Interpretable Time-series Classification on Few-shot Samples

1st Wensi Tang
Centre for AI, FEIT
University of Technology Sydney
Sydney, Australia
Wensi.Tang@student.uts.edu.au

2nd Lu Liu
Centre for AI, FEIT
University of Technology Sydney
Sydney, Australia
Lu.Liu-10@student.uts.edu.au

3rd Guodong Long
Centre for AI, FEIT
University of Technology Sydney
Sydney, Australia
Guodong.Long@uts.edu.au

Abstract—Recent few-shot learning works focus on training a model with prior meta-knowledge to fast adapt to new tasks with unseen classes and samples. However, conventional time-series classification algorithms fail to tackle the few-shot scenario. Existing few-shot learning methods are proposed to tackle image or text data, and most of them are neural-based models that lack interpretability. This paper proposes an interpretable neural-based framework, namely *Dual Prototypical Shapelet Networks (DPSN)* for few-shot time-series classification, which not only trains a neural network-based model but also interprets the model from dual granularity: 1) global overview using representative time series samples, and 2) local highlights using discriminative shapelets. In particular, the generated dual prototypical shapelets consist of representative samples that can mostly demonstrate the overall shapes of all samples in the class and discriminative partial-length shapelets that can be used to distinguish different classes. We have derived 18 few-shot TSC datasets from public benchmark datasets and evaluated the proposed method by comparing with baselines. The DPSN framework outperforms state-of-the-art time-series classification methods, especially when training with limited amounts of data. Several case studies have been given to demonstrate the interpret ability of our model.

Index Terms—time series classification, few-shot learning, interpretability

I. INTRODUCTION

Training a classification model with very few labeled training samples, namely few-shot classification, has attracted broad interest recently. In particular, the few-shot classification model should be able to categorise unseen samples to a class according to the generalised concepts and knowledge that is extracted from very few seen examples (training samples) in this class.

Time-series classification has been broadly applied in intelligent-based real-world applications, including heart disease diagnoses (e.g., ECG200 data set), motion detection (e.g., GunPoint data set), traffic analysis (e.g., MelbournePedestrian data set), and earthquake prediction (e.g., Earthquakes data set), etc. However, conventional time-series classification methods fail to tackle the few-shot scenario. Solving the few-shot time-series classification problem will not only result in the development of a fast adaptive model for time-series data but also will benefit applications that have difficulty in acquiring enough labeled samples. For instance, both using functional MRI to provide a voxel-level damage assessment
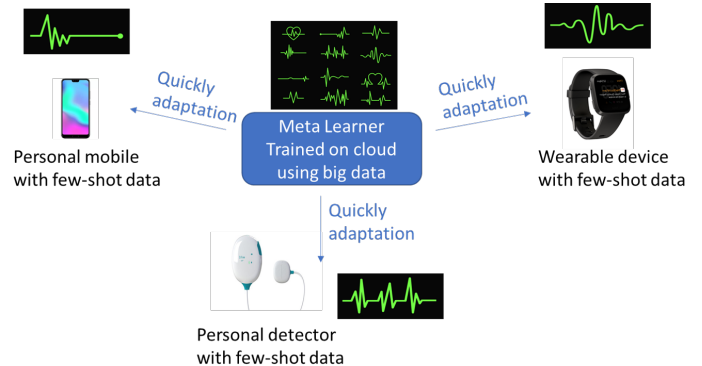

Fig. 1. Application scenarios for few-shot time-series classification on wearable devices and smartphones. Each end-user can use his or her own data to train a private intelligent model while preserving privacy.

for an Alzheimer's disease patient [1] and using a mobile EEG headset to provide audio feedback on a learner's attention level [2] suffer from insufficient supervision information.

Recently proposed neural network-based few-shot learning models achieve state-of-the-art performance in many image and text classification related tasks [3]–[5]. Few-shot time series classification is a new area with a promising future due to the following: firstly, with the development of the Internet of Things (IoT) and edge computing, numerous edge devices are deployed to continuously collect environment data which is usually in time series format, then these devices can make intelligent decisions according to its own data [6]. The smart applications in edge devices can be regarded as a typical problem of few-shot time-series classification [7]. Secondly, the rising privacy concerns on personal data, especially the European Union published General Data Protection Regulation (GDPR), will push the intelligent service from big data centering model towards the few-shot distributed personal model. As shown in Figure 1, each end-user should have a private model that is trained with the user's own few-shot data. Lastly, the wide-spreading use of wearable devices and smartphones enables the development of an intelligent service which is based on the end-users' daily time-series data, e.g. medical time series, moving speed, and sleeping quality [8], [9].

Model interpretability is another challenge when addressing

this problem for the reason that most neural-based models are black-box models that lack interpretability. A reasonable interpretation is critically important to convince end-users that the trained few-shot model has a good generalisation capability even though the model is trained with very few training samples.

This paper proposes a novel Dual Prototypical Shapelet Network (DPSN) that can simultaneously solve the aforementioned few-shot time-series classification problem and address the model interpretability challenge. In particular, the classification model is built on an end-to-end framework that includes a KNN-based lazy learner, and a neural layer which is applied to learn the metrics to transform time series data into a new space. Dual interpretation of the classifier will be generated by two different model interpretability techniques. The first type of interpretation is a selected time-series example, namely a representative shapelet, to be used as a prototype to demonstrate the overall shape of all the samples in a class. The other interpretation output is a sub-sequence of a time-series example, namely discriminative shapelet, that can be used to distinguish different classes. The framework of DPSN is shown in Figure 2.

