Adversarial Dynamic Shapelet Networks

Qianli Ma,[1] **Wanqing Zhuang,**[1] **Sen Li,**[1] **Desen Huang,**[1] **Garrison W. Cottrell**[2]
[1]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China
[2]Department of Computer Science and Engineering, University of California, San Diego, CA, USA
qianlima@scut.edu.cn, scutzwq@gmail.com

Abstract

Shapelets are discriminative subsequences for time series classification. Recently, learning time-series shapelets (LTS) was proposed to learn shapelets by gradient descent directly. Although learning-based shapelet methods achieve better results than previous methods, they still have two shortcomings. First, the learned shapelets are fixed after training and cannot adapt to time series with deformations at the testing phase. Second, the shapelets learned by back-propagation may not be similar to any real subsequences, which is contrary to the original intention of shapelets and reduces model interpretability. In this paper, we propose a novel shapelet learning model called Adversarial Dynamic Shapelet Networks (ADSNs). An adversarial training strategy is employed to prevent the generated shapelets from diverging from the actual subsequences of a time series. During inference, a shapelet generator produces sample-specific shapelets, and a dynamic shapelet transformation uses the generated shapelets to extract discriminative features. Thus, ADSN can dynamically generate shapelets that are similar to the real subsequences rather than having arbitrary shapes. The proposed model has high modeling flexibility while retaining the interpretability of shapelet-based methods. Experiments conducted on extensive time series data sets show that ADSN is state-of-the-art compared to existing shapelet-based methods. The visualization analysis also shows the effectiveness of dynamic shapelet generation and adversarial training.

Introduction

Time series classification is a problem where a set of time series from different categories are given to an algorithm that learns to map them to their corresponding categories. Such problems are ubiquitous in daily life, including categorizing financial records, weather data, electronic health records, and so on. In recent years, a novel approach called shapelets (Ye and Keogh 2009) has attracted a great deal of attention in this domain. Shapelets are discriminative subsequences of time series, and have been successfully applied to time series classification tasks. The category of a time series is distinguished by the presence or absence of one or more shapelets somewhere in the whole series. Since time series from different categories are often distinguished by their local pattern rather than by global structure, shapelets are an effective approach to this problem. Moreover, shapelet-based methods can provide interpretable decisions (Ye and Keogh 2009) since the important local patterns found from original subsequences are used to identify the category of time series.

The original shapelet-based algorithm constructs a decision tree classifier by recursively searching the best shapelet data split (Ye and Keogh 2009). The candidate subsequences are assessed by information gain, and the best shapelet is found at each node of the tree through enumerating all candidates. Lines et al. (Lines et al. 2012) proposed an algorithm called shapelet transformation that finds the top k shapelets in a single pass and then uses the shapelets to transform the original time series, where each attribute of the new representation is the distance between the original series and one of the shapelets. The transformed data can be used in conjunction with any sort of classifier. Thus it can achieve better performance while simultaneously reducing the run time.

Recently, Grabocka et al. (Grabocka et al. 2014) proposed a novel shapelet discovery approach called learning time-series shapelets (LTS), which uses gradient descent to learn shapelets directly rather than searching over a large number of candidates. Since LTS learns the shapelets jointly with the classifier, it further improves the classification accuracy. Motivated by LTS, other learning-based shapelet methods (Shah et al. 2016; Zhang et al. 2016) have been proposed. However, although these learning-based methods achieve better performance than previous methods, they still have two main shortcomings. First, since the shapelets learned with these methods are fixed after training, they cannot deal with novel deformations of local patterns, which will result in the failure to recognize important patterns at test time. Second, the original shapelet-based methods provide interpretability because the shapelets are based on discriminative subsequences. However, there is no constraint for existing learning-based shapelet methods that the shapelets should resemble subsequences, which reduces their interpretability. Learning-based methods can have arbitrary shapes and even diverge from the real subsequences, contrary to the original intent of shapelet methods.

In this paper, we propose a novel model called Adversarial Dynamic Shapelet Networks (ADSNs). Inspired by Jia et al. (Jia et al. 2016) who proposed Dynamic Filter Networks (DFNs) that generate the convolutional filters conditioned on input data, ADSNs use a shapelet generator to generate a set of shapelets conditioned on subsequences of the input time series. Here, although the parameters of the shapelet generator are fixed after training, the shapelet generator can generate different shapelets according to different input samples at test phase. The time series is then transformed by the sample-specific shapelets, and the new representations are passed through a softmax layer to calculate the final probability distribution for each label. We model the shapelet generating process as a two-player minimax game following the Generative Adversarial Network (GAN) (Goodfellow et al. 2014) approach. Specifically, a discriminator is trained to distinguish the generated shapelets from the actual subsequences of the input. Therefore, the generated shapelets will be similar to the subsequences of the input time series. Moreover, we add a shapelet diversity regularization term (Zhang et al. 2018) to the objective function to increase the diversity of the generated shapelets and avoid mode collapse (Salimans et al. 2016). Our contributions can be summarized as follows:

- We propose a *shapelet generator* to dynamically produce shapelets that are sample-specific, improving the modeling flexibility and classification performance.
- In order to prevent the generated shapelets from creating arbitrary shapes, an adversarial training strategy is employed to ensure the generated shapelets are similar to the actual subsequences of the time series. To the best of our knowledge, this is the first work that uses an adversarial training strategy to learn shapelets.
- Our experimental results on a large number of time series datasets show that the proposed model achieves state-of-the-art performance and the effectiveness of our model is demonstrated through visualization analysis.

