Salient Subsequence Learning for Time Series Clustering

Qin Zhang, Jia Wu, *Member, IEEE*, Peng Zhang, Guodong Long, and Chengqi Zhang, *Senior Member, IEEE*

Abstract—Time series has been a popular research topic over the past decade. Salient subsequences of time series that can benefit the learning task, e.g., classification or clustering, are called *shapelets*. Shapelet-based time series learning extracts these types of salient subsequences with highly informative features from a time series. Most existing methods for shapelet discovery must scan a large pool of candidate subsequences, which is a time-consuming process. A recent work, [1], uses regression learning to discover shapelets in a time series; however, it only considers learning shapelets from labeled time series data. This paper proposes an Unsupervised Salient Subsequence Learning (USSL) model that discovers shapelets without the effort of labeling. We developed this new learning function by integrating the strengths of *shapelet learning*, *shapelet regularization*, *spectral analysis* and *pseudo-label* to simultaneously and automatically learn shapelets to help clustering unlabeled time series better. The optimization model is iteratively solved via a coordinate descent algorithm. Experiments show that our USSL can learn meaningful shapelets, with promising results on real-world and synthetic data that surpass current state-of-the-art unsupervised time series learning methods.

Index Terms—Time series, shapelet, unsupervised feature learning, clustering

1 Introduction

Time series research in real-world applications is ubiquitous across areas such as finance [2], [3], medicine [4], [5], trajectory analysis [6], [7], and human action segmentation [8], [9]. Feature discovery from unsupervised time series is meaningful since it benefits not only the task of time series clustering but also finding latent interesting patterns from the raw time series sequences. For example, the work [10] introduces an example that using selected unsupervised features to cluster non-invasive fetal ECG signals to improve diagnostic systems. A cardiologist has confirmed that the feature obtained by the unsupervised method corresponds to the p-wave of the ECG signal, which describes the changing heart morphology of developing fetus.

- *Q. Zhang is with Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW 2007, Australia, and is also with the Data Centre of Social Network Group (SNG), Tencent, Shenzhen, Guangzhou 518057, China. E-mail: amberqzhang@tencent.com.*
- *J. Wu is with the Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, NSW 2109, Australia, and is also with Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW 2007, Australia. E-mail: jia.wu@mq.edu.au.*
- *P. Zhang is with the Ant Financial Service Group, Hangzhou 310012, China. E-mail: hanyi.zp@antfin.com.*
- *G. Long and C. Zhang are with Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, NSW 2007, Australia. E-mail: {guodong.long, chengqi.zhang}@uts.edu.au.*

Manuscript received 26 Oct. 2016; revised 1 Jan. 2018; accepted 27 May 2018. Date of publication 14 June 2018; date of current version 13 Aug. 2019. (Corresponding author: Jia Wu.) Recommended for acceptance by J. Ye. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TPAMI.2018.2847699

The main challenge of time series clustering is to find informative and salient subsequence features [11]. These features in the literature of time series analysis [10] are called *shapelets*. Shapelets disovery has become one of the research focuses in time series analysis, where a line of work [10], [12] has been proposed to either extract exact shapelets or learn approximate shapelets from time series.

A seminal work by Ye and Keogh [13] discovers the best shapelets by fully scanning all of a time series' segment candidates. All the shapelets are ranked according to a predefined distance metric, such as information gain, and the segments that best predict the class labels are selected as shapelets. However, this method suffers from high time complexity despite efforts to speed-up the algorithm [14], [15], [16]. Additionally, all these methods are yet to overcome the challenge of a large segment pool [1]. For example, the Synthetic Control dataset [17] contains 600 time series samples, each being 60 in length, which means the number of candidate segments for all lengths is 1.098×10^6! Such a massive number of candidates to choose from presents a massive challenge for shapelet selection approaches.

Recently, Grabocka et al. [1] proposed a brand new perspective on shapelet discovery in time series that uses regression learning to find shapelets rather than searching for shapelets in a candidate pool. This approach opened a new door for *learning shapelets from time series*. The beauty of this approach is that the learned shapelets are detached from and different to the original time series curves [18], which results in time efficiency. Moreover, the shapelets are robust to noise [19], as shown in Fig. 1.

However, this approach introduces labeling cost problems becouse it requires supervised learning. This motivated us to build a more economical shapelet learning

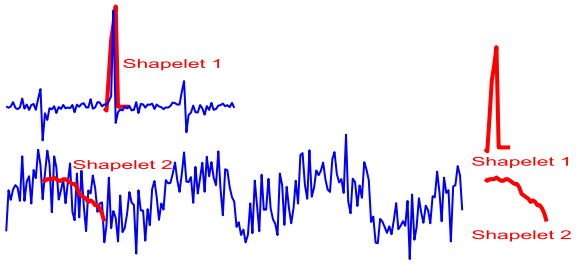


Fig. 1. The top left blue curve is a rectangular signal with Gaussian noise while the bottom left blue curve is a sinusoidal signal with Gaussian noise. The short curves (red in bold) on the right, are learned shapelets. We can see they are robust to noise and different with any original time series segment.

model that can automatically learn shapelets without the effort of human labeling. We call the learned unsupervised shapelet as *lu-shapelet*, with the aiming that it comtains distinguishing power and outstanding utility for unlabeled time series clustering.

We propose a novel **U**nsupervised **S**alient **S**ubsequence **L**earning (USSL) model for automatically learning shapelets within unlabeled time series data. First, a *pseudo-label* converts unsupervised learning into supervised learning. Then, a *regularized least-squares technique* and *spectral analysis* are used to learn the *lu*-shapelets, the pseudo-class labels, and the pseudo classification boundaries. Next, a shapelet regularization term is added to avoid learning similar *lu*-shapelets. Finally, a coordinate descent algorithm to simultaneously obtains the pseudo-class labels and learns the best *lu*-shapelets in an iterative manner.

Unlike existing unsupervised shapelet selection models [10], [12], which find shapelets via an exhaustive search of every candidate segment with respect to the original time series, our method aims to directly learns the optimal shapelets, *lu*-shapelets, that can best assist time series learning from unlabeled data. The learned shapelets will not be exactly as same as the original time series segments. This paper's main contributions are:

- USSL. Our model extends supervised shapelet learning models to handle unlabeled time series data by combining the strengths of pseudo-class labels, spectral analysis, regularized least-squares techniques and shapelet regularization.
- Empirical validation of USSL's efficiency and effectiveness though experiments on 36 real-world datasets and one synthetic dataset. Our tests demostrate promising performance, with superior results over other state-of-the-art unsupervised approaches.

The remainder of the paper includes: related works in Section 2; preliminaries in Section 3; the proposed USSL model in Section 4; the learning algorithm and related analysis in Section 5, experiments in Section 6 and finally the conclusions in Section 7.

2 Related Work

2.1 Non-Shapelet-Based Unsupervised Feature Learning

Researchers have also proposed a series of unsupervised methods for choosing informative features from unsupervised data [20], [21]. The traditional criterion in unsupervised feature selection requires choosing the features that best preserve the data similarity and manifold structure of the original feature space [22], [23], [24]. However, these methods cannot incorporate important information implied in the data and, therefore, cannot be directly applied into shapelet-learning problems. The existing unsupervised feature selection methods individually evaluated each feature's importance, and selected the best one by one [22], [23], neglecting the correlations between different features [24], [25].

The state-of-the-art methods for unsupervised feature seclection choose features by exploiting feature correlation and discriminative information simultaneously [26], [27], [28], [29]. Unsupervised discriminative feature selection [26] chooses the best informative features to represent data by considering their manifold structures.

A popular approach is to introduce pseudo-class labels into unsupervised feature selection to predict good cluster indicators [27], [30], [31]. Outliers and noise data are another important factors that significantly affect the final feature-selection performance [32]. Real-world data is normally not distributed in an ideal fashion: data outliers and noise are commonplace [33], [34]. Thus, it is important and necessary to make unsupervised feature selection robust to outliers and noise [32], [35].

2.2 Shapelet-Based Unsupervised Feature Learning

Shapelets also have been used to cluster time series [10], [36], [37]. Jesin et al. [10] proposed an algorhtm called "u-Shapelets", which uses unsupervised shapelets to cluster unlabeled time series. The method selects a set of unsupervised shapelets to separate the original dataset by searching and removing a subset. It repeats iteratively until no data remains, and uses a greedy search algorithm to try to maximize the gap between different groups divided by the unsupervised shapelets [38].

Paparrizos et al. [12] proposed a novel shape-based method, called k-shape, to cluster time series while keeping the domains independent and efficient [39]. K-shape uses an iteratively scalable refinement procedure that creates well-separated and homogeneous clusters. Specifically, it adopts a normalized cross-correlation measure to define the distance of two time series, whereas other methods [40] consider shapes and curves. This measure of distance remains invariant during shifting and scaling [41]. The time series assignments, as well as the cluster centroids are computed and updated in each iteration based on the normalized cross-correlation.

2.3 Shapelet-Based Supervised Feature Learning

2.3.1 Shapelets Selection

The works of [13], [42] introduced precise class label prediction by selecting the best short time series segments. The central concept leverages a given class label to score all the candidate shapelets (i.e., segments of the given training time series) for their predictability [43], [44]. The seminal work in the study of time series shapelets, [13], recursively searches for informative shapelets by building a decision-tree classifier featuring distances determined by information gain. Mood's median, F-Stat and Kruskall-Wallis are also used for shapelet selection [45], [46].