The paper's main contributions are summarised as follows:

- The first study solves the few-shot time-series classification task with neural-based methods.
- It proposes a novel dual prototypical shapelet network that not only outperforms several state-of-the-art time-series classification methods in a few-shot scenario but can also generate interpretation.
- The proposed dual interpretation method is a novel interpretability one for time-series classification.

The remainder of the paper is organised as follows. Section II introduces the related work. The methodology is described in Section III followed by Section IV with the experiment results, and the final part is conclusion in Section VI.

II. RELATED WORKS

A. Time Series Classification

Time series classification is a classical research topic with various solutions. For example, the window method is widely used to extract distinctive sub-sequences to avoid the alignment issue. New distance metrics, such as DTW distance [10], are introduced to mitigate the influences caused by the length variance of the discriminate sub-sequence and alignment problem. Discrete Fourier Transform (DFT) is used to minimize the alignment and noise problem by mapping time series data into a frequency domain. Other sophisticated methods include a tree-based ensemble method (random forest) and SVM with a quadratic and cubic kernel, etc [11]. Based on the techniques mentioned above, many algorithms have been proposed. From the University of California, Riverside (UCR) time series classification archive [12], WEASEL, ST and BOSS are three SOTA traditional algorithms for TSC. There are also numerous deep learning-based methods to solve TSC tasks [13], [14].

B. Few-shot Learning

In the pre-deep learning era, data is not collected systematically so that few-shot learning is proposed to target one task with limited samples. Bayesian model is used to encode the priors, which is a reflection of the object category distribution [15]. Attribute annotations of every category can help to alleviate the few-shot challenge even if no sample is available for every class, which is called zero-shot learning [16].

Data repositories are accumulating gradually. With the advent of datasets such as ImageNet [17] and the improvement of deep neural networks such as ResNet [18], comes the big data era. Few-shot learning in the big data era focuses on the setting of continual learning or life-long learning, where new classes or new tasks with limited samples are considered as the main challenge for few-shot learning. To meet this challenge, researchers apply the idea of meta learning where a meta-learner extracts the common knowledge over various tasks or classes so that this meta knowledge can be transferred to new tasks with few samples. The meta learner can grasp a shared compelling initialization for learner models [19], a shared metric space for K-nearest neighbors search [4], a graph [20], [21], an extra memory of representative feature maps [22] and an optimization strategy for fast adaptation [23]. The ideas have been broadly used in applications for computer vision [24], natural language processing [25] and health [26] domains.

Most of the meta-learning based approaches for few-shot learning mainly target the task of image classification. Images can be collected from websites or camera devices and image datasets have vast amounts of data, e.g., ImageNet has more than 14 million hand-annotated images; The CIFAR-10 and CIFAR-100 are labeled subsets of the 80 million tiny images dataset [27]. However, time-series data has limited data sources and the annotations often require expertise knowledge, which has a higher cost. Public time-series datasets usually have less than 1,000 samples, e.g., Gunpoint, ECG200, Wine etc. Privacy time-series datasets may have even fewer samples. As a result, we define our problem as few-shot time-series learning on the small data area, where we focus on one task with limited samples instead of a continual learning setting with the idea of meta learning. To the best of our knowledge, we are the first to focus on the problem of few-shot time-series learning.

C. Interpretability for Neural-based Model

Interpretability is critically important for neural-based models due to its black-box nature. Conventional model interpretation analyses the model's parameters and outcomes to generate an interpretation for the model, for example, [28] analysis feature importance for generating results, explaining the model using prototype examples [29] or influential examples [30], and discovering influential input patterns using activation maximisation criteria [31]. Gradient analysis is also important for model interpretation, e.g. Layer-Wise Relevance Propagation [32], and gradient sensitivity analysis [31]. Many research
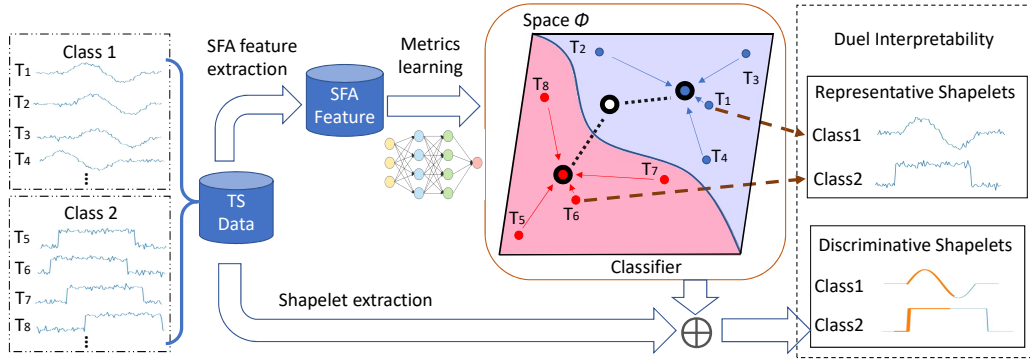
Fig. 2. The framework of DPSN. The framework contains three components: 1) feature extraction will transform the Time series into an SFA feature and Shapelet feature; 2) The classification part will build a prototype of each class by aggregating SFA feature via a metric learning neural network; 3) The dual interpretability has two-fold explanations including the representative shapelet which was identified by a prototype from the classification part, and discriminative shapelets which can be learned by combining information from shapelet feature and prototype.

works have tried to modify the existing DNN/CNN/RNN-based or attention-based neural model to generate intermediate information for model interpretation.