Related Work

The original shapelet classifier has two major limitations. First, it is slow: the shapelet discovery process is extremely time-consuming. Second, by making the decision tree an integral part of the algorithm, it cannot combine the shapelets with other classifiers. More recent shapelet-based methods address these deficiencies, and can be roughly divided into two categories: 1) accelerating shapelet discovery; 2) using the shapelets to transform the data into a feature space that can be used by other classification algorithms.

Accelerating Shapelet Discovery. Ye and Keogh (Ye and Keogh 2009) stop the distance computation early, and use entropy pruning to avoid large calculations when searching for the best shapelet. Chang et al. (Chang et al. 2012) implemented a parallel version of shapelet discovery on GPUs, significantly reducing the runtime. Rakthanmanon and Keogh (Rakthanmanon and Keogh 2013) used a technique called symbolic aggregate approximation (SAX) (Lin et al. 2007) to transform the raw time series into a discrete and low dimensional representation, accelerating the search process. Grabocka et al. (Grabocka, Wistuba, and Schmidt-Thieme 2015) proposed pruning the shapelets that are similar, and using a supervised selection process to choose shapelets based on how well they improve classification accuracy. Hou et al. (Hou, Kwok, and Zurada 2016) proposed a novel shapelet discovery approach that learns the shapelet positions by using a generalized eigenvector method, and a fused lasso regularizer to get a sparse and "blocky" solution. This approach treats the shapelet discovery task as a numerical optimization problem and is faster than previous shapelet-based methods. Although the above-mentioned methods improved computational efficiency, there is still room for improving accuracy.

Shapelet Transformation. Instead of embedding shapelet discovery within a decision tree classifier, the Shapelet Transformation (ST) algorithm finds the most discriminative subsequences as shapelets in a single pass through the data (Lines et al. 2012). The shapelets are used to transform the time series data into new representations, in which each attribute is the distance of a time series to one of the shapelets.

Grabocka et al. (Grabocka et al. 2014) proposed using gradient descent to learn the top k shapelets directly, rather than searching among candidate subsequences. A shapelet transformation is then applied to the time series. Shah et al. (Shah et al. 2016) also used gradient descent to learn shapelets, but replaced the Euclidean distance measure with Dynamic Time Warping (DTW) distance (Berndt and Clifford 1994). Zhang et al. (Zhang et al. 2016) used learned shapelets for time series clustering task, using the shapelet transformation before clustering.

Whether obtaining shapelets by searching or learning, these shapelet-based methods use fixed shapelets after training, and thus cannot be adapted to time series with deformations, reducing the modeling flexibility. In contrast, we propose a sample-specific shapelet learning model that dynamically generates the shapelets according to the input series, and use an adversarial training strategy to constrain the shapelets to be similar to the actual subsequences.

Proposed Method

The general structure of an ADSN is illustrated in Figure 1. The shapelet generator is used to generate a set of shapelets conditioned on subsequences of input time series. Then the dynamic shapelet transformation is performed on the input time series to extract the discriminative features, and a softmax layer is used to calculate the final probability distribution for each class. A diversity regularization term constrains the generated shapelets to be different from each other, while the adversarial training strategy ensures the generated shapelets are similar to the actual subsequences.

Shapelet Generator.

To generate shapelets of length L conditioned on the input time series, we first use an L-length sliding window with a stride of 1 to extract the subsequences of the time series, and then use one convolutional layer to generate the shapelets.

Given a set of n time series $\mathbf{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n\}$, each time series $\mathbf{t}_i$ contains m ordered real values, denoted as $\mathbf{t}_i =$

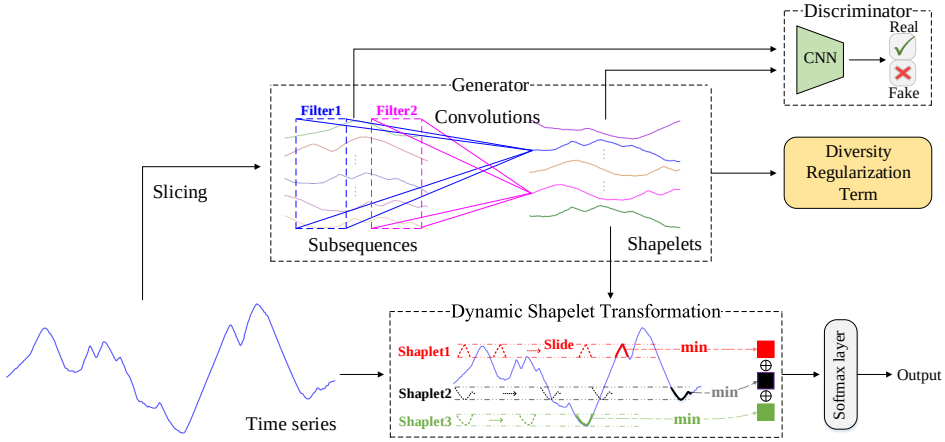


Figure 1: General architecture of the Adversarial Dynamic Shapelet Network (ADSN).

$(t_{i,1}, t_{i,2}, \ldots, t_{i,m})^T$. With a sliding window length L and stride of 1, we can obtain P subsequences with length L, where $P = m - L + 1$ is the number of subsequences.