Time series datasets normally contain huge numbers of candidate subsequences that require brute force to select

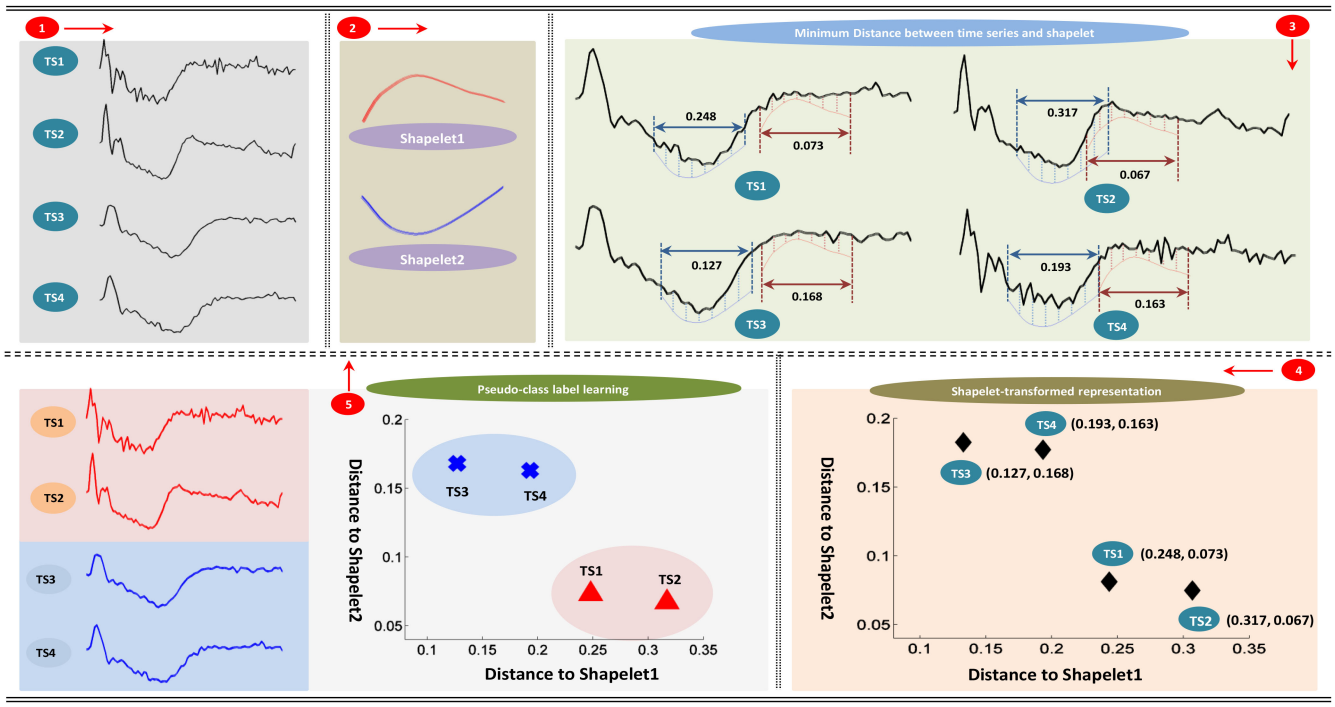


Fig. 2. The framework of the proposed Unsupervised Salient Subsequence Learning (USSL) model. From the original time series (①), we first learn shapelets using a shapelet similarity minimization principle (②); then, the distances of the learned shapelets and time series are calculated (③); the original data are mapped into the shapelet-based space (④). In shapelet-space, USSL learns the pseudo-class labels and a pseudo classifier using spectral analysis and regularized least-squares minimization (⑤); then, we update the shapelet with the new learned pseudo-class labels and the pseudo classifier (②); we then repeat this process until convergence. Finally, the USSL ensures the optimal shapelets and the pseudo-class labels.

shapelets, but they result in infeasible runtimes [1], [47]. Hence, many researchers have proposed various methods to increase their speed. Some are smart implementations that use entropy pruning on the information gain heuristic and abandon distance computations early [13]. Others prune the search space and re-use computations [14], [48]. Pruning of candidates by searching for potentially interesting candidates in a SAX representation [15] or by using infrequent shapelets [49], [50] is also seen.

2.3.2 Shapelet Learning

Instead of exhaustively searching shapelets, a recent work [1] proposed a new perspective on learning optimal shapelets. By representing shapelets as parameters that are learnable through regression techniques, rather than simply searching a pool of candidate segments in a set of training data, this new approach achieves statistically significant improvements over other shapelet-based classification methods. Regression learning methods are able to learn and obtain arbitrary shapelets; analysts are no longer limited to a restricted set of candidates.

Unlike these former research streams, our USSL method combines unsupervised feature selection with shapelet learning by introducing *unsupervised shapelet learning* to cluster unlabeled time series with auto-learned shapelets.

3 Preliminaries

In this paper, We represent scalars with the letters $(a, b, \ldots; \alpha, \beta, \ldots)$. Vectors are donated with lower-case bold letters $(\mathbf{a}, \mathbf{b}, \ldots)$, and matrices are represented with upper-case bold letters $(\mathbf{A}, \mathbf{B}, \ldots)$. The elements of a vector $\mathbf{a}$ are denoted as $\mathbf{a}_{(k)}$, where k is a positive integer and not greater than the length of the vector. $\mathbf{A}_{(ij)}$ is used to represent the element of matrix $\mathbf{A}$ that is located in jth column and ith row. $\mathbf{A}_{(:,j)}$ and $\mathbf{A}_{(i,:)}$ denote the jth column vector and ith row vector of the matrix respectively. We use $\mathbf{t}_{a,b}$ to represent the segment of an original time series that runs from time point a to point b. Sets are represented with upper-case squiggle letters $(\mathcal{A}, \mathcal{B}, \ldots)$.

Consider a time series dataset $\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n\}$. Each example $\mathbf{t}_i$ consists of a set of ordered real values, i.e., $(\mathbf{t}_{i(1)}, \mathbf{t}_{i(2)}, \ldots, \mathbf{t}_{i(q_i)})$, where q_i denotes the length of time series $\mathbf{t}_i$ and $1 \leq i \leq n$. With $q^{\mathcal{T}} = \max\{q_i, 1 \leq i \leq n\}$, we use $\mathbf{T} \in \mathbb{R}^{q^{\mathcal{T}} \times n}$ to represent the corresponding matrix of dataset $\mathcal{T}$. For time series which are shorter than $q^{\mathcal{T}}$, they are extended with zero vectors of appropriate lengths.

The top-k most salient and informative shapelet set $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k\}$ is what we wish to learn. Each shapelet $\mathbf{s}_i, 1 \leq i \leq k$, represents a vector of real values and the lengths of shapelets are hyper-parameters with user-specified integer values. Suppose $q^{\mathcal{S}}$ is the largest length for shapelets in $\mathcal{S}$, than matrix $\mathbf{S} \in \mathbb{R}^{k \times q^{\mathcal{S}}}$ is the corresponding matrix of dataset $\mathcal{S}$, with the short shapelets extended by zero vectors of appropriate lengths.

4 Unsupervised Shapelet Learning

This section presents the formulation of USSL model (see Eq. (7) Section 4.6). The model converts the original time series into a shapelet-based space first, then combines the shapelet diversity terms, spectral regularization, and regularized least square minimization are conbined together for final model. A conceptual illustration of the proposed USSL framework has been provided in Fig. 2.

In Section 4.1 we introduce the shapelet-transformed representation [1], which transfers the time series from its original space to the feature (i.e., shapelet) space. Sections 4.2 to 4.5 explain the pseudo-class label, spectral analysis, least-squares minimization, and shapelet similarity regularization, which are the essential components of the learning model.

4.1 Shapelet-Transformed Representation

Shapelet transformation was introduced by Lines et al. in [46] to reduce a lengthy time series to a much shorter vector in shapelet space. The shapelet-transformation of the time series orderliness preserves the shape information well.

Consider a time series set $\mathcal{T} = \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n\}$ and a set of shapelets $\mathcal{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k\}$. The shapelet-transformed matrix is represented as $\mathbf{X} \in \mathbb{R}^{k \times n}$, where each element $\mathbf{X}_{(\mathbf{s}_i, \mathbf{t}_j)}$, simplified as $\mathbf{X}_{(ij)}$, represents the distance between time series $\mathbf{t}_j$ and shapelet $\mathbf{s}_i$. $\mathbf{X}_{(ij)}$ can be calculated as

$$\mathbf{X}_{(ij)} = \min_{g=1,\ldots,\overline{q}} \frac{1}{l_i} \sum_{h=1}^{l_i} \left(\mathbf{t}_{j(g+h-1)} - \mathbf{s}_{i(h)}\right)^2, \tag{1}$$

where $\overline{q} = q_j - l_i + 1$ denotes the number of segments of l_i-length in $\mathbf{t}_j$; l_i represents the length of shapelet $\mathbf{s}_i$; q_j is the length of $\mathbf{t}_j$. $\mathbf{X}_{(ij)}$ is a function with respect to shapelet set $\mathcal{S}$, i.e., $\mathbf{X}(\mathbf{S})_{(ij)}$, by omitting the variable $\mathbf{S}$ for simplicity.

Since Eq. (1) is non-differential and the partial derivative $\frac{\partial X_{(ij)}}{\partial S_{i(h)}}$ is not defined, we approximate it with *soft minimum function* [1] in Eq. (2)

$$\mathbf{X}_{(ij)} \approx \frac{\sum_{q=1}^{\overline{q}} d_{ijq} \cdot e^{\alpha d_{ijq}}}{\sum_{q=1}^{\overline{q}} e^{\alpha d_{ijq}}}, \tag{2}$$

where $d_{ijq} = \frac{1}{l_i} \sum_{h=1}^{l_i} (\mathbf{t}_{j(q+h-1)} - \mathbf{s}_{i(h)})$. α is introduced to control precision. When $\alpha \to -\infty$, Eq. (2) approaches to Eq. (1). We set $\alpha = -100$ in our experiments following the research in [46].

4.2 Pseudo-Class Labels

Unlabeled training examples are the biggest challenge for unsupervised learning. Therefore, *pseudo-class labels* need to be introduced. Assume the time series dataset belongs to c categories, which means the pseudo-class matrix $\mathbf{Y} \in \mathbb{R}^{c \times n}$ contains c pseudo-labels. The probability for time series t_j belonging to ith category is denoted by $\mathbf{Y}_{(ij)}$. And, if for $\forall \tilde{i} \neq i$, there is $\mathbf{Y}_{(ij)} > \mathbf{Y}_{(\tilde{i},j)}$, then $\mathbf{t}_j$ belongs to cluster i.