Another research direction for model interpretability is to treat the original model as a black-box, then use another interpretable model to simulate or approximate the original model without opening it. For example, [33] distill a black box neural network into a soft decision tree that is an inherently interpretable model [34] and generate a decision tree to interpret the RNN-based deep model for time series classification.

III. DUAL PROTOTYPICAL SHAPELET NETWORKS

Following most recent time series learning works [35]–[37], we learn from training dataset $\mathcal{T} = \{T_1, ..., T_N\}$ annotated by labels $\mathcal{Y} = \{y_1, ..., y_N\}$. N is the number of training data. We use K to denote the number of classes. We undertake classification in more challenging scenarios where the number of training samples for every class is reduced to 2, i.e., $\|\mathcal{T}^k\|$=2. We do not test when $\|\mathcal{T}^k\|$=1 for it makes no sense to learn a prototype from one sample without prior knowledge. The overall framework of the proposed DPSN method is shown in Figure 2. The whole procedure is divided into two stages: feature extraction using SFA (Sec. III-A) and classification using a prototypical network (Sec. III-B).

Algorithm 1 Training process for DPSN

Input: SFA feature $\mathcal{X}$, Label $\mathcal{Y}$; Transformation net $f_\phi(\cdot)$. Number of samples for classification in every iteration N_q.
Output: Trained transformation net $f_\phi(\cdot)$; Prototype of each class $\{C_k\}_{k=1}^K$.
1: Initialize f_ϕ with random weight; $L \leftarrow 0$
2: **while** not converge **do**
3: Sample a batch of features $\mathcal{X}^k$ for every class $k \in \mathcal{Y}$
4: Generate C_k for every class using Equation 1
5: For each class k, randomly pick N_q samples from $\mathcal{X}^k$
6: Calculate loss using these examples by Equation 3
7: Apply Mini-batch SGD to minimize L and update ϕ,
8: **end while**

A. SFA Feature Extraction

The symbolic Fourier approximation(SFA) [38] word histogram is a handcrafted feature, which has achieved promising results in time series classification [11], [36]. However, typically, the SFA word histogram is sparse and high-dimensional. Thus, we do some dimension reduction on the histogram. We remove those words from histogram if their values are all 0 for each training data. In testing, we do not count words, which have already been removed. In brief, we transform the time series to dense SFA features i.e., $\mathcal{T} \rightarrow \mathcal{X} = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_N\}$, for classification below.

SFA word histogram requires two hyperparameter sampling window size L_S and a $\boldsymbol{w}$ for discrete Fourier transform (DFT) noise reduction. We follow the method in [11] to pick these two hyperparameters.

B. Prototypical Network

We use a prototypical network, which is originally introduced to tackle the few-shot image classification task, to solve the few-shot time series classification problem. The prototype network is a KNN based classifier, which classifies samples of the class with the nearest prototype. Prototype C_k for class k is defined by the feature average of $\mathcal{X}^k$:

$$C_k = \frac{1}{\|\mathcal{X}^k\|} \sum_{x \in \mathcal{X}^k} f_\phi(\boldsymbol{x}), \qquad (1)$$

where $\mathcal{X}^k$ denotes SFA features of sampled training samples in class k, $\boldsymbol{x}$ is SFA feature for a time series and $f(\cdot)$ is the feature extractor parameterised by ϕ.

Even though most time series networks use 1d CNN for feature transformation f_ϕ, we prefer a fully connected layer to map the feature from the SFA domain into the prototype domain. There are two reasons to use the SFA feature. Firstly, the SFA feature is a time invariance feature. Thus, we don't have to use CNN for it can obtain a time invariance feature. Secondly, the SFA feature is a handcrafted feature, which makes it is easy to know what frequencies are denoised during feature extraction. And this characteristic is essential for shapelet finding.

Classification result is calculated as a softmax over all classes $k' \in \mathcal{Y}$ using Euclidean distance between a sample and every prototypes:

$$\Pr(y = k|\boldsymbol{x}) = \frac{\exp(-\|f_\phi(\boldsymbol{x}) - \boldsymbol{C}_k\|^2)}{\sum_{k'} \exp(-\|f_\phi(\boldsymbol{x}) - \boldsymbol{C}_{k'}\|^2)} \quad (2)$$

The cross entropy loss L of data sampled from the distribution $\mathcal{D}$ can be formulated as:

$$L = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} - \log \Pr(y|\boldsymbol{x}; \phi) \quad (3)$$

More details for this training strategy are shown in Algorithm. 1.

C. Interpret Model Using Representative Shapelets

As discussed in the training process subsection, the classifier will discover an aggregated central point as a prototype that can mostly represent all samples in the class. Intuitively, if we could transform the SFA feature-based prototype to a time series format, we can obtain a good representative sample of the class. However, this operation is impossible because the SFA features are not derivative. Therefore, we choose a representative example by finding the time series example which is the nearest neighbour of the prototype in space ϕ. This representative sample is the prototype approximation that can demonstrate the overall shapes of all supportive samples in the class. These representative shapelets for classes are a type of explanation for the trained DPSN classifier.

Given a class k, the prototype is generated by simply calculating the weighted sum of all supportive samples in the class, and the representative sample is selected using the following equation

$$T_i, i = \arg \min_i F(f_\phi(x_i), \boldsymbol{C}_k) \quad (4)$$

where F is the distance function which could be Euclidean distance or another, e.g. cosine dissimilarity, or KL divergence. The distance calculation is based on the representation in space ϕ. Thus, in our work, we use Euclidean distance as it is easy to understand by humans [39].