We concatenate all the subsequences of i-th time series and denote the result as $\mathbf{O}_i$, where $\mathbf{O}_i \in \mathbf{R}^{L \times P}$ is given by

$$\mathbf{O}_i = \mathbf{t}_{i,1:L} \oplus \mathbf{t}_{i,2:L+1} \oplus \cdots \oplus \mathbf{t}_{i,P:m} \tag{1}$$

where $\mathbf{t}_{i,p:p+L-1} \in \mathbf{R}^{L \times 1}$ denotes the L-length subsequence of i-th time series starting at time step p and $\oplus$ denotes the concatenation operator. Then a convolution operation is applied on $\mathbf{O}_i$ along the direction of the length with stride of 1 to obtain the generated shapelets. Let $\mathbf{s}_{i,j} \in \mathbf{R}^{L \times 1}$ denote the j-th generated shapelet conditioned on the i-th time series, then the shaplet $\mathbf{s}_{i,j}$ is defined by

$$\mathbf{s}_{i,j} = \mathbf{W}_j * \mathbf{O}_i + b_j \tag{2}$$

where $\mathbf{W}_j \in \mathbf{R}^{w \times P}$ denotes the j-th filter with w-width, b_j is the bias, $*$ denotes the convolution operation. The activation function is not used here because the generated shapelets are trained to be similar to the actual subsequences and their values should not be limited to the range of an activation function. Similarly, we can generate the top k shapelets of i-th input time series as follows:

$$\mathbf{S}_i = \{\mathbf{s}_{i,1}, \mathbf{s}_{i,2}, \cdots, \mathbf{s}_{i,j}, \cdots, \mathbf{s}_{i,k}\} \tag{3}$$

Hence, multiple sample-specific shapelets are obtained from each input time series.

Dynamic Shapelet Transformation.

After the shapelet generating procedure, the sample-specific shapelets are used to transform the original time series to the new representations, where each attribute is the distance between the original series and one of the shapelets. Let $\mathbf{h} \in \mathbf{R}^{n \times k}$ denote the shapelet-transformed representation, and its each element $h_{i,j}$ denotes the Euclidean distance between $\mathbf{t}_i$ and shapelet $\mathbf{s}_{i,j}$ as follows:

$$h_{i,j} = \min_{p=1,\cdots,P} \sqrt{\sum_{l=1}^{L} (t_{i,p+l-1} - s_{i,j,l})^2} \tag{4}$$

where $s_{i,j,l}$ is the l-th value of shapelet $\mathbf{s}_{i,j}$. Since the shapelets used here are generated dynamically conditioned on input time series, we call this transformation the dynamic shapelet transformation.

Finally, the transformed representation is fed into a softmax layer to obtain the conditional distribution over each category label as follows:

$$\hat{\mathbf{y}}_i = \mathbf{W}_{out} \mathbf{h}_i \tag{5}$$

$$P(C|\mathbf{t}_i) = softmax(\hat{\mathbf{y}}_i) \tag{6}$$

where $\mathbf{h}_i$ is the feature vector generated by Equation 4, $\mathbf{W}_{out}$ is the weights of softmax layer, $\hat{\mathbf{y}}_i$ denotes the output vector and $P(C|\mathbf{t}_i)$ denotes the conditional label distribution of the i-th sample. Dropout is applied to $\mathbf{h}$ to avoid overfitting.

Adversarial Training Strategy.

Since shapelets are discriminative subsequences, the shapelets generated by the shapelet generator should be similar to the actual subsequences. However, if we generate the shapelets without any constraint, the shapelets could be arbitrary shapes. Hence, we need to constrain the shapelets to be similar to real subsequences in order to preserve interpretability. Note that we don't expect that the shapelets should be exactly the same as subsequences, which will result in an all-zero representation after shapelet transformation. We just bridge the gap between the distribution of generated shapelets and subsequences of time series.

To this end, a discriminator D is trained to determine whether the shapelets are generated by the shapelet network, or are subsequences from the actual time series. We model the training process as a two-player minimax game. By adding an adversarial loss, the shapelet generator is trained to generate the shapelets similar to the actual subsequences and to fool the discriminator D. We alternately update the parameters of discriminator D and the parameters of ADSN. This enhances both the discriminator to be more powerful and the generated shapelets to be more similar to subsequences. Formally, the discriminator D is implemented

with a two-layer convolutional neural network and trained by minimizing the following loss function:

$$L_D = -\sum_i \sum_p \log(D(\mathbf{t}_{i,p:p+L-1})) - \sum_i \sum_j \log(1 - D(\mathbf{s}_{i,j})) \tag{7}$$

where $D(\cdot)$ denotes the classification result of the discriminator. To optimize discriminator, the target label is 1 for $D(\mathbf{t}_{i,p:p+L-1})$ and 0 for $D(\mathbf{s}_{i,j})$.

Diversity Regularization Term.

Similar shapelets will lead to similar transformation results, which will create correlated features from the shapelet transformation. However, mode collapse in adversarial training will make the generated shapelets similar to each other. Therefore, enhancing the diversity of shapelets is necessary.

We introduce a diversity regularization term to prevent the model from generating shapelets that are similar to each other and alleviate the mode collapse problem. Specifically, the Frobenius norm of the shapelets similarity matrix is used. The similarities between shapelets of i-th time series are denoted as a matrix $\mathbf{G}_i \in \mathbf{R}^{k \times k}$, where each element of $\mathbf{G}_i(\mathbf{s}_{i,j}, \mathbf{s}_{i,j'})$ denotes the similarity between shapelet $\mathbf{s}_{i,j}$ and shapelet $\mathbf{s}_{i,j'}$. Following the protocol used in previous work (Zhang et al. 2018), a radial basis function (RBF) is employed to calculate the similarity between two shapelets as follows:

$$\mathbf{G}_i(\mathbf{s}_{i,j}, \mathbf{s}_{i,j'}) = \exp(-\frac{d(\mathbf{s}_{i,j}, \mathbf{s}_{i,j'})}{\sigma^2}) \tag{8}$$

where $d(\mathbf{s}_{i,j}, \mathbf{s}_{i,j'})$ is the Euclidean distance between two shapelets and σ denotes the parameter for RBF kernel. The σ are fixed to 1 in our methods.