4.3 Spectral Analysis

Spectral analysis has been widely adopted in unsupervised learning, like in clustering [51], which was first introduced in [52]. The guiding principle is that closely-related examples are likely to have same pseudo-class label [51], [53]. We use $\mathbf{G} \in \mathbb{R}^{n \times n}$ to denote the similarity matrix of the shapelet-transformed time series, i.e., matrix $\mathbf{X}$. Then we calculate $\mathbf{G}$ as

$$\mathbf{G}_{(ij)} = e^{-\frac{\|\mathbf{X}_{(:,i)} - \mathbf{X}_{(:,j)}\|^2}{\sigma^2}}. \tag{3}$$

where σ is the parameter for RBF kernel.

Based on $\mathbf{G}$, we expect similar data instances to share same pseudo-class labels. Guided by this, the spectral regularization term is formulated as follows:

$$\begin{aligned} &\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{G}_{(ij)} \|\mathbf{Y}_{(:,i)} - \mathbf{Y}_{(:,j)}\|_2^2, \\ &= \frac{1}{2} \sum_{k=1}^{c} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathbf{G}_{(ij)} \left(\mathbf{Y}_{(ki)} - \mathbf{Y}_{(kj)}\right)^2, \\ &= \sum_{k=1}^{c} \mathbf{Y}_{(k,:)} (\mathbf{D}_G - \mathbf{G}) \mathbf{Y}_{(k,:)}^{\top}, \\ &= tr(\mathbf{Y} \mathbf{L}_G \mathbf{Y}^{\top}). \end{aligned} \tag{4}$$

where $\mathbf{D}_G$ is a diagonal matrix and $\mathbf{D}_G(i,i) = \sum_{j=1}^{n} \mathbf{G}_{(ij)}$, $\mathbf{L}_G = \mathbf{D}_G - \mathbf{G}$ is the related Laplacian matrix.

4.4 Least-Squares Minimization

Let $\mathbf{W} \in \mathbb{R}^{k \times c}$ be the pseudo-classifier. We wish to achieve the least-squares error minimization with the following function:

$$\min_{\mathbf{W}} \ \|\mathbf{W}^T \mathbf{X} - \mathbf{Y}\|_F^2. \tag{5}$$

4.5 Shapelet Similarity Minimization

Given that we wish to learn shapelets with diverse shapes, the model is penalized if it outputs similar shapelets. Assume the shapelet similarity matrix as $\mathbf{H} \in \mathbb{R}^{k \times k}$, and the similarity between two shapelets $\mathbf{s}_i$ and $\mathbf{s}_j$ is represented by the element $\mathbf{H}_{(\mathbf{s}_i, \mathbf{s}_j)}$. We use $\mathbf{H}_{(ij)}$ to denote $\mathbf{H}_{(\mathbf{s}_i, \mathbf{s}_j)}$ for simplicity, and we have

$$\mathbf{H}_{(ij)} = e^{-\frac{\|d_{ij}\|^2}{\sigma^2}}, \tag{6}$$

where d_{ij} denotes the distance between shapelet $\mathbf{s}_i$ and $\mathbf{s}_j$, which can be calculated in a similar way to Eq. (2).

4.6 Unsupervised Salient Subsequence Learning

The formal unsupervised salient subsequence learning model is given in Eq. (7). It is a joint optimization problem that includes three variables: a pseudo classifier $\mathbf{W}$; the pseudo-class labels $\mathbf{Y}$; and the candidate shapelets $\mathbf{S}$.

$$\begin{aligned} \min_{\mathbf{W},\mathbf{S},\mathbf{Y}} \ &\frac{1}{2} tr(\mathbf{Y} \mathbf{L}_G(\mathbf{S}) \mathbf{Y}^{\top}) + \frac{\lambda_1}{2} \|\mathbf{H}(\mathbf{S})\|_F^2 \\ &+ \frac{\lambda_2}{2} \|\mathbf{W}^T \mathbf{X}(\mathbf{S}) - \mathbf{Y}\|_F^2 + \frac{\lambda_3}{2} \|\mathbf{W}\|_F^2. \end{aligned} \tag{7}$$

The first part of this optimization problem is a spectral regularization term to preserve the local structures between the time series. The second part diversifies the shapelets by minimizing the shapelet similarities. The third part includes the last two terms, which is the regularized least-squares minimization to learn the optimal pseudo-class labels and the pseudo classifier.

Note that the details of matrices $\mathbf{X}$, $\mathbf{L_G}$ and $\mathbf{H}$ are shown in Eqs. (2), (4), (6) respectively, all of which depend on the candidate shapelets $\mathbf{S}$. They are written explicitly as variables with respect to $\mathbf{S}$ in Eq. (7), i.e., $\mathbf{L_G}(\mathbf{S})$, $\mathbf{H}(\mathbf{S})$, and $\mathbf{X}(\mathbf{S})$. A coordinate descent algorithm is used to solve the proposed model, which is present next.

5 The Algorithm

Here, We intruduce the details of the coordinate descent algorithm, followed by an analysis of its convergence properties, initialization methods, and complexity.

5.1 Learning Algorithm

In line with the idea of a coordinate descent algorithm, some variables are updated iteratively, while the remaining variables are fixed. This process is repeated until convergence. The details are provided in Algorithm 1.

Algorithm 1. USSL

Require:
$\mathbf{T}$: unsupervised time series
k, r, l_{min}: parameters on shapelet number and length
i_{max}: maximum internal iterations
η: learning rate
$\lambda_1, \lambda_2, \lambda_3$: weight parameters
α, σ: similarity parameter and kernel parameter
Ensure:
$\mathbf{S}^*$: best shapelets
$\mathbf{Y}^* \in \mathbb{R}^{c\times n}$: optimal pseudo-class labels
Initialize: $\mathbf{S}_0, \mathbf{W}_0, \mathbf{Y}_0$
while Not convergent **do**
  Calculate: $\mathbf{X}_t(\mathbf{T}, \mathbf{S}_t|\alpha)$ based on Eq. (2)
    $\mathbf{L}_{Gt}(\mathbf{T}, \mathbf{S}_t|\alpha, \sigma)$ based on Eq. (4)
    $\mathbf{H}_t(\mathbf{S}_t|\alpha)$ based on Eq. (6)
  update $\mathbf{Y}_{t+1}$, $\mathbf{W}_{t+1}$**:**
    $\mathbf{Y}_{t+1} \leftarrow \lambda_2 \mathbf{W}_t^T \mathbf{X}_t (\mathbf{L}_{Gt} + \lambda_2 \mathbf{I})^{-1}$
    $\mathbf{W}_{t+1} \leftarrow (\lambda_2 \mathbf{X}_t \mathbf{X}_t^T + \lambda_3 \mathbf{I})^{-1} (\lambda_2 \mathbf{X}_t \mathbf{Y}_{t+1}^T)$.
  update $\mathbf{S}_{t+1}$**:**
    for $i = 1, \ldots, i_{max}$ **do**
      $\mathbf{S}_{i+1} \leftarrow \mathbf{S}_i - \eta \nabla \mathbf{S}_i$
      $\nabla \mathbf{S}_i = \frac{\partial \mathbf{F}(\mathbf{S}_i|\mathbf{X}_{t+1}, \mathbf{Y}_{t+1})}{\partial \mathbf{S}}$ is from Eq. (17)
    end for
    $\mathbf{S}_{t+1} = \mathbf{S}_{i_{max}+1}$
  $t \leftarrow t + 1$
end while
return $\mathbf{S}^* = \mathbf{S}_{t+1}$; $\mathbf{Y}^* = \mathbf{Y}_{t+1}$; $\mathbf{W}^* = \mathbf{W}_{t+1}$.

5.1.1 Updating $\mathbf{Y}$ by Fixing $\mathbf{W}$ and $\mathbf{S}$

By fixing $\mathbf{W}$ and $\mathbf{S}$, the function in Eq. (7) degenerates to Eq. (8),

$$\min_{\mathbf{Y}} \ \mathbf{F}(\mathbf{Y}) = \frac{1}{2} tr(\mathbf{Y}\mathbf{L}_G\mathbf{Y}^T) + \frac{\lambda_2}{2} \|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_F^2. \tag{8}$$

The derivative of the above function in terms of $\mathbf{Y}$ is,

$$\begin{aligned} \frac{\partial \mathbf{F}_\mathbf{Y}}{\partial \mathbf{Y}} &= \mathbf{Y}\mathbf{L}_G - \lambda_2(\mathbf{W}^T\mathbf{X} - \mathbf{Y}) \\ &= \mathbf{Y}(\mathbf{L}_G + \lambda_2 \mathbf{I}) - \lambda_2 \mathbf{W}^T\mathbf{X}. \end{aligned} \tag{9}$$

Let Eq. (9) equal 0, the solution for $\mathbf{Y}$ can be obtained from

$$\mathbf{Y} = \lambda_2 \mathbf{W}^T\mathbf{X}(\mathbf{L}_G + \lambda_2 \mathbf{I})^{-1}. \tag{10}$$

where $\mathbf{I}$ represents the identity matrix. And the final update strategy for $\mathbf{Y}$ is

$$\mathbf{Y}_{t+1} = \lambda_2 \mathbf{W}_t^\top \mathbf{X}_t (\mathbf{L}_{Gt} + \lambda_2 \mathbf{I})^{-1}. \tag{11}$$

5.1.2 Updating $\mathbf{W}$ by Fixing $\mathbf{S}$ and $\mathbf{Y}$

By fixing $\mathbf{S}$ and $\mathbf{Y}$, Eq. (7) degenerates to

$$\min_{\mathbf{W}} \ \mathbf{F}(\mathbf{W}) = \frac{\lambda_2}{2} \|\mathbf{W}^T\mathbf{X} - \mathbf{Y}\|_F^2 + \frac{\lambda_3}{2} \|\mathbf{W}\|_F^2. \tag{12}$$

The derivative of the above function in terms of $\mathbf{W}$ is,

$$\begin{aligned} \frac{\partial \mathbf{F}_\mathbf{W}}{\partial \mathbf{W}} &= \lambda_2 \mathbf{X}(\mathbf{W}^T\mathbf{X} - \mathbf{Y})^T + \lambda_3 \mathbf{W} \\ &= (\lambda_2 \mathbf{X}\mathbf{X}^T + \lambda_3 \mathbf{I})\mathbf{W} - \lambda_2 \mathbf{X}\mathbf{Y}^T. \end{aligned} \tag{13}$$