D. Interpret Model Using Discriminative Shapelets

When people introduce or describe a particular object, they need to use specific or unique attributes that can enable the audience to distinguish the object from others. A time-series sample includes numerous sub-sequences, each of which consists of a shapelet and a noise. Given a class, its discriminative shapelet should 1) exist in every same-class sample, and 2) not exist in any diff-class sample. We denote the representative sample by T and extract many sub-sequence T^i using a slider window. The shapelet of T^i is S^i, discriminative shapelets are denoted as S.

Given the class k, its best or optimal discriminative shapelet S_k can be found by maximising the likelihood in Equation 5.

$$S_k = \arg \max_i L_o(S^i, \mathcal{T}, \mathcal{Y}^k) \quad (5)$$

Algorithm 2 Discriminative Shapelet Discovery

Input: Time series data $\mathcal{T}$, SFA feature $\mathcal{X}$, Label $\mathcal{Y}$, well trained linear transformation network $f_\phi(\cdot)$
Output: discriminative shapelet of each class $\{S_k\}_{k=1}^K$
1: calculating $\{C_k\}_{i=1}^K$ by Equation 1
2: **for** C_k in $\{C_k\}_{i=1}^K$ **do**
3:     Identity T by Equation 4
4:     **for** each sub-sequence T^i in T **do**
5:         Applying DFT on T^i
6:         Use the first w complex number to reconstruct S^i
7:         Calculate $D = [d_{S^i T_1}, ..., d_{S^i T_N}]$ by Equation 9
8:         Calculate f-test score of D and Y^k
9:     **end for**
10:    save the S^i with highest f-test score as S_k
11: **end for**
12: **Return**

---

where $\mathcal{T}$ and $\mathcal{Y}^k$ combine to product a bi-classification dataset where the positive class is the given class k and the negative class is all other classes. The likelihood function needs to be aligned to the aforementioned two requirements of discriminative shapelets. In particular, we measure the likelihood using f-test as demonstrated in Equation 6.

$$L_o = \frac{between \; class \; variability}{within \; class \; variability} \quad (6)$$

where the large value of "between-class variability" indicates that the selected shapelet does "not exist" in the diff-class samples, and the small value of the "within-class variability" indicates that the shapelet is likely to "exist" in the same-class samples. The shapelet has the largest f-test value is the discriminative shapelet.

The "between-class variability" measures the distance between different classes using the following equation:

$$\frac{\sum_{i=1}^K N_i * (\bar{d}_i - \bar{d})^2}{K - 1} \quad (7)$$

where N_i is the number of samples i the $i-th$ class, $\bar{d}_i$ denotes the mean value of all samples in the $i-th$ class, $\bar{d}$ denotes the overall mean of all classes, and K is the number of classes that is 2 in the bi-classification scenario.

The "within-class variability" calculates the variance of all the distance-values in the same class using the following equation.

$$\frac{\sum_{i=1}^K \sum_{j=1}^{N_i} (d_{ij} - \bar{d}_i)^2}{N - K} \quad (8)$$

As discussed, d is the value to estimate the likelihood. d is the distance between a shapelet and a time-series example. In particular, a time-series T will be transformed to a set of subsequences T^i that is the same length as shapelet S. As we expect to check whether or not the shapelet exists in the time-series, distance d_{ST} is the minimal distance between S and the set of subsequences T^i. as calculated by Equation 9.