Overall Loss Function. Finally, the overall training loss L_{ADSN} of ADSN is defined by

$$L_{cls} = -\frac{1}{n} \sum_{i=1}^{n} \sum_{r=1}^{c} 1\{y_{i,r} = 1\} \log \frac{\exp(\hat{y}_{i,r})}{\sum_{l=1}^{c} \exp(\hat{y}_{i,l})} \tag{9}$$

$$L_{div} = \|\mathbf{G}_1 \oplus \mathbf{G}_2 \oplus \cdots \oplus \mathbf{G}_n\|_F^2 \tag{10}$$

$$L_{adv} = -\frac{1}{n \times k} \sum_{i=1}^{n} \sum_{j=1}^{k} \log(D(\mathbf{s}_{i,j})) \tag{11}$$

$$L_{ADSN} = L_{cls} + \lambda_{div} L_{div} + \lambda_{adv} L_{adv} \tag{12}$$

where L_{cls}, L_{div} and L_{cls} are classification loss, diversity regularization term and adversarial loss, respectively. $\mathbf{y}_i$ is the target label of the i-th sample, c is the number of categories, $\mathbf{G}_i$ is defined by Equation 8, $\|\cdot\|_F^2$ is the Frobenius norm, λ_{div} and λ_{adv} are the regularization parameters.

In practice, we find that the discriminator can easily distinguish the generated shapelets from the actual subsequences. Thus, we alternately optimize the discriminator once and ADSN three times to ensure the training process is stable. Also, the number of actual subsequences P of a sample time series depends on the length of the sample. P may be much larger or much smaller than the number of generated shapelets k (a hyperparameter) for some datasets, which also leads the training of discriminator to be unstable. Therefore, we randomly sample k subsequences for each sample to decrease P if $P > k$, while copying more subsequences when $P < k$. In this way, we can ensure there is a similar number of subsequences compared to the shapelets.

Experiments

We conduct experiments on the 85 UCR (Chen et al. 2015) and 8 UEA (Hills et al. 2014) time series datasets. Due to the space limitation, we follow the protocol used in previous shapelet-based papers (Grabocka et al. 2014; Lines et al. 2012; Rakthanmanon and Keogh 2013; Hou, Kwok, and Zurada 2016) and only report the results on 18 UCR datasets and 8 UEA datasets in this section. The full results are shown in section G of the supplementary material. Each data set was split into training and testing set using the standard split. The statistics of these 26 datasets are shown in section A of the supplementary material.

The width of the convolutional filter in the generator and discriminator are fixed to 3. The number of channels of the first and second convolution layer in the discriminator are 16 and 32, respectively. λ_{div} and λ_{adv} are set to 0.01 and 0.05, respectively. The hyper-parameters of ADSN are tuned through a grid search approach based on cross validation. The number of shapelets is chosen from $k \in \{30, 60, 90, 120\}$. The dropout rate applied to the softmax layer is evaluated over $\{0, 0.25, 0.5\}$. We choose the shapelet lengths according to the length of the time series. In fact, we use two shapelet lengths for each time series, i.e., L is better denoted as L_1 and L_2. We try a variety of lengths (L_1, L_2) from the set $\{(0.1, 0.2), (0.2, 0.3), \cdots, (0.7, 0.8)\}$. Each value is a fraction of the time series length (e.g., $L = (0.1, 0.2)$ means 10% and 20% of the time series length). The hyperparameters of ADSN on each dataset are shown in section A of the supplementary material.

The experiments are run on the TensorFlow platform using an Intel Core i7-6850K, 3.60-GHz CPU, 64-GB RAM and a GeForce GTX 1080-Ti 11G GPU. The Adam (Kingma and Ba 2014) optimizer is employed with an initial learning rate of 0.001. The pseudo code of ADSN is shown in section F of the supplementary material. The supplementary material mentioned in this paper is available on github[1].

Comparison with Shapelet-based Methods

ADSN is compared with 6 shapelet-based methods, including the fast shapelet algorithm (FSH) (Rakthanmanon and Keogh 2013), the scalable discovery algorithm (SD) (Grabocka, Wistuba, and Schmidt-Thieme 2015), learning time-series shapelets (LTS) (Grabocka et al. 2014), the ultra-fast shapelet algorithm (UFS) (Wistuba, Grabocka, and Schmidt-Thieme 2015), shapelet transformation with linear SVM classifier (IGSVM) (Hills et al. 2014) and the fused lasso generalized eigenvector method (FLAG) (Hou, Kwok, and Zurada 2016). The details of these methods are described in section B of the supplementary material. The results of these shapelet-based methods are collected

[1] https://github.com/qianlima-lab/ADSN.

Table 1: Accuracy of ADSN and 6 shapelet-based methods.

