilde{\mathcal{D}}_{neg}}{\partial \mathcal{D}_{i,j}^{-}} \cdot 4||f(x_i^-) - f(x_j^-)||_2 \tag{8}$$

Since Eq. 7 is differentiable, we use back propagation over the entire neural network based upon mini-batch stochastic gradient descent together with Adam optimizer [6] to optimize our ShapeNet parameters.

Discussions. There are two main advantages of our cluster-wise triplet loss function over the previous ones [4]. Firstly, 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 have not been considered before. Secondly, the important property of shapelets is that a shapelet is a subsequence (the anchor) of a time series such 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.

4 Multivariate Shapelet Transformation

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. 8 in the full paper) 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 time series that has 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 $\tilde{\mathbb{D}} = \tilde{\mathbb{T}}^{M \times k}$.

5 Datasets

A well-known benchmark of MTS datasets, namely UEA Archive, was tested. The detailed information of the datasets can be obtained from [1][1]. Table 2 shows the datasets' information, settings of the experiments with the datasets, where *Train*, *Test*, *Dimension*, *Length*, and *Class* are the numbers of time series in the training set and the testing set, the variable of each instance, the length of time series, and the number of classes, respectively.

1. http://www.timeseriesclassification.com

Algorithm 3: Multivariate Shapelet Transformation

Input: MTS dataset $\mathbb{D} = \mathbb{T}^{M \times V \times N}$, $f(\Omega)$, k
Output: Shapelets $\mathcal{S}_k$
Initialize the priority queue $f(\mathcal{S}) = \emptyset, \mathcal{S} = \emptyset$;
{Shapelet discovery}
$\bigcup_{i=1}^{Y} (f(\Omega))^i \leftarrow \mathsf{kmeans}(f(\Omega))$;
for $i = \{1, 2, \cdots, Y\}$ **do** `// each cluster`
  $min = +\infty$;
  $f(\mathcal{S}).\mathsf{push}(f(min))$;
  foreach $f(e) \in (f(\Omega))^i$ **do**
    $tmp = ||(f(\Omega))^i.\mathsf{centroid} - f(e)||_2^2$;
    if $tmp < min$ **then**
      $min = tmp$;
      $f(\mathcal{S}).\mathsf{pop}()$;
      $f(\mathcal{S}).\mathsf{push}(f(e))$;
Calculate $\mathcal{U}$ (Eq. 8 in the full paper) for each candidate in $f(\mathcal{S})$;
Sort each candidate based on $\mathcal{U}$ and select top-k candidates, denoted as $f(\mathcal{S}_k)$;
Retrieve $\mathcal{S}_k$ of $f(\mathcal{S}_k)$ from $\mathbb{D}$;
{Shapelet transformation}
for $m = \{1, 2, \cdots, M\}$ **do**
  for $j = \{1, 2, \cdots, k\}$ **do**
    $v = S_j.variable$;
    $d_{m,j} = \mathsf{dist}(\mathbb{T}_m^v, S_j)$;
    $\tilde{\mathbb{T}}_m.\mathsf{append}(d_{m,j})$;
  $\tilde{\mathbb{T}}_m =< d_{m,1}, d_{m,2}, \cdots, d_{m,k} >$;
$\tilde{\mathbb{D}} = \tilde{\mathbb{T}}^{M \times k}$;
return $\mathcal{S}_k$

TABLE 2
Multivariate time series datasets information

Dataset	Train	Test	Dimension	Length	Class
ArticularyWordRecognition	275	300	9	144	25
AtrialFibrillation	15	15	2	640	3
BasicMotions	40	40	6	100	4
CharacterTrajectories	1422	1436	3	182	20
Cricket	108	72	6	1197	12
DuckDuckGeese	60	40	1345	270	5
EigenWorms	128	131	6	17984	5
Epilepsy	137	138	3	206	4
ERing	30	30	4	65	6
EthanolConcentration	261	263	3	1751	4
FaceDetection	5890	3524	144	62	2
FingerMovements	316	100	28	50	2
HandMovementDirection	320	147	10	400	4
Handwriting	150	850	3	152	26
Heartbeat	204	205	61	405	2
InsectWingbeat	30000	20000	200	78	10
JapaneseVowels	270	370	12	29	9
Libras	180	180	2	45	15
LSST	2459	2466	6	36	14
MotorImagery	278	100	64	3000	2
NATOPS	180	180	24	51	6
PEMS-SF	267	173	963	144	7
PenDigits	7494	3498	2	8	10
Phoneme	3315	3353	11	217	39
RacketSports	151	152	6	30	4
SelfRegulationSCP1	268	293	6	896	2
SelfRegulationSCP2	200	180	7	1152	2
SpokenArabicDigits	6599	2199	13	93	10
StandWalkJump	12	15	4	2500	3
UWaveGestureLibrary	120	320	3	315	8

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

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

[4] J.-Y. Franceschi, A. Dieuleveut, and M. Jaggi. Unsupervised scalable representation learning for multivariate time series. In *NeurIPS*, pages 4652–4663, 2019.

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

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

[7] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15(2):107–144, 2007.

[8] P. Senin and S. Malinchik. Sax-vsm: Interpretable time series classification using sax and vector space model. In *ICDM*, pages 1175–1180, 2013.

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