$$d_{ST} = \min_{i=1...ns} F(T^i, S) \quad (9)$$

Classifier(ways, original training size)	6-shots Accuracy(std) %			8-shots Accuracy(std) %		
	DPSN	BOSS	ST	DPSN	BOSS	ST
ArrowHead(3-way, 36)	**78.72(3.72)**	67.6(7.2)	51.26(7.68)	**81.34(2.32)**	74.46(4.54)	47.77(9.57)
BME(3-way, 30)	72.0(8.93)	**82.73(2.03)**	64.07(4.74)	74.36(8.53)	**86.13(2.73)**	69.2(4.9)
CBF(3-way, 30)	**97.58(1.67)**	96.1(1.96)	89.76(1.7)	**97.22(1.73)**	95.99(1.46)	88.63(2.12)
Chinatown(2-way, 20)	**80.41(3.11)**	78.55(4.7)	71.59(18.36)	**83.38(2.16)**	81.68(2.64)	71.71(16.65)
ECG200(2-way, 100)	**83.57(2.07)**	75.9(5.39)	65.6(3.01)	**86.52(1.36)**	78.0(4.45)	67.1(3.73)
GunPoint(2-way, 50)	**95.42(1.71)**	84.53(4.42)	79.87(7.51)	**95.85(1.75)**	88.0(3.27)	58.73(11.27)
GunPointAgeSpan(2-way, 135)	**86.89(3.94)**	80.57(4.27)	63.64(15.04)	**86.89(3.94)**	80.57(4.27)	63.64(15.04)
GunPointOldVersusYoung(2-way, 135)	**81.3(8.72)**	77.24(5.94)	68.83(9.33)	**81.3(8.72)**	77.24(5.94)	68.83(9.33)
ItalyPowerDemand(2-way, 67)	**80.89(6.25)**	78.81(3.94)	62.61(11.46)	**80.89(6.25)**	78.81(3.94)	62.61(11.46)
MoteStrain(2-way, 20)	**82.41(2.39)**	79.03(2.13)	77.87(3.93)	**87.32(2.78)**	80.62(1.19)	79.72(2.46)
Plane(7-way, 105)	**99.75(0.53)**	99.24(0.71)	55.43(10.61)	94.34(1.74)	**99.52(0.64)**	53.43(9.43)
SonyAIBORobotSurface1(2-way, 20)	55.12(2.61)	53.71(9.33)	**58.5(6.87)**	55.93(2.76)	54.19(9.34)	**58.84(7.26)**
SonyAIBORobotSurface2(2-way, 27)	**81.22(2.1)**	79.72(1.58)	68.9(2.81)	**84.1(1.32)**	83.06(1.76)	75.02(3.59)
SyntheticControl(6-way, 300)	76.47(5.21)	76.57(1.2)	**76.73(8.62)**	83.96(2.4)	79.8(1.63)	**87.53(7.17)**
ToeSegmentation1(2-way, 49)	**96.18(1.15)**	85.39(3.45)	77.54(5.06)	**96.48(0.78)**	85.96(4.54)	79.91(4.13)
TwoLeadECG(2-way, 23)	**93.48(1.69)**	89.8(3.02)	75.83(9.27)	**95.01(0.53)**	91.43(2.0)	74.84(4.86)
UMD(3-way, 36)	85.62(2.62)	**85.9(5.96)**	85.56(4.78)	89.22(4.1)	**90.97(2.34)**	86.88(5.41)
Wine(2-way, 57)	**64.52(6.75)**	57.22(6.17)	50.0(0.0)	**66.35(7.33)**	57.41(3.88)	50.0(0.0)
average accuracy Rank	**1.25**	2.25	2.5	**1.19**	2.33	2.48
average std Rank	**1.55**	1.95	2.5	**1.71**	1.86	2.43
Wins	**14**	2	4	**13**	3	4

TABLE I
THE ACCURACY(STANDARD DEVIATION) RESULT OF 18 DATASETS IN 4-SHOT AND 8-SHOT SCENARIO, AND STATISTICAL RESULTS OF AVERAGE ACCURACY AND STANDARD DEVIATION RANK. THE WINS ROW IS THE TOTAL NUMBER OF BEST ACCURACY IN 18 DATASETS

where F is the distance function, ns is the number of generated sub-sequences from time-series T.

1) Procedure for discovering discriminative shapelets: To discover the discriminative shapelets, we use the following four steps.

- Extract sub-sequences from the selected time series data using a sliding window.
- Extract signals from the sub-sequences using DFT.
- Measure the shapelet distance between each signal and the time series in the training data.
- Use f-test to find the best signal as the discriminative shapelets.

The pseudo code for discovering discriminative shapelets is described in Algorithm 2

IV. EXPERIMENTS

A. Dataset Selection

To experimentally study the proposed model, we chose and derived 18 few-shot time-series classification datasets from UCR [12] dataset which included 128 time-series datasets. In particular, we prefer dataset with limited training samples and short length for each time series. We rank the 128 UCR datasets from UCR by the $length \times size\ of\ the\ training\ sample$ and choose the top 18 datasets to be converted to few-shot sample scenario.

The sampling procedure needs a hyper-parameter named sample ratio which is expressed as a fraction of the total number of training samples. Moreover, for each sample ratio, we will sample them 10 times. In this experiment, we set sample ratio number from 0.1 to 0.9 with the step of 0.1. Thus, for one UCR dataset, we can build 9×10 few-shot datasets from it. We experimented on each few-shot datasets and also on the original UCR dataset.

Specifically, assume the sample ratio is 0.3, and the original dataset has 100 training data, we will randomly sample 30 training data ten times. We will have 10 new datasets and each dataset has 30 training data. The test datasets were kept the same as the test dataset in UCR benchmark.

We do not pre-process the input data, except to replace NaN value with zero for any unequal time series data.

B. Baselines

Two closely related baselines are used in our experiments. We compare our method with BOSS [11], which also uses SFA histograms. We also compare our method to Shapelet-Transform (ST) [41], which also uses shapelets distance to extract the discriminative feature. The source code for BOSS and SFA feature extraction is seen in [1]. The source code of ST from [2].

C. Experiment Setup

Both SFA feature and ST need hyper-parameters like window size and etc. In the original code from UCR dataset, they use cross validation to select the best hyper-parameters. However, for few-shot datasets, the number of training samples can be very small. Thus, it is impossible to do cross validation to get those hyper-parameter. Thus, we use the same hyper-parameter as the cross validation selected in the total dataset.

For prototypical neural networks, the transformation network has two fully connected layers with the neuron number of hidden layer as 256 and the output dimension as 64. We use Adam to train our model for 1000 epochs with the learning rate of 0.002, the learning rate decay of 1e-5, and the momentum of 0.7. We published our code here[3].