Let Eq. (13) equal 0, we obtain the solution for $\mathbf{W}$ as

$$\mathbf{W} = (\lambda_2 \mathbf{X}\mathbf{X}^T + \lambda_3 \mathbf{I})^{-1} (\lambda_2 \mathbf{X}\mathbf{Y}^T). \tag{14}$$

Thus, the update of $\mathbf{W}$ is

$$\mathbf{W}_{t+1} = (\lambda_2 \mathbf{X}_t \mathbf{X}_t^\top + \lambda_3 \mathbf{I})^{-1} (\lambda_2 \mathbf{X}_t \mathbf{Y}_{t+1}^T). \tag{15}$$

5.1.3 Updating $\mathbf{S}$ by Fixing $\mathbf{W}$ and $\mathbf{Y}$

With a fixed $\mathbf{W}$ and $\mathbf{Y}$, function in Eq. (7) degenerates to

$$\begin{aligned} \min_{\mathbf{S}} \ \mathbf{F}(\mathbf{S}) = &\frac{1}{2} tr(\mathbf{Y}\mathbf{L}_G(\mathbf{S})\mathbf{Y}^T) \\ &+ \frac{\lambda_1}{2} \|\mathbf{H}(\mathbf{S})\|_F^2 + \frac{\lambda_2}{2} \|\mathbf{W}^T\mathbf{X}(\mathbf{S}) - \mathbf{Y}\|_F^2. \end{aligned} \tag{16}$$

The above function is non-convex with the variable $\mathbf{S}$. Thus, we resort to an iterative algorithm to obtain the optimal solution for $\mathbf{S}$, i.e., $\mathbf{S}_{i+1} = \mathbf{S}_i - \eta \nabla \mathbf{S}_i$, where η denotes the learning rate and $\nabla \mathbf{S}_i = \frac{\partial \mathbf{F}(\mathbf{S}_i)}{\partial \mathbf{S}}$. Iteration guarantees the convergence. The derivative for Eq. (16) in terms of $\mathbf{S}_{(mp)}$ is

$$\begin{aligned} \frac{\partial \mathbf{F}(\mathbf{S})}{\partial \mathbf{S}_{(mp)}} = &\frac{1}{2} \mathbf{Y}^T\mathbf{Y} \frac{\partial \mathbf{L}_G(\mathbf{S})}{\partial \mathbf{S}_{(mp)}} \\ &+ \lambda_1 \mathbf{H}(\mathbf{S}) \frac{\partial \mathbf{H}(\mathbf{S})}{\partial \mathbf{S}_{(mp)}} \\ &+ \lambda_2 \mathbf{W}(\mathbf{W}^T X - Y) \frac{\partial \mathbf{X}(\mathbf{S})}{\partial \mathbf{S}_{(mp)}}, \end{aligned} \tag{17}$$

where $m = 1, \ldots, k$, and $p = 1, \ldots, l_m$.

Since $\mathbf{L}_G = \mathbf{D}_G - \mathbf{G}$ and $\mathbf{D}_G(i,i) = \sum_{j=1}^n \mathbf{G}_{(ij)}$, $\partial \mathbf{L}_G(\mathbf{S}) / \partial \mathbf{S}_{(mp)}$ turns to calculating $\partial \mathbf{G}_{(ij)} / \partial \mathbf{S}_{(mp)}$, i.e.,

$$\frac{\partial \mathbf{G}_{(ij)}}{\partial \mathbf{S}_{(mp)}} = -\frac{2\mathbf{G}_{(ij)}}{\sigma^2} \left(\sum_{q=1}^{k} (\mathbf{X}_{(qi)} - \mathbf{X}_{(qj)}) \right) \left(\frac{\partial \mathbf{X}_{(qi)}}{\partial \mathbf{S}_{(mp)}} - \frac{\partial \mathbf{X}_{(qj)}}{\partial \mathbf{S}_{(mp)}} \right), \tag{18}$$

and

$$\frac{\partial \mathbf{X}_{(ij)}}{\partial \mathbf{S}_{(mp)}} = \frac{1}{E_1^2} \sum_{q=1}^{\overline{q}_{ij}} e^{\alpha d_{ijq}} ((1 + \alpha d_{ijq}) E_1 - \alpha E_2) \frac{\partial d_{ijq}}{\partial \mathbf{S}_{(mp)}}, \tag{19}$$

where $E_1 = \sum_{q=1}^{\overline{q}_{ij}} e^{\alpha d_{ijq}}$, $E_2 = \sum_{q=1}^{\overline{q}_{ij}} d_{ijq} e^{\alpha d_{ijq}}$, $\overline{q}_{ij} = q_j - l_i + 1$ and $d_{ijq} = \frac{1}{l_i} \sum_{h=1}^{l_i} (\mathbf{t}_{j(q+h-1)} - \mathbf{s}_{i(h)})$.

$$\frac{\partial d_{ijq}}{\partial \mathbf{S}_{(mp)}} = \begin{cases} 0 & if \ \ i \neq m \\ \frac{2}{l_m} (\mathbf{S}_{(mp)} - T_{j,q+p-1}) & if \ \ i = m \end{cases}. \tag{20}$$

The second term in Eq. (17) turns to calculate Eq. (21),

$$\frac{\partial \mathbf{H}_{(ij)}}{\partial \mathbf{S}_{(mp)}} = -\frac{2}{\sigma^2} \tilde{d}_{ij} e^{-\frac{1}{\sigma^2} \tilde{d}_{ij}^2} \frac{\partial \tilde{d}_{ij}}{\partial \mathbf{S}_{(mp)}}, \tag{21}$$

where $\tilde{d}_{ij}$ is the distance of shapelets $\mathbf{s}_i$ and $\mathbf{s}_j$. The way to calculate $\tilde{d}_{ij}$ and $\frac{\partial \tilde{d}_{ij}}{\partial \mathbf{S}_{(mp)}}$ is the same as $\mathbf{X}_{(ij)}$ and $\frac{\partial \mathbf{X}_{(ij)}}{\partial \mathbf{S}_{(mp)}}$, respectively.

In summary, the gradient $\nabla \mathbf{S}_i = \frac{\partial \mathbf{F}(\mathbf{S})}{\partial \mathbf{S}_i}$ can be calculated using Eqs. (17), (18), (19), (20), and (21).

5.2 Convergence Analysis

This section provised an analysis and the proofs of the USSL algorithm's convergence based on the conclusion: Algorithm 1 convergents to its local optima under a well determined (normally small) learning rate η. Four convergence curve examples in Section 6.7 provide further support.

Lemma 1. *Consider the following problem*

$$\begin{aligned} &\min\ f(x) \\ &s.t.\ x \in \mathbb{X}, \end{aligned} \tag{22}$$

where $\mathbb{X}$ is a box (possibly unbounded) in $\mathbb{R}^n$, and $f(x)$ is Lipschitz continuously differentiable. If the set of optimal solutions for (22) is nonempty and there exists a scalar constant $\gamma > 0$ makes its gradient $\nabla f(x)$ satisfy

$$\|\nabla f(x) - \nabla f(y)\|^2 < \gamma \|f(x) - f(y)\|^2, \tag{23}$$

then the following equality always holds [54], [55]:

$$f(x) - f(y) \leq \nabla f(x)^T (x - y) - \frac{1}{2\gamma} \|\nabla f(x) - \nabla f(y)\|^2. \tag{24}$$

Theorem 1. *With the assumptions in Lemma 1, the gradient descent algorithm for a convex function $f(x)$ with the update rule $x_{t+1} = x_t - \lambda \nabla f(x_t)$ will converge to its optimal solution x^* as long as learning rate λ is small enough that $0 < \lambda < \frac{1}{\gamma}$.*

Proof. We only need to prove that x_{t+1} is closer to x^* than x_t since $f(x)$ is convex. We have

$$\begin{aligned} \|x_{t+1} - x^*\|^2 &= \|x_t - \lambda \nabla f(x_t) - x^*\|^2 \\ &= \|x_t - x^*\|^2 - 2\lambda \nabla f(x)^T (x_t - x^*) \\ &\quad + \lambda^2 \|\nabla f(x_t)\|^2. \end{aligned} \tag{25}$$

With Lemma 1, we have

$$\begin{aligned} f(x_t) - f(x^*) \leq \nabla f(x_t)^T (x_t - x^*) \\ - \frac{1}{2\gamma} \|\nabla f(x_t) - \nabla f(x^*)\|^2. \end{aligned} \tag{26}$$

Further, we have $\nabla f(x^*) = 0$ and $f(x_t) \geq f(x^*)$ since x^* is the optimal solution. Combined with Eq. (26), we obtain

$$-\nabla f(x_t)^T (x_t - x^*) \leq -\frac{1}{2\gamma} \|\nabla f(x_t)\|^2. \tag{27}$$

Combine Eqs. (25) and (27), we obtain

$$\begin{aligned} \|x_{t+1} - x^*\|^2 &\leq \|x_t - x^*\|^2 - \frac{\lambda}{\gamma} \|\nabla f(x_t)\|^2 \\ &\quad + \lambda^2 \|\nabla f(x_t)\|^2 \\ &= \|x_t - x^*\|^2 - \lambda (\frac{1}{\gamma} - \lambda) \|\nabla f(x_t)\|^2. \end{aligned} \tag{28}$$

Given the learning rate of $0 < \lambda < \frac{1}{\gamma}$, we have $\|x_{t+1} - x^*\|^2 < \|x_t - x^*\|^2$. Therefore, Theorem 1 is proven. □

Deduction 1. *For function*

$$f(x) = \sum_i^N a_i (w_i x + b_i)^2, x \in \mathbb{X}, N > 1, \tag{29}$$

a gradient descent algorithm with update rule $x_{t+1} = x_t - \lambda \nabla f(x_t)$ will converge to its optimal solution x^*, with the learning rate $\lambda < \frac{\tau}{\sum_i^N 4 a_i^2 w_i^4}$, where $\tau > 0$ is the lower bound for $\sum_i^N a_i^2 w_i^2 \|w_i(x + y) + 2b_i\|^2, \forall x, y \in \mathbb{X}, N > 1$.