Dataset	FSH	SD	IGSVM	FLAG	UFS	LTS	w/o div&adv	w/o adv	w/o div	ADSN
Adiac	57.3	52.2	23.5	75.2	69.8	51.9	77.0	**80.1**	79.3	79.8
Beef	50.0	50.0	90.0	83.3	66.7	76.7	86.7	90.0	90.0	**93.3**
Chlorine.	58.8	59.6	57.1	76.0	73.8	73.0	87.7	87.7	87.0	**88.0**
Coffee	92.9	96.4	**100.0**	**100.0**	96.4	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Diatom.	87.3	86.6	93.1	96.4	95.8	94.2	96.7	97.8	98.0	**98.7**
DP_Little	60.6	55.7	66.6	68.3	67.4	**73.4**	71.6	71.3	70.7	72.7
DP_Middle	58.8	55.3	69.5	71.3	66.5	74.1	74.3	74.1	77.4	**78.4**
DP_Thumb	63.4	54.4	69.6	70.5	68.6	**75.2**	71.8	72.1	73.5	73.6
ECGFiveDays	99.8	91.5	99.0	92.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
FaceFour	92.0	83.0	**97.7**	90.9	93.2	94.3	95.5	95.5	95.5	**97.7**
GunPoint	94.0	93.1	**100.0**	96.7	98.7	99.6	96.7	97.3	97.7	98.7
ItalyPower.	91.0	88.0	93.7	94.6	94.0	95.8	96.2	96.8	96.9	**97.2**
Lightning7	65.2	65.2	63.0	76.7	68.5	79.0	79.5	79.6	76.7	**80.8**
MedicalImages	64.7	66.0	55.2	71.4	71.1	71.3	71.4	71.7	71.6	**72.0**
MoteStrain	83.8	78.3	88.7	88.8	87.2	90.0	87.5	**90.9**	89.0	90.6
MP_Little	56.9	62.7	70.7	69.3	71.7	74.3	75.7	75.7	75.5	**75.8**
MP_Middle	60.3	64.5	76.9	75.0	74.8	77.5	78.3	**79.8**	78.1	79.1
Otoliths/Herring	60.9	64.1	64.1	64.1	57.8	59.4	67.2	**70.3**	65.6	**70.3**
PP_Little	57.6	55.8	72.1	67.1	66.8	71.0	71.0	71.4	**74.1**	71.5
PP_Middle	61.6	60.5	75.9	73.8	75.4	74.9	78.1	77.4	78.4	**78.6**
PP_Thumb	55.8	61.8	**75.5**	67.4	67.2	70.5	69.1	71.2	69.3	69.5
Sony.	68.6	85.0	92.7	**92.9**	79.0	91.0	91.0	91.2	91.3	91.5
Symbols	92.4	86.5	84.6	87.5	88.8	94.5	94.8	95.5	95.2	**96.3**
SyntheticC.	94.7	98.3	87.3	99.7	99.7	97.3	99.7	99.7	99.3	**100.0**
Trace	**100.0**	96.0	98.0	99.0	96.0	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
TwoLeadECG	92.5	86.7	**100.0**	99.0	83.6	**100.0**	97.9	98.4	97.4	98.6
AVG rank	8.6	9.1	6.2	6.0	7.1	5.0	4.3	3.1	3.7	**2.0**
best	1	0	5	2	1	6	3	7	5	**15**
p-value	1.23E-05	8.30E-06	4.91E-04	2.86E-05	1.82E-05	2.35E-03	2.68E-05	8.05E-03	3.29E-04	-

from (Hou, Kwok, and Zurada 2016) (AAAI-2016). To verify the effectiveness of the diversity regularization term and the adversarial training, we also show a comparison between the full ADSN model and its three ablation models: 1) ADSN without adversarial loss (w/o adv); 2) ADSN without diversity regularization (w/o div); 3) ADSN without diversity regularization and adversarial loss(w/o div&adv).

As shown in Table 1, ADSN achieves the best results on 15 of the 26 datasets and also the highest average rank of 2.0. Among the other classifiers, LTS performs better by learning the shapelets jointly with the classifier. Due to the dynamic generation of shapelets, ADSN outperforms LTS. The full ADSN model is always superior to all of its ablations, demonstrating the effectiveness of the adversarial training and diversity regularization. To further analyze the performance, we make a pairwise comparison for each shapelet-based method against ADSN. Specifically, we conducted the Wilcoxon signed rank test (Demšar 2006) to measure the significance of the difference. As shown in Table 1, ADSN is significantly superior to all of the other shapelet methods at p <0.01 level.

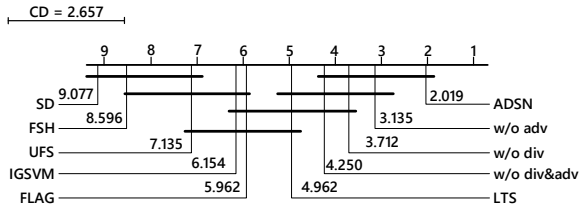


Figure 2: Critical difference diagram over the average rank of ADSN and 6 shapelet-based methods using Nemenyi test.

We also conduct the Nemenyi non-parametric statistical test (Demšar 2006) and plot the critical difference diagram. The results are shown in Figure 2. Classifiers that are not statistically significantly different are joined by horizontal lines. The critical difference is 2.657, which means that two classifiers are not significantly different at p <0.05 level when the rank difference is less than 2.657. While ADSN is numerically superior to the three baseline methods, their performance is not statistically different. However, ASDN is statistically superior to all of the other shapelet methods.