---

[1]https://github.com/johannfaouzi/pyts
[2]https://github.com/rtavenar/tslearn/blob/master/tslearn/docs/index.rst
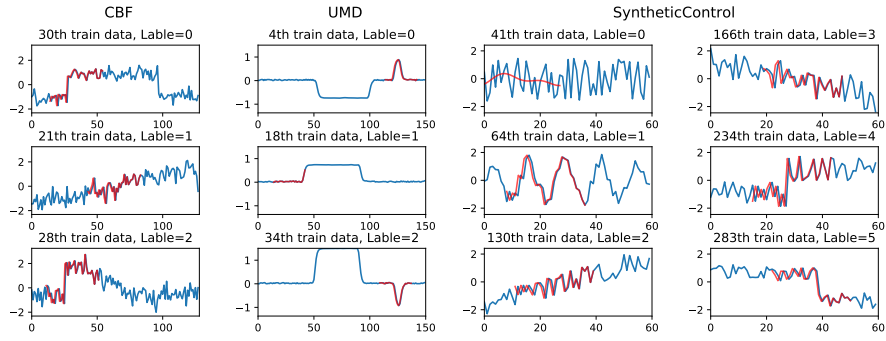[3]https://github.com/Wensi-Tang/DPSN

Fig. 3. This figure shows the interpretability results of three handcrafted dataset. Names of the datasets are above each graph. On the top of each line chart, the title has two numbers ith and label=j. These numbers indicate that the ith data in the training set is selected to represent class j. The blue line is the ith data, while the red line is the discriminative shapelet from our framework.
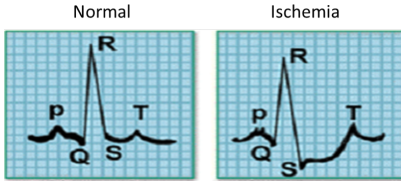

Fig. 4. The visualization of differences between Normal and Ischemia on ECG200 [40]. The lower value of S and long time from S to T is called ST-segment depression, and lower value of R is called invalid amplitude
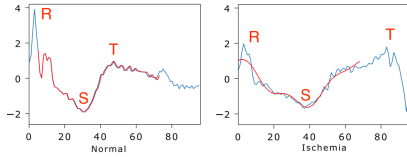

Fig. 5. The visualization of differences between Normal and Ischemia on ECG200 dataset. The red lines indicate the Shapelets found by DPSN, which include the value of S and time interval between S and T. The results match the demonstration shown in Fig. 4

D. Experiment Results

1) Experiment Result of Classification: In Tab I, we demonstrate the 6-shots and 8 shots result of selected 18 few-shot time-series datasets. We can see that when less training data is used, for more than half of 18 datasets, DPSN outperforms baselines on accuracy with lower variance. All results can be found on this link [4].

2) Interpretability of DPSN: For all figures in this paper, horizontal and vertical axes are the same as UCR datasets, and the unit is not labeled as it is not important for the classification task [12]. We demonstrate the interpretability of DPSN in Figure 3. We choose three simulated datasets. Those datasets are chosen because they are easy for humans to understand, and in the following section, we will demonstrate the interpretability of DPNS by using two real case studies. A visualization of all results can be found on this link [5].

E. Classification Result Analysis

The UCR branch has time-series datasets from various sources; thus, it is hard to develop one classification method which could outperform every technique in every dataset. Therefore, in this part, we want to analyze the relationship between DPSN's performance and the characteristic of time-series. This analysis will benefit model selection.

From a visualization of full result on this link[6], we could see that DPSN has a great relationship with BOSS. This relationship can also be seen in Tab. I. The reason is that despite redesigning the distance metric method and classifier, the SFA feature histogram which we use is the same as BOSS. This means that will inherit problems from the SFA feature.

Furthermore, the problem is that if a dataset can be described by the SFA feature, which characteristics of a dataset would help us make a selection between BOSS or DPSN. When compared with the 6-shot and 8-shot results, we could see that only for the BME dataset, the BOSS is always superior to DPSN. This is because the BME dataset has more than one cluster in its class. Thus, learning only one prototype for each class is not sufficient. This kind of problem can be solved by the method described in [42]

Another phenomenon is that despite the expectations of the result of 8-shot to be better than the results of 6-shot, it not true for all datasets. For example, as Tab I shows, for both accuracy and standard deviation, the 6-shot results of Plan dataset is better than that of the 8-shot. This phenomenon can also be found in other works of few-shot learning tasks like the one in [43]. The reason is that the relationship between performance and size of training datasets is not a monotonic function. There are some fluctuations in it.

From the visualization of all results we can find that the DPSN outperforms BOSS and ST for all sample ratios. Which means that the DPSN can be viewed as a general method. However, the reason we still claim our method, is a few-shot method is that the technique is a type of K-NN classifier, which

---

[4]https://github.com/Wensi-Tang/DPSN/blob/master/results/result.txt

[5]https://github.com/Wensi-Tang/DPSN/tree/master/SFA_Python-master/test/image_result

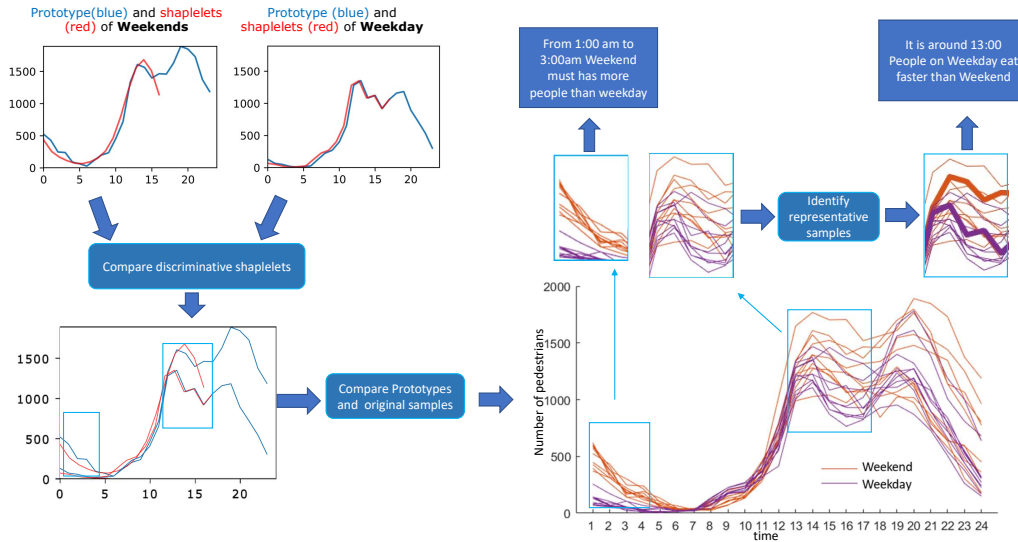[6]https://github.com/Wensi-Tang/DPSN/blob/master/All_classification_result.pdf