Proof. From the left side, we have

$$\begin{aligned} \|\nabla f(x) - \nabla f(y)\|^2 &= \sum_i^N \|2a_i w_i (w_i x + b_i) \\ &\quad - 2a_i w_i (w_i y + b_i)\|^2 \\ &= \sum_i^N 4 a_i^2 w_i^4 \|x - y\|^2. \end{aligned} \tag{30}$$

From the right side, we have

$$f(x) - f(y) = \sum_i^N a_i [w_i^2 (\|x\|^2 - \|y\|^2) + 2w_i b_i (x - y)], \tag{31}$$

$$\|f(x) - f(y)\|^2 = \sum_i^N a_i^2 w_i^2 \|w_i(x + y) + 2b_i\|^2 \|x - y\|^2. \tag{32}$$

There exists a lower bound $\tau > 0$ for $\sum_i^N a_i^2 w_i^2 \|w_i(x + y) + 2b_i\|^2$ with $\forall x, y \in \mathbb{X}$, since $N > 1$. To the convex function $f(x)$, compare Eq. (30) with Eq. (32), according to Theorem 1, Deduction 1 is proven. □

Theorem 2. *USSL, Algorithm 1, converges to its local optima.*

Proof. In each iteration, for updates Y and W, the degenerated problems (8) and (12) have the same form as (29). Based on Deduction 1, Eq. (33) always holds

$$F(Y_{t+1}, W_{t+1}, S_t) \leq F(Y_{t+1}, W_t, S_t) \leq F(Y_t, W_t, S_t), \tag{33}$$

where $F(Y, W, S)$ is the objective function in Eq. (7), and S_t is fixed in this part. When updating S, while Y and W are fixed as Y_{t+1} and W_{t+1}, for each single variable $S_{(mp)}$, $m = 1, \ldots, k, p = 1, \ldots, l_m$, following the idea of gradient descent algorithm, with a small enough learning rate η, we have Eq. (34) holds according to Theorem 1,

$$\begin{aligned} F(S_{t+1}, Y_{t+1}, W_{t+1}) &= F(S_{i_{max}+1}, Y_{t+1}, W_{t+1}) \\ &\leq F(S_{i_{max}}, Y_{t+1}, W_{t+1}) \\ &\ldots \\ &\leq F(S_t, Y_{t+1}, W_{t+1}). \end{aligned} \tag{34}$$

This means Algorithm 1 guarantees a constant decrease in the objective function. Additionally, because the objective function's lower limit is zero, the theorem is proven. □

5.3 Initialization

Since Algorithm 1 only converges to its local optima, here we discuss initialization skills. $\mathbf{S}_0$, $\mathbf{Y}_0$ and $\mathbf{W}_0$ are expected to be initialized. Specifically, we initialize $\mathbf{S}_0$ with centroids of the same-length segments obtained from a raw time series, given that centroids typically represent the original

TABLE 1
Statistics of the Benchmark Time Series Datasets

No.	Dataset	# Train\Test	Length	classes	No.	Dataset	# Train\Test	Length	classes
1	ArrowHead	36\175	251	3	19	MiddlePhalanxOutlineCorrect	600\291	80	2
2	Beef	30\30	470	5	20	MiddlePhalanxTW	399\154	80	6
3	BeetleFly	20\20	512	2	21	MoteStrain	20\1252	84	2
4	BirdChicken	20\20	512	2	22	OSULeaf	200\242	427	6
5	Car	60\60	577	4	23	Plane	105\105	144	7
6	ChlorineConcentration	467\3840	166	3	24	ProximalPhalanxOutlineAgeGroup	400\205	80	3
7	Coffee	28\28	286	2	25	ProximalPhalanxTW	400\205	80	6
8	DiatomSizeReduction	16\306	345	4	26	SonyAIBORobotSurface1	20\601	70	2
9	DistalPhalanxOutlineAgeGroup	400\139	80	3	27	SonyAIBORobotSurface2	27\953	65	2
10	DistalPhalanxOutlineCorrect	600\276	80	2	28	SwedishLeaf	500\625	128	15
11	ECG200	100\100	96	2	29	Symbols	25\995	398	6
12	ECGFiveDays	23\861	136	2	30	ToeSegmentation1	40\228	277	2
13	GunPoint	50\150	150	2	31	ToeSegmentation2	36\130	343	2
14	Ham	109\105	431	2	32	TwoLeadECG	23\1139	82	2
15	Herring	64\64	512	2	33	TwoPatterns	1000\4000	128	4
16	Lightning2	60\61	637	2	34	Wafer	1000\6164	152	2
17	Meat	60\60	448	3	35	Wine	57\54	234	2
18	MiddlePhalanxOutlineAgeGroup	400\154	80	3	36	WordsSynonyms	267\638	270	25

data's main patterns. Based on $\mathbf{S}_0$, the original time series are transferred to the shapelet-based space as matrix $\mathbf{X}(\mathbf{S_0})$. In the shapelet-based space, $\mathbf{Y}_0$ is initialize with the cluster labels, and $\mathbf{W}_0$ is initialized with the centers of the clusters obtained by k-means. This initialization enables fast convergence.

5.4 Complexity Analysis

This section analyzes the time complexity of the USSL algorithm.

As stated in Algorithm 1, USSL initializes at a constant pace (line 1); then, it is solved iteratively. When matrix $\mathbf{X}_t$ is calculated in each iteration (line 3), suppose the longest time series in T has q elements, and the maximum length of shapelets is l. It takes $O(qnkl)$, where k denotes the number of learned shapelets, n denotes the size of T. Similarly, when calculating the matrix $\mathbf{L}_{G_t}$ (line 4) and $\mathbf{H}_t$ (line 5), it takes $O(kn^2)$ and $O(l^2k^2)$, respectively. When updating the pseudo-class label matrix $\mathbf{Y}_{t+1}$ and the pseudo classifier $\mathbf{W}_{t+1}$ (line 6-8), it takes $O(ckn + cn^2 + n^3)$ and $O(nk^2 + k^3 + nkc + k^2c)$. c is the number of clusters. When updating S_{t+1} (line 9-14), it takes $O(i_{max}(q^2n^2 + n^2c + k^2l^2 + k^3 + ckn))$, where i_{max} is the largest iteration of the internal process. In summary, the total complexity is $O(I(qnlk + kn^2 + lk^2 + cnk^2 + n^3 + kn + cn^2 + + k^3 + k^2c + i_{max}(q^2n^2 + n^2c + k^2l^2 + k^3 + ckn))$ where I is supposed to be the maximum number of iterations of the whole algorithm. Since $l \ll q$ and $k, c \ll n$, USSL algorithm's time complexity is $O(I(n^3 + q^2n^2i_{max}))$.

6 Experiments

We validated USSL's performance with a variety of experiments conducted on a Windows 7 machine with 8 GB memory and 3.00 GHz CPU. The source codes are available at https://github.com/BlindReview/shapelet.

6.1 Datasets

To validate the proposed algorithm's performance, we adopted 36 time series benchmark datasets from the UCR archive [17], [56]. Each dataset contained between 40 and 7,164 sequences. Within-dataset sequences were of equal length; between-dataset sequence length varied from 80 to 637. These datasets were annotated, and every sequence belonged to only one class. One training set and one test set were conluded in each raw dataset. We used training sets to tune the parameters and the test sets for evaluation. The details are shown in Table 1.

6.2 Measures

The Rand Index (RI) [57], normalized mutual information (NMI) [58], the Jaccard score [59], and the Folkes and Mallow index [10], [60] are the usual measures for evaluating clustering performance. The most popular are RI and NMI, with the others considered variants. As such, we used the *RI* and *NMI* as our evaluation methods.

RI can be calculated as

$$Rand\ index = \frac{TP + TN}{TP + TN + FP + FN}, \tag{35}$$

where TP denotes the number of time series pairs that belong to the same cluster in the pseudo-class matrix $\mathbf{Y}^*$, and the same class in genuine class labels $\mathbf{L}_{true}$. Similarly, TN represents the number of pairs that belong to different clusters in $\mathbf{Y}^*$ and in $\mathbf{L}_{true}$; FP represents the number of pairs that are assigned the same label in $\mathbf{Y}^*$ but belong to different classes in $\mathbf{L}_{true}$. FN denotes the number of pairs that have diverse labels in $\mathbf{Y}^*$ but belong to the same class in $\mathbf{L}_{true}$.

NMI is calculated as

$$NMI = \frac{\sum_{i=1}^{M}\sum_{j=1}^{M} N_{ij} log(\frac{N \cdot N_{ij}}{|G_i||A_j|})}{\sqrt{(\sum_{i=1}^{M} |G_i| log\frac{|G_i|}{N})(\sum_{j=1}^{M} |A_j| log\frac{A_j}{N})}}, \tag{36}$$

where N represents the total number of time series. $|G_i|$, $|A_j|$ are the numbers of time series in cluster G_i and A_j. $N_{ij} = |G_i \cap A_j|$ denotes the number of time series belonging to the intersection of sets G_i and A_j.

In these two measures, values close to 1 indicates high quality clustering [10], [12].