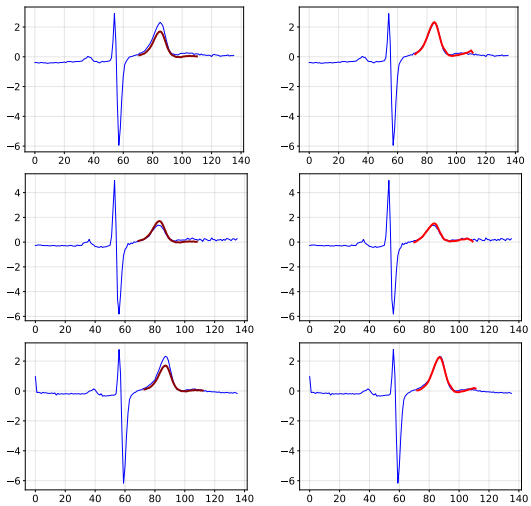
Comparison with State-of-the-art Methods

We further compare ADSN with 11 state-of-the-art methods. These 11 classifiers can be divided into 4 categories:

1. Distance-based methods include derivative DTW (DD_{DTW}) (Górecki and Łuczak 2013) and derivative transform distance (DTD_C) (Górecki and Łuczak 2014);
2. Feature-based methods include bag of SFA symbols (BOSS) (Schäfer 2015), time series fores (TSF) (Deng et al. 2013), time series bag of features (TSBF) (Baydogan, Runger, and Tuv 2013) and learned pattern similarity (LPS) (Baydogan and Runger 2016);
3. Ensemble-based methods include elastic ensembles (EE) (Lines and Bagnall 2015) and collection of transformation ensembles (COTE) (Bagnall et al. 2015);
4. Deep learning methods (Wang, Yan, and Oates 2017) include multilayer perceptrons (MLP), fully convolutional networks (FCN) and residual network (ResNet). The results of these methods are collected from (Bagnall et al. 2017; Wang, Yan, and Oates 2017).

Table 2: Accuracy of ADSN and 11 state-of-the-art methods.

Dataset	DD_{DTW}	DTD_C	BOSS	TSF	TSBF	LPS	EE	COTE	MLP	FCN	ResNet	ADSN
Adiac	70.1	70.1	76.5	73.1	77.0	77.0	66.5	79.0	75.2	**85.7**	82.6	79.8
Beef	66.7	66.7	80.0	76.7	56.7	60.0	63.3	86.7	83.3	75.0	76.7	**93.3**
Chlorine.	70.8	71.3	66.1	72.0	69.2	60.8	65.6	72.7	87.2	84.3	82.8	**88.0**
Coffee	**100.0**	**100.0**	**100.0**	96.4	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
Diatom.	96.7	91.5	93.1	93.1	89.9	90.5	94.4	92.8	96.4	93.0	93.1	**98.7**
ECGFiveDays	76.9	82.2	**100.0**	95.6	87.7	87.9	82.0	99.9	97.0	98.5	95.5	**100.0**
FaceFour	83.0	81.8	**100.0**	93.2	**100.0**	94.3	90.9	89.8	83.0	93.2	93.2	97.7
Gun_Point	98.0	98.7	**100.0**	97.3	98.7	99.3	99.3	**100.0**	93.3	**100.0**	99.3	98.7
Otoliths/Herring	54.7	54.7	54.7	60.9	64.1	57.8	57.8	62.5	68.7	**70.3**	59.4	**70.3**
ItalyPower.	95.0	95.1	90.9	96.0	88.3	92.3	96.2	96.1	96.6	97.0	96.0	**97.2**
Lightning7	67.1	65.8	68.5	75.3	72.6	74.0	76.7	80.8	64.4	**86.3**	83.6	80.8
MedicalImages	73.7	74.5	71.8	75.5	70.5	74.6	74.2	75.8	72.9	**79.2**	77.2	72.0
MoteStrain	83.3	76.8	87.9	86.9	90.3	92.2	88.3	93.7	86.9	**95.0**	89.5	90.6
Sony.	74.2	71.0	63.2	78.7	79.5	77.4	70.4	84.5	72.7	96.8	**98.5**	91.5
Symbols	95.3	96.3	**96.7**	91.5	94.6	96.3	96.0	96.4	85.3	96.2	87.2	96.3
SyntheticC.	99.3	99.7	96.7	98.7	99.3	98.0	99.0	**100.0**	95.0	99.0	**100.0**	**100.0**
Trace	**100.0**	99.0	**100.0**	99.0	98.0	98.0	99.0	**100.0**	82.0	**100.0**	**100.0**	**100.0**
TwoLeadECG	97.8	98.5	98.1	75.9	86.6	94.8	97.1	99.3	85.3	**100.0**	**100.0**	98.6
AVG rank	8.3	8.4	6.8	7.7	8.0	7.6	7.8	4.1	7.9	3.5	4.7	**3.4**
best	2	1	6	0	2	1	1	4	1	**9**	5	**9**
p-value	1.12E-03	1.79E-03	5.38E-03	4.55E-04	1.34E-03	1.92E-03	1.39E-03	9.62E-02	5.03E-04	7.76E-01	2.01E-01	-



(a) LTS (b) ADSN

Figure 3: Three samples from the *ECGFiveDays* dataset and the learned shapelets corresponding to the T-wave.

The details of these methods are also described in section B of the supplementary material.