Fig. 6. A comparison of shapelets and original samples for Chinatown dataset. The left side shows the prototypes for pedestrians on weekends or weekdays. The important features of the prototypes, which are indicated by the boxes are carefully analyzed together with the original samples on the right side.

practically is not an ideal solution for a dataset consisting of a large amount of training data.

V. CASE STUDIES

This section elaborates the interpretability of our model using several case studies in real-world datasets and applications.

A. Heartbeat Anomaly Detection

In the ECG200 time-series dataset [40], a binary classifier can be trained to detect the anomalous heartbeat time-series wavelets. As shown in Fig. 4, there are two demonstrations for different shapelets of two kinds of heartbeat wavelets: normal heartbeat and Myocardial Infarction (Ischemia). We can find two wavelets that share a similar pattern which starts from a low value Q, then goes to a sharp peak P and returns back to a low value S. Differences between normal and Ischemia lies on that Ischemia has a longer time from S to T (another high point after S), and a lower value of R than a normal heartbear, which is called invalid amplitude.

Our DPSN model produces one prototype for each class. As shown in Fig. 5, the left prototype is a normal heartbeat signal that can quickly recover from S to T while the Ischemia signal has a relatively slow recovery curve in the heartbeat wavelet. Moreover, the discriminative shapelets found by DPSN, which is highlighted by red lines match the sub-sequence between R and T in the demonstrated example in Fig. 4. The recovery time from S to T is prolonged in the case of Ischemia on the right side. The value of R for Ischemia is lower than the normal heartbeat case.

B. Analysis of Pedestrian Patterns on Weekday and Weekend

The City of Melbourne, Australia has deployed a pedestrian counting system to automatically count pedestrian flow within the city and to gain deeper understanding of public resource usage in multiple city locations at different times of the day.

Data analysis can facilitate the decision maker to make better urban planning decisions for future city development in a human-centric way. As shown in the original features of the dataset, we can draw some interesting conclusions: 1) in an urban area, e.g. Chinatown, there are more pedestrians during the weekdays because many people work in urban areas and stay in theri home suburbs on the weekends. 2) pedestrians are more likely to have a short lunch-time during weekdays and spend more time on the lunch during weekends.

As demonstrated in Figure 6, the prototypes for each class generated from our model shows the above patterns. Moreover, two partial-length shapelets are extracted to identify the most discriminative characteristics. We found that there are many pedestrians at midnight on the weekends whilst there are hardly any people on the streets at midnight during the weekdays. This characteristic of overnight activities can be used as important features to classify the weekends and weekdays. The characteristics of populations, as mentioned above, are not the best discriminative feature. Because the population in a location is usually impacted by events that will eventually bring noise to this feature. In summary, the prototypes can demonstrate the most obvious patterns of the data, however, the discriminative shapelets can reveal the most important feature for classification.

VI. CONCLUSIONS

We propose the novel Dual Prototypical Shapelet Networks model for interpretable time-series classification on few-shot samples. In experiments, DPSN outperforms handcraft distance measurement, and the proposed model can be interpreted on dual granularity: representative sample with overview and discriminative shapelets with detail view. We use two case studies to demonstrate how the interpretability can facilitate the model explanation and derive new knowledge for data analysis.

REFERENCES

[1] G. F. Busatto, B. S. Diniz, and M. V. Zanetti, "Voxel-based morphometry in alzheimer's disease," *Expert review of neurotherapeutics*, vol. 8, no. 11, pp. 1691–1702, 2008.

[2] J. C.-Y. Sun and K. P.-C. Yeh, "The effects of attention monitoring with eeg biofeedback on university students' attention and self-efficacy: The case of anti-phishing instructional materials," *Computers & Education*, vol. 106, pp. 73–82, 2017.

[3] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *ICLR*, 2019.

[4] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, pp. 4077–4087, 2017.

[5] S. Motiian, Q. Jones, S. Iranmanesh, and G. Doretto, "Few-shot adversarial domain adaptation," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 6670–6680, Curran Associates, Inc., 2017.

[6] G. Long, L. Chen, X. Zhu, and C. Zhang, "Tcsst: transfer classification of short & sparse text using external data," in *Proceedings of the 21st ACM international conference on Information and knowledge management*, pp. 764–772, 2012.

[7] J. Jiang, S. Ji, and G. Long, "Decentralized knowledge acquisition for mobile internet applications," *World Wide Web*, pp. 1–17, 2020.