TABLE 2
Rand Index (RI) Comparisons on 36 Time Series Datasets

Dataset	k-means	UDFS	NDFS	RUFS	RSFS	KSC	KDBA	k-shape	u-shapelte	USSL
ArrowHead	0.6905	0.7254	0.7381	**0.7476**	0.7108	0.7254	0.7222	0.7254	0.6460	0.7159
Beef	0.6713	0.6759	0.7034	**0.7149**	0.6975	0.7057	0.6713	0.5402	0.6966	0.6966
BeetleFly	0.4789	0.4947	0.5579	0.6053	0.6516	0.6053	0.6053	0.6053	0.7316	**0.8105**
BirdChicken	0.4947	0.4947	0.7316	0.5579	0.6632	0.7316	0.6053	0.6632	0.5579	**0.8105**
Car	0.6345	0.6757	0.6260	0.6667	0.6708	0.6898	0.6254	0.7028	0.6418	**0.7345**
ChlorineConcentration	0.5241	0.5282	0.5225	**0.5330**	0.5316	0.5256	0.5300	0.4111	0.5318	0.4997
Coffee	0.7460	0.8624	**1.0000**	0.5476	**1.0000**	**1.0000**	0.4841	**1.0000**	**1.0000**	**1.0000**
DiatomSizeReduction	0.9583	0.9583	0.9583	0.9333	0.9167	**1.0000**	0.9583	**1.0000**	0.7083	**1.0000**
Dist.Phal.Outl.AgeGroup	0.6171	0.6531	0.6239	0.6252	0.6539	0.6535	**0.6750**	0.6020	0.6273	0.6650
Dist.Phal.Outl.Correct	0.5252	0.5362	0.5383	0.5252	0.5327	0.5235	0.5203	0.5252	0.5098	**0.5962**
ECG200	0.6315	0.6533	0.6315	0.7018	0.6916	0.6315	0.6018	0.7018	0.5758	**0.7285**
ECGFiveDays	0.4783	0.5020	0.5573	0.5020	0.5953	0.5257	0.5573	0.5020	0.5968	**0.8340**
GunPoint	0.4971	0.5029	0.5102	0.6498	0.4994	0.4971	0.5420	0.6278	0.6278	**0.7257**
Ham	0.5025	0.5219	0.5362	0.5107	0.5127	0.5362	0.5141	0.5311	0.5362	**0.6393**
Herring	0.4965	0.5099	0.5164	0.5238	0.5151	0.4940	0.5164	0.4965	0.5417	**0.6190**
Lighting2	0.4966	0.5119	0.5373	0.5729	0.5269	0.6263	0.5119	0.6548	0.5192	**0.6955**
Meat	0.6595	0.6483	0.6635	0.6578	0.6657	0.6723	0.6816	0.6575	0.6742	**0.7740**
Mid.Phal.Outl.AgeGroup	0.5351	0.5269	0.5350	0.5315	0.5473	0.5364	0.5513	0.5105	0.5396	**0.5807**
Mid.Phal.Outl.Correct	0.5000	0.5431	0.5047	0.5114	0.5149	0.5014	0.5563	0.5114	0.5218	**0.6635**
Mid.Phal.TW	0.0983	0.1225	0.1919	0.7920	0.8062	**0.8187**	0.8046	0.6213	0.7920	0.7920
MoteStrain	0.4947	0.5579	0.6053	0.5579	0.6168	0.6632	0.4789	0.6053	0.4789	**0.8105**
OSULeaf	0.5615	0.5372	0.5622	0.5497	0.5665	0.5714	0.5541	0.5538	0.5525	**0.6551**
Plane	0.9081	0.8949	0.8954	0.9220	0.9314	0.9603	0.9225	0.9901	**1.0000**	**1.0000**
Prox.Phal.Outl.AgeGroup	0.5288	0.4997	0.5463	0.5780	0.5384	0.5305	0.5192	0.5617	0.5206	**0.7939**
Prox.Phal.TW	0.4789	0.4947	0.6053	0.5579	0.5211	0.6053	0.5211	0.5211	0.4789	**0.7282**
SonyAIBORobotSurface	0.7721	0.7695	0.7721	0.7787	0.7928	0.7726	0.7988	0.8084	0.7639	**0.8105**
SonyAIBORobotSurfaceII	0.8697	0.8745	0.8865	0.8756	0.8948	**0.9039**	0.8684	0.5617	0.8770	0.8575
SwedishLeaf	0.4987	0.4923	0.5500	0.5192	0.5038	0.4923	0.5500	0.5333	0.6154	**0.8547**
Symbols	0.8810	0.8548	0.8562	0.8525	0.9060	0.8982	**0.9774**	0.8373	0.9603	0.9200
ToeSegmentation1	0.4873	0.4921	0.5873	0.5429	0.4968	0.5000	0.6143	0.6143	0.5873	**0.6718**
ToeSegmentation2	0.5257	0.5257	0.5968	0.5968	0.5826	0.5257	0.5573	0.5257	0.5020	**0.6778**
TwoPatterns	0.8529	0.8259	0.8530	0.8385	**0.8588**	0.8585	0.8446	0.8046	0.7757	0.8318
TwoLeadECG	0.5476	0.5495	0.6328	0.8246	0.5635	0.5464	0.5476	0.8246	0.5404	**0.8628**
wafer	0.4925	0.4925	0.5263	0.5263	0.4925	0.4925	0.4925	0.4925	0.4925	**0.8246**
Wine	0.4984	0.4987	0.5123	0.5021	0.5033	0.5006	0.5064	0.5001	0.5033	**0.8985**
WordsSynonyms	0.8775	0.8697	0.8760	**0.8861**	0.8817	0.8727	0.8159	0.7844	0.8230	0.8540
Average (improvement)	0.5975 (28.47%)	0.6077 (26.31%)	0.6402 (19.90%)	0.6478 (18.49%)	0.6543 (17.32%)	0.6582 (16.62%)	0.6334 (21.19%)	0.6419 (19.58%)	0.6402 (19.90%)	**0.7676**

6.3 Baseline Methods and Parameter Setting

We compared USSL with nine representative unsupervised feature selection and feature learning methods for time series. They consisted of three algorithm types: 1) original time series-based unsupervised feature selection methods, including k-means, UDFS, NDFS, RUFS, and RSFS; 2) distance-based unsupervised time series clustering methods, including KSC, k-DBA, and k-shape; and 3) shapelet-based unsupervised feature learning methods, such as the u-shapelet method. The details are as follows:

- k-means: Uses k-means on the entire time series.
- UDFS [26]: Unsupervised discriminative feature selection that simultaneously explores manifold structure, local discriminative information and feature correlations.
- NDFS [27]: Nonnegative discriminative feature selection that adopts $l_{2,1}$ regularised regression and nonnegative spectral analysis as a joint framework for selecting features.
- RUFS [35]: Robust unsupervised feature selection that uses robust orthogonal nonnegative matrix factorization to perform feature learning jointly.
- RSFS [28]: Robust spectral learning for unsupervised feature selection, which joints spectral regression with sparse graph embedding.
- KSC [61]: Uses k-means for clustering by adopting a pairwise scaling distance measure and computing the spectral norm of a matrix for centroid computation.
- k-DBA [62]: Adopts k-means and dynamic time warping distance to obtain centroids, via a DBA method.
- k-shape [12]: Adopts a scalable iterative refinement procedure to explore the shapes of time series that have a normalized cross-correlation measure.
- u-shapelet [10]: a time series clustering method that deliberately ignores the rest of the data and only uses local patterns to cluster the time series.

We tuned the parameters of these benchmark methods according to the descriptions in their original papers. To establish the best parameters for USSL, we tuned a series of hyper-parameters.

For the length of a shapelet, folowing the settings of the work in [1], we first set it as a minimum value, donated as l_{min}. It is then expanded to different lengths with the

TABLE 3
Normalized Mutual Information (NMI) Comparisons on 36 Time Series Datasets

Dataset	k-means	UDFS	NDFS	RUFS	RSFS	KSC	KDBA	k-shape	u-shapelte	USSL
ArrowHead	0.4816	0.5240	0.4997	0.5975	0.5104	0.5240	0.4816	0.5240	0.3522	**0.6322**
Beef	0.2925	0.2718	0.3647	0.3799	0.3597	**0.3828**	0.3340	0.3338	0.3413	0.3338
BeetleFly	0.0073	0.0371	0.1264	0.1919	0.2795	0.2215	0.2783	0.3456	0.5105	**0.5310**
BirdChicken	0.0371	0.0371	0.3988	0.1187	0.3002	0.3988	0.2167	0.3456	0.2783	**0.6190**
Car	0.2540	0.2319	0.2361	0.2511	0.2920	0.2719	0.2691	0.3771	0.3655	**0.4650**
ChlorineConcentration	0.0129	0.0138	0.0075	**0.0254**	0.0159	0.0147	0.0164	0.0000	0.0135	0.0133
Coffee	0.5246	0.6945	**1.0000**	0.2513	**1.0000**	**1.0000**	0.0778	**1.0000**	**1.0000**	**1.0000**
DiatomSizeReduction	0.9300	0.9300	0.9300	0.8734	0.8761	**1.0000**	0.9300	**1.0000**	0.4849	**1.0000**
Dist.Phal.Outl.AgeGroup	0.1880	0.3262	0.1943	0.2762	0.3548	0.3331	**0.4261**	0.2911	0.2577	0.3846
Dist.Phal.Outl.Correct	0.0278	0.0473	0.0567	0.1071	0.0782	0.0261	0.0199	0.0527	0.0063	**0.1626**
ECG200	0.1403	0.1854	0.1403	0.2668	0.2918	0.1403	0.1886	0.3682	0.1323	**0.3776**
ECGFiveDays	0.0002	0.0600	0.1296	0.0352	0.1760	0.0682	0.1983	0.0002	0.1498	**0.6502**
GunPoint	0.0126	0.0220	0.0334	0.2405	0.0152	0.0126	0.1288	0.3653	0.3653	**0.4878**
Ham	0.0093	0.0389	0.0595	0.0980	0.0256	0.0595	0.0265	0.0517	0.0619	**0.3411**
Herring	0.0013	0.0253	0.0225	0.0518	0.0236	0.0027	0.0000	0.0027	0.1324	**0.1718**
Lighting2	0.0038	0.0047	0.0851	0.1426	0.0326	0.1979	0.0850	0.2670	0.0144	**0.3727**
Meat	0.2510	0.2832	0.2416	0.1943	0.3016	0.2846	0.3661	0.2254	0.2716	**0.9085**
Mid.Phal.Outl.AgeGroup	0.0219	0.1105	0.0416	0.1595	0.0968	0.1061	0.1148	0.0722	0.1491	**0.2780**
Mid.Phal.Outl.Correct	0.0024	0.0713	0.0150	0.0443	0.0321	0.0053	0.0760	0.0349	0.0253	**0.2503**
MiddlePhalanxTW	0.4134	0.4276	0.4149	0.5366	0.4217	0.4486	0.4497	0.5229	0.4065	**0.9202**
MoteStrain	0.0551	0.1187	0.1919	0.1264	0.2373	0.3002	0.0970	0.2215	0.0082	**0.5310**
OSULeaf	0.0208	0.0200	0.0352	0.0246	0.0463	0.0421	0.0327	0.0126	0.0203	**0.3353**
Plane	0.8598	0.8046	0.8414	0.8675	0.8736	0.9218	0.8784	0.9642	**1.0000**	**1.0000**
Prox.Phal.Outl.AgeGroup	0.0635	0.0182	0.0830	0.0726	0.0938	0.0682	0.0377	0.0110	0.0332	**0.6813**
Prox.Phal.TW	0.0082	0.0308	0.2215	0.1187	0.0809	0.1919	0.2167	0.1577	0.0107	**1.0000**
SonyAIBORobotSurface	0.6112	0.6122	0.6112	0.6278	0.6368	0.6129	0.5516	**0.7107**	0.5803	0.5597
SonyAIBORobotSurfaceII	0.5444	0.4802	0.5413	0.5107	0.5406	0.5619	0.5481	0.0110	0.5903	**0.6858**
SwedishLeaf	0.0168	0.0082	0.0934	0.0457	0.0269	0.0073	0.1277	0.1041	0.3456	**0.9186**
Symbols	0.7780	0.7277	0.7593	0.7174	0.8027	0.8264	**0.9388**	0.6366	0.8691	0.8821
ToeSegmentation1	0.0022	0.0089	0.2141	0.0880	0.0174	0.0202	0.2712	0.3073	0.3073	**0.3351**
ToeSegmentation2	0.0863	0.0727	0.1713	0.1713	0.1625	0.0863	0.2627	0.0863	0.1519	**0.4308**
TwoPatterns	0.4696	0.3393	0.4351	0.4678	0.4608	0.4705	0.4419	0.3949	0.2979	**0.4911**
TwoLeadECG	0.0000	0.0004	0.1353	0.1238	0.0829	0.0011	0.0103	0.0000	0.0529	**0.5471**
wafer	0.0010	0.0010	0.0546	**0.0746**	0.0194	0.0010	0.0000	0.0010	0.0010	0.0492
Wine	0.0031	0.0045	0.0259	0.0065	0.0096	0.0094	0.0211	0.0119	0.0171	**0.7511**
WordsSynonyms	0.5435	0.4745	0.5396	0.5623	**0.5462**	0.4874	0.4527	0.4154	0.3933	0.4984
Average (Improvement)	0.2132 (155.30%)	0.2240 (142.99%)	0.2764 (96.92%)	0.2624 (107.43%)	0.2812 (93.56%)	0.2808 (93.84%)	0.2659 (104.70%)	0.2841 (91.59%)	0.2777 (96.00%)	**0.5443**