As shown in Table 2, ADSN achieves the highest average rank of 3.4. Although ADSN and FCN both achieve the best result on 9 datasets, ADSN is numerically superior in average rank. Similarly, we conduct the Wilcoxon signed ranks test. As shown in Table 2, ADSN is significantly better than all the non-deep-learning methods at p <0.01 level except COTE. However, it is noteworthy that COTE ensembles 35 classifiers and thus inevitably suffers from high computational complexity, while ADSN is a single, more elegant model. Although ADSN is superior in average rank, it is not significantly better than FCN and ResNet. These two methods are two deep learning methods containing multiple hidden layers, while ADSN uses only one convolution layer to generate the shapelets and has fewer parameters. More importantly, as a shapelet-based method, ADSN has better interpretability compared to deep learning models.

Visualization Analysis

Dynamic shapelet generation. Shapelets can represent the local patterns existing in time series. However, the shapelets learned by LTS are fixed after training. They cannot deal with the deformations of the local patterns and may fail to recognize the important local patterns at the test phase. In contrast, ADSN can generate different shapelets according to different input time series and improve modeling flexibility. To explore the effectiveness of dynamic shapelet generation, we show the shapelets learned by LTS and ADSN on the $ECGFiveDays$ dataset. Specifically, we randomly choose three samples of class 1 from the test set, and plot these samples and the learned shapelets corresponding to the T-wave. The T-wave of the electrocardiogram represents the period of the heart relaxing and preparing for the next contraction[2]. The learned shapelet corresponding to the T-wave is the shapelet with the smallest distance to the T-wave subsequences.

As shown in Figure 3, since the shapelet learned by LTS is fixed after training, it can not recognize different T-waves of the same category well at the testing phase (fixed shapes for three different samples). In contrast, ADSN can generate different shapelets according to different input time series and improve modeling flexibility. Therefore, the important local patterns of different samples of the same category can be identified, and this makes the samples of the same category classified as the same category more easily. To illustrate this, we employ t-SNE (Maaten and Hinton 2008) to map the original time series and the shapelet transformation representation into a 2-D space, and then visualize the 2-D coordinates. As shown in Figure 4, after the dynamic shapelet transformation of ADSN, the samples of class 1 are

[2]https://en.m.wikipedia.org/wiki/T_wave

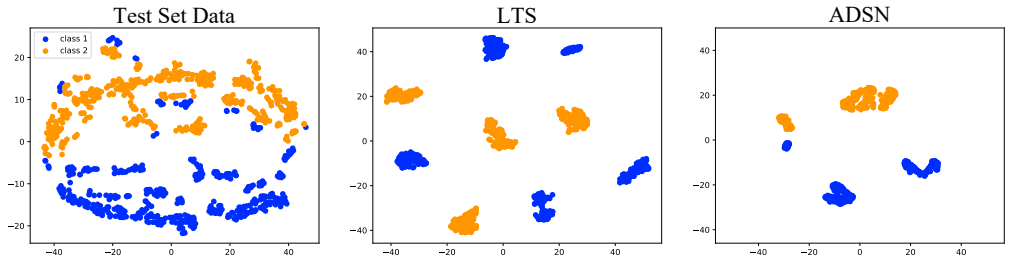


Figure 4: The visualizations with t-SNE on the *ECGFiveDays* datasets.

clustered together more compactly compared to LTS. This is evidence that sample-specific shapelets that are dynamically generated by ADSN improve the modeling flexibility and performance (More visualization analysis verify the effectiveness of dynamic shapelet transformation see Section C of the supplementary material).

Adversarial training. To explore the effectiveness of adversarial training, we first explore the effects of the adversarial training strategy on the shapelets generated by the ADSN. As shown in Figure 7 in the supplementary material, without adversarial training, ADSN will generate shapelets that are dissimilar to the real subsequence. In contrast, the shape of shapelets is more closer to real subsequences when increasing the value of λ_{adv}. The impact of λ_{adv} is quantitatively analyzed in the supplementary material.

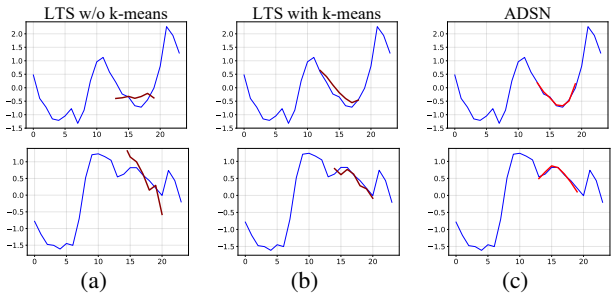


Figure 5: Two samples from the *ItalyPower* dataset and the corresponding shapelet with smallest distance.

Furthermore, we compare the shapelets generated by LTS and ADSN. As shown in Figure 5, we choose 2 samples from the *ItalyPower* dataset, and plot the shapelet with smallest distance to the sample. We call this shapelet the smallest-distance-shapelet. The distances are calculated by Equation 4. In Figure 5(a), without k-means initialization, the smallest-distance-shapelet learned by LTS is dissimilar to the real subsequence. In Figure 5(b), although LTS initializes the shapelets through a k-means clustering of subsequences, there is no constraint on the shape of the shapelets during the training process, and the smallest-distance-shapelet learned by LTS is still dissimilar to the real subsequence. Moreover, using the k-means centroids of all subsequences as the initial value suffers from high computational complexity when the number of time series is large or the sequence is long. In contrast, the smallest-distance-shapelets generated by ADSN are very similar to the real subsequences.

Diversity regularization. To explore the effects of diversity regularization, we show the generated shapelets of ADSN under different values of λ_{div}. As shown in Figure 9 in the supplementary material, the generated shapelets converge to similar shapes when the value of λ_{div} is 0. When we increase the value of λ_{div}, the generated shapelets become more and more dissimilar from each other and diverge to different shapes. This shows that the diversity regularization can increase the diversity of generated shapelets and thus alleviate the mode collapse problem. The impact of λ_{div} is also quantitatively analyzed in the supplementary material.