[8] H. Wahbeh, C. Calabrese, H. Zwickey, and D. Zajdel, "Binaural beat technology in humans: a pilot study to assess neuropsychologic, physiologic, and electroencephalographic effects," *The Journal of Alternative and Complementary Medicine*, vol. 13, no. 2, pp. 199–206, 2007.

[9] X. Gao, H. Cao, D. Ming, H. Qi, X. Wang, X. Wang, R. Chen, and P. Zhou, "Analysis of eeg activity in response to binaural beats with different frequencies," *International Journal of Psychophysiology*, vol. 94, no. 3, pp. 399–406, 2014.

[10] C. A. Ratanamahatana and E. Keogh, "Three myths about dynamic time warping data mining," in *Proceedings of the 2005 SIAM international conference on data mining*, pp. 506–510, SIAM, 2005.

[11] P. Schäfer, "The boss is concerned with time series classification in the presence of noise," *Data Mining and Knowledge Discovery*, vol. 29, no. 6, pp. 1505–1530, 2015.

[12] H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Yanping, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The ucr time series classification archive," October 2018. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

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

[14] W. Tang, G. Long, L. Liu, T. Zhou, J. Jiang, and M. Blumenstein, "Rethinking 1d-cnn for time series classification: A stronger baseline," *arXiv preprint arXiv:2002.10061*, 2020.

[15] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2006.

[16] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 3, pp. 453–465, 2013.

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[19] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135, JMLR. org, 2017.

[20] L. Liu, T. Zhou, G. Long, J. Jiang, L. Yao, and C. Zhang, "Prototype propagation networks (PPN) for weakly-supervised few-shot learning on category graph," in *International Joint Conferences on Artificial Intelligence (IJCAI)*, 2019.

[21] L. Liu, T. Zhou, G. Long, J. Jiang, and C. Zhang, "Learning to propagate for graph meta-learning," in *Neural Information Processing Systems (NeurIPS)*, 2019.

[22] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *International Conference on Machine Learning (ICML)*, 2016.

[23] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International Conference on Learning Representations (ICLR)*, 2017.

[24] E. Schwartz, L. Karlinsky, J. Shtok, S. Harary, M. Marder, S. Pankanti, R. Feris, A. Kumar, R. Giries, and A. M. Bronstein, "Repmet: Representative-based metric learning for classification and one-shot object detection," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[25] J. Gu, Y. Wang, Y. Chen, K. Cho, and V. O. Li, "Meta-learning for low-resource neural machine translation," in *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[26] S. X. Zhang, F. Tang, H. Dodge, J. Zhou, and F. Wang, "Metapred: Meta-learning for clinical risk prediction with limited patient electronic health records," in *ACM SIGKDD Conference ON Knowledge Discover and Data Mining (KDD)*, 2019.

[27] A. Krizhevsky, "Learning multiple layers of features from tiny images," tech. rep., Citeseer, 2009.

[28] A. Fisher, C. Rudin, and F. Dominici, "Model class reliance: Variable importance measures for any machine learning model class, from the" rashomon" perspective," *arXiv preprint arXiv:1801.01489*, 2018.

[29] B. Kim, R. Khanna, and O. O. Koyejo, "Examples are not enough, learn to criticize! criticism for interpretability," in *Advances in Neural Information Processing Systems*, pp. 2280–2288, 2016.

[30] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1885–1894, JMLR. org, 2017.

[31] G. Montavon, W. Samek, and K.-R. Müller, "Methods for interpreting and understanding deep neural networks," *Digital Signal Processing*, vol. 73, pp. 1–15, 2018.

[32] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PloS one*, vol. 10, no. 7, p. e0130140, 2015.

[33] N. Frosst and G. Hinton, "Distilling a neural network into a soft decision tree," *arXiv preprint arXiv:1711.09784*, 2017.

[34] M. Wu, M. C. Hughes, S. Parbhoo, M. Zazzi, V. Roth, and F. Doshi-Velez, "Beyond sparsity: Tree regularization of deep models for interpretability," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[35] Q. Zhang, J. Wu, H. Yang, Y. Tian, and C. Zhang, "Unsupervised feature learning from time series.," in *IJCAI*, pp. 2322–2328, 2016.

[36] P. Schäfer and U. Leser, "Fast and accurate time series classification with weasel," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pp. 637–646, ACM, 2017.

[37] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances," *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, 2017.

[38] P. Schäfer and M. Högqvist, "Sfa: a symbolic fourier approximation and index for similarity search in high dimensional datasets," in *Proceedings of the 15th International Conference on Extending Database Technology*, pp. 516–527, ACM, 2012.

[39] Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang, "Salient subsequence learning for time series clustering," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 9, pp. 2193–2207, 2018.

[40] R. T. Olszewski, "Generalized feature extraction for structural pattern recognition in time-series data," tech. rep., CARNEGIE-MELLON UNIV PITTSBURGH PA SCHOOL OF COMPUTER SCIENCE, 2001.

[41] J. Hills, J. Lines, E. Baranauskas, J. Mapp, and A. Bagnall, "Classification of time series by shapelet transformation," *Data Mining and Knowledge Discovery*, vol. 28, no. 4, pp. 851–881, 2014.

[42] K. R. Allen, E. Shelhamer, H. Shin, and J. B. Tenenbaum, "Infinite mixture prototypes for few-shot learning," *arXiv preprint arXiv:1902.04552*, 2019.

[43] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, U. Evci, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, *et al.*, "Meta-dataset: A dataset of datasets for learning to learn from few examples," *arXiv preprint arXiv:1903.03096*, 2019.