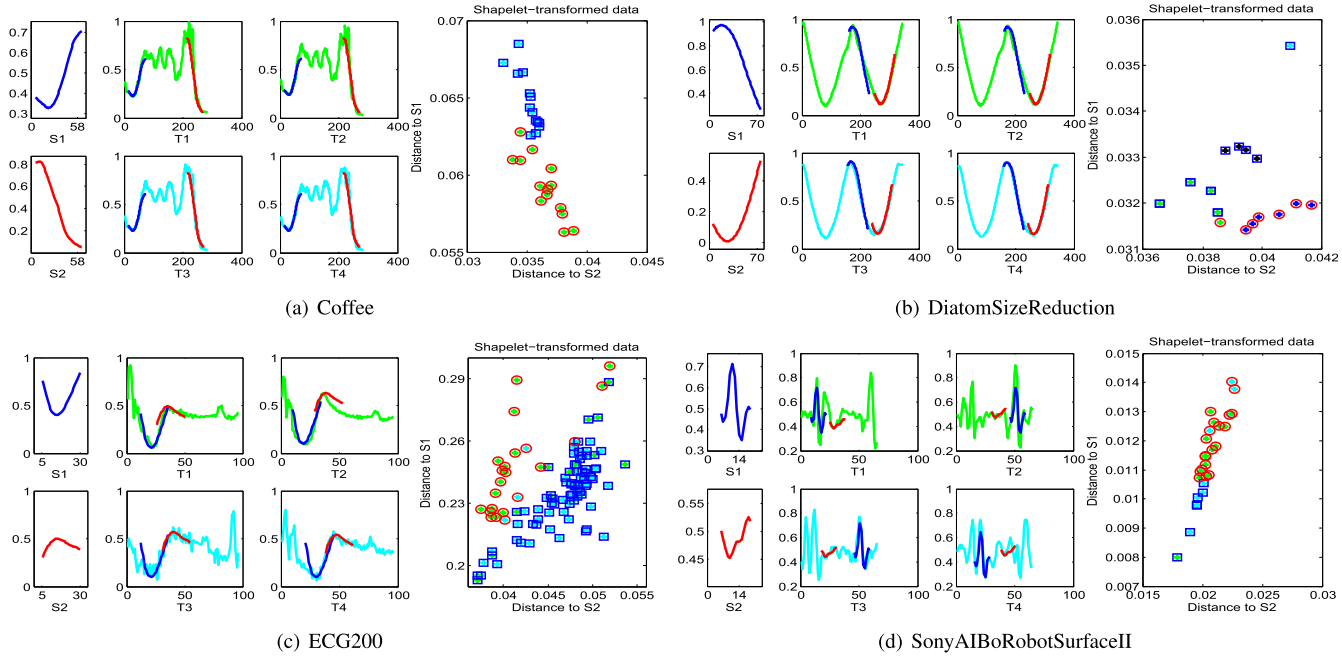
parameter r, i.e., $\{l_{min}, 2 \times l_{min}, \ldots, r \times l_{min}\}$. Each length $i \times l_{min}$ contains k_i shapelets and $k = \sum_{i=1}^{r} k_i$. Obviously, $\mathcal{S} \in \bigcup_{i=1}^{r} \mathbb{R}^{k_i \times (i \times l_{min})}$, where $r \times l_{min} \ll q_i$ keeps the shapelets compact.

We used a grid search approach with the number of lu-shapelets $k\ was searched from\ \{1, 2, \ldots, 5\}$, and the minimum length of lu-shapelets l_{min} is searched in $\{0.05, 0.1, \ldots, 0.25\}$, where $l_{min} = 0.05$ means 5 percent of the original time series length. Parameter r was searched from $\{1, 2, 3\}$. The weight parameters $\lambda_1, \lambda_2, \lambda_3$ were searched from $\{10^{-8}, 10^{-6}, \ldots, 10^{8}\}$. The learning rate was kept fixed at $\eta = 0.01$, and the loop was set to be no more than 50 iterations, which has been experimentally proven to be sufficiently large. To better evaluate performance, we set number of clusters to be the same as the genuine classes of the original datasets. Additionally, we set $\alpha = -100$ by following [1], and set $\sigma = 1$ experimentally.

6.4 Comparison

We evaluated USSL against the selected baselines using RI and NMI measures. Tables 2 and 3 summarise the experimental results on the 36 datasets. Best results are highlighted in bold.

The results in both showed that the USSL approach was effective, and performance was significantly better than the other baselines. USSL performed best with respect to RI on 27 datasets and to NMI on 29 datasets of the total 36. The average improvement in terms of RI compared to the nine baselines was 20.85 percent (greatest 28.47 percent, least 16.62 percent), while the average improvement in terms of NMI was 109.15 percent (greatest 155.30 percent, least 91.59 percent). This verifies the proposed USSL's ability to select salient lu-shapelets as features with good clustering results. We attribute the significant improvement to the following reasons. First, USSL learns the pseudo-class label indicators and the best lu-shapelets simultaneously, which enables USSL to learn the most informative and salient features in a supervised learning way. Second, the similarity of the lu-shapelets and the spectral structure of the raw data are explored simultaneously, which means that similar time series share the same pseudo-class label and the lu-shapelets are devised. Thus, clustering performance improved. Third, the lu-shapelets are learned from the original time series



(a) Coffee (b) DiatomSizeReduction

(c) ECG200 (d) SonyAIBoRobotSurfaceII

Fig. 3. An illustration of USSL on the four datasets: Coffee, DiatomSizeReduction, ECG200 and SonyAIBoRobotSurfaceII. The left part of each figure shows the two lu-shapelets S1 and S2 learned from the original time series. The middle shows the closest match of the lu-shapelets to the original time series examples, where T_1 and T_2 are drawn from one of the classes and T_3 and T_4 are drawn from the other class. We can see that lu-shapelets are informative features. The right shows the shapelet-transformed plots of the original time series dataset. Different colors show different class labels. The dots in red circles and blue squares are the labels obtained by USSL, which closely matches the original class labels.

rather than selected from the candidate segments which avoids redundant noise.

Due to space limitations, Fig. 3 only shows an illustration of USSL on four datasets: Coffee; DiatomSizeReduction; ECG200 and SonyAIBORobotSurfaceII. For each dataset, we illustrated two lu-shapelets first, then the mapped time series in two-dimensional shapelet space. Intuitively, USSL can extract the most significant shapelets for clustering from the original data, which makes each time series distinguishable. After mapping the original time series to the shapelet-transformed space for all four datasets, clear pseudo classification boundaries were obvious. The dots in red circles and blue squares are the cluster labels obtained by USSL, which closely match the original class labels.

6.5 Time Series with Unequal Length

Here, we show the generalizability of USSL with a synthetic dataset that is generated by following the work of [63] and [10]. It includes 10 sinusoidal-signal sequences of 200 in length and 10 rectangular-signal sequences of 100 in length.

TABLE 4
Experiment on Synthetic Dataset

	Name	Number	Length
Synthetic Dataset	Rectangular	10	100
	Sinusoidal	10	200
lu-shapelets	Shapelet 1		12
	Shapelet 2		30
Results	RI		1.00
	NMI		1.00

Gaussian noise was embedded in the original data, which was generated by five Gaussian processes that had a mean of $\nu \in [0, 2]$ and a variance of $\sigma^2 \in [0, 10]$, chosen uniformly at random. Details are showed in Table 4.

Fig. 4 illustrates the details of two lu-shapelets learned by USSL. The first one is a sharp spike in length of 12 that matches the most prominent feature of the rectangular signal. The second one was 30 in length and very similar to the standard curve of the sinusoidal signal. We see USSL was able to automatically learn the representative shapelets from unsupervised time series. USSL's generalizability was also verified by handling both time series and shapelets of unequal length. Additionally, USSL produced the highest RI and NMI value of 1 on this dataset, which equates to a perfect clustering result. By contrast, the work in [63] only reached 0.9, even when using class label information during training.