Conclusion

In this paper, we propose a novel model called Adversarial Dynamic Shapelet Networks (ADSNs) that dynamically generates the shapelets for each time series, and use the adversarial training strategy to restrict the generated shapelets to be similar to the actual subsequences of the samples rather than arbitrary shapes. Previous shapelet learning methods achieve better results than traditional shapelet-based classifiers, but there are some drawbacks to these methods. First, the shapelets are fixed after training and cannot adapt well to time series with deformations. More importantly, the learned shapelets may not be similar to any real subsequences, which will reduce model interpretability. In contrast, ADSN generates sample-specific shapelets via a dynamic generating process, improving model flexibility. The adversarial training strategy is used to restrict the generated shapelets to be similar to the real subsequences. The extensive experimental results verify the performance of the proposed method. Furthermore, the visualization analysis demonstrates the effectiveness of the dynamic shapelet generating and adversarial training.

Acknowledgments

We thank the anonymous reviewers for their helpful feedbacks. The work described in this paper was partially funded by the National Natural Science Foundation of China (Grant Nos. 61502174, 61872148), the Natural Science Foundation of Guangdong Province (Grant Nos. 2017A030313355, 2017A030313358, 2019A1515010768), the Guangzhou Science and Technology Planning Project (Grant Nos. 201704030051, 201902010020).

References

Bagnall, A.; Lines, J.; Hills, J.; and Bostrom, A. 2015. Time-series classification with cote: the collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering* 27(9):2522–2535.

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.

Baydogan, M. G., and Runger, G. 2016. Time series representation and similarity based on local autopatterns. *Data Mining and Knowledge Discovery* 30(2):476–509.

Baydogan, M. G.; Runger, G.; and Tuv, E. 2013. A bag-of-features framework to classify time series. *IEEE transactions on pattern analysis and machine intelligence* 35(11):2796–2802.

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

Chang, K.-W.; Deka, B.; Hwu, W.-M. W.; and Roth, D. 2012. Efficient pattern-based time series classification on gpu. In *2012 IEEE 12th International Conference on Data Mining*, 131–140. IEEE.

Chen, Y.; Keogh, E.; Hu, B.; Begum, N.; Bagnall, A.; Mueen, A.; and Batista, G. 2015. The ucr time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/.

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

Deng, H.; Runger, G.; Tuv, E.; and Vladimir, M. 2013. A time series forest for classification and feature extraction. *Information Sciences* 239:142–153.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Górecki, T., and Łuczak, M. 2013. Using derivatives in time series classification. *Data Mining and Knowledge Discovery* 26(2):310–331.

Górecki, T., and Łuczak, M. 2014. Non-isometric transforms in time series classification using dtw. *Knowledge-Based Systems* 61:98–108.

Grabocka, J.; Schilling, N.; Wistuba, M.; and Schmidt-Thieme, L. 2014. Learning time-series shapelets. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 392–401. ACM.

Grabocka, J.; Wistuba, M.; and Schmidt-Thieme, L. 2015. Scalable discovery of time-series shapelets. *arXiv preprint arXiv:1503.03238*.

Hills, J.; Lines, J.; Baranauskas, E.; Mapp, J.; and Bagnall, A. 2014. Classification of time series by shapelet transformation. *Data Mining and Knowledge Discovery* 28(4):851–881.

Hou, L.; Kwok, J. T.; and Zurada, J. M. 2016. Efficient learning of timeseries shapelets. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Jia, X.; De Brabandere, B.; Tuytelaars, T.; and Gool, L. V. 2016. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, 667–675.

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

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

Lines, J., and Bagnall, A. 2015. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery* 29(3):565–592.

Lines, J.; Davis, L. M.; Hills, J.; and Bagnall, A. 2012. A shapelet transform for time series classification. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 289–297. ACM.

Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of machine learning research* 9(Nov):2579–2605.

Rakthanmanon, T., and Keogh, E. 2013. Fast shapelets: A scalable algorithm for discovering time series shapelets. In *proceedings of the 2013 SIAM International Conference on Data Mining*, 668–676. SIAM.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, 2234–2242.

Schäfer, P. 2015. The boss is concerned with time series classification in the presence of noise. *Data Mining and Knowledge Discovery* 29(6):1505–1530.

Shah, M.; Grabocka, J.; Schilling, N.; Wistuba, M.; and Schmidt-Thieme, L. 2016. Learning dtw-shapelets for time-series classification. In *Proceedings of the 3rd IKDD Conference on Data Science, 2016*, 3. ACM.

Wang, Z.; Yan, W.; and Oates, T. 2017. Time series classification from scratch with deep neural networks: A strong baseline. In *2017 International Joint Conference on Neural Networks (IJCNN)*, 1578–1585.

Wistuba, M.; Grabocka, J.; and Schmidt-Thieme, L. 2015. Ultra-fast shapelets for time series classification. *arXiv preprint arXiv:1503.05018*.

Ye, L., and Keogh, E. 2009. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 947–956. ACM.

Zhang, Q.; Wu, J.; Yang, H.; Tian, Y.; and Zhang, C. 2016. Unsupervised feature learning from time series. In *IJCAI*, 2322–2328.

Zhang, Q.; Wu, J.; Zhang, P.; Long, G.; and Zhang, C. 2018. Salient subsequence learning for time series clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.