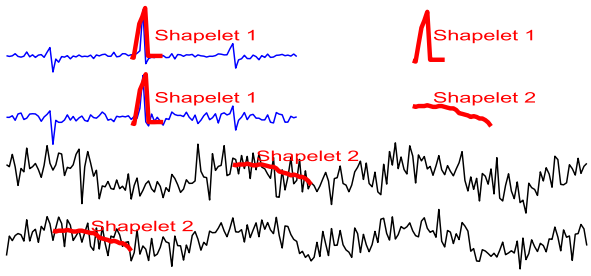


Fig. 4. An example of the generalizability of USSL on sinusoidal signals and rectangular signals. The top left blue curves are rectangular signals with Gaussian noise of length 100 while the bottom black curves are sinusoidal signals with Gaussian noise of length 200. The top right short curves marked in bold red are two learned shapelets of lengths 12 and 30 respectively. We see that USSL can generally handle both the time series and shapelets of different length. The RI and NMI obtained by USSL for this dataset are both 1, which means perfect clustering.

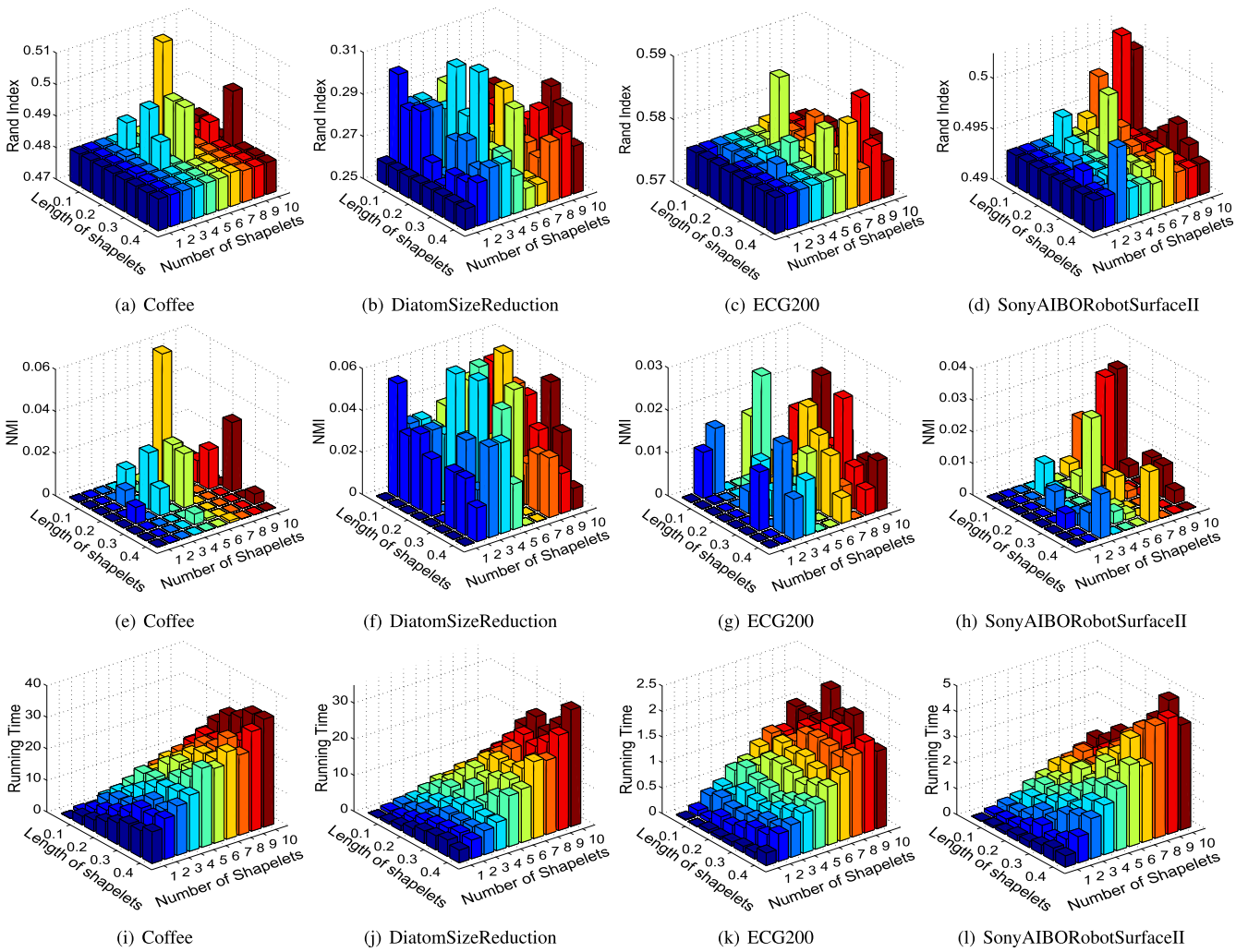


Fig. 5. The performance of USSL with respect to the number and the length of the lu-shapelets (measured by the Rand index, the NMI and the running time). In terms of RI and NMI, USSL reaches its best performance at the top-middle or top-right corners. More running time is needed when the number and the length of the lu-shapelets increase. This shows that the USSL does not need to learn long lu-shapelets since short lu-shapelets achieve sufficient or better clustering performance and also save time. In terms of the number of lu-shapelets, it is not necessary to set a large value. In most of the datasets, too many lu-shapelets would lead to repeating or overlapping. For the time series sets with complex lines, like SonyAIBORobotSurfaceII, a reasonably large number of lu-shapelets can be set to learn the numerous distinct curves in original time series. Thus, both length and number of lu-shapelets do not need to be large values in practice.

6.6 Parameter Study

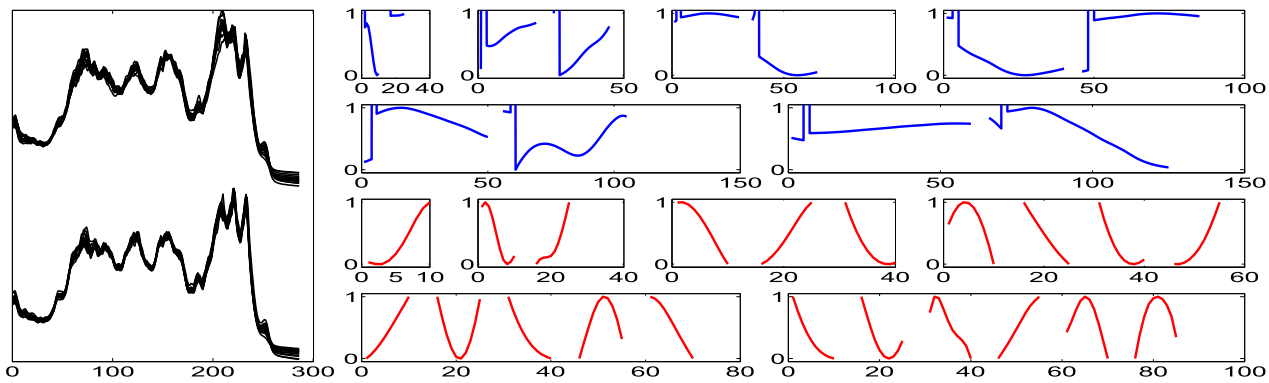
In this section, we tested two key USSL parameters: the number of lu-shapelets needed to learn k, and the minimum length of lu-shapelets l_{min}. We varied k from 1 to 10, with a step of 1, and l_{min} from 0.05 to 0.4 of the total length of original time series, with a step of 0.05. The parameter r was set to 1 for simplicity, which means we only learned lu-shapelets of the same length (l_{min}) in this experiment.

Fig. 5 shows the variations in the RI, NMI, and running time using different lengths and different numbers of lu-shapelets learned on four datasets. We can see RI and NMI values reached their peak at the top-middle area on the first three datasets, and at the top-right corner on the SonyAIBORobotSurfaceII dataset. The running time continuously increased when the number or length of the lu-shapelets increased. This means USSL does not need to learn long lu-shapelets; short lu-shapelets achieve sufficient or better clustering performance. It also saves time. Further, there is no need to learn too many lu-shapelets (in most datasets, this leads to repeating or overlapping). For time series sets with complex lines, like SonyAIBORobotSurfaceII, a reasonably large number of shapelets can be set to learn the numerous distinct shapes of the original time series.

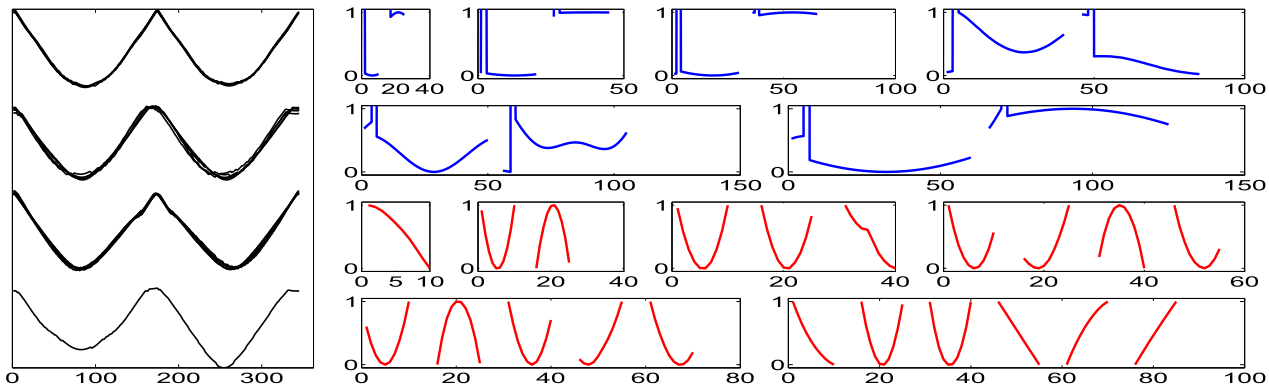
Fig. 6 illustrates the curves of lu-shapelets learned by USSL. The length of lu-shapelets is varied from 10 to 60, with a step of 10 whilst first fixing the lu-shapelets number as 2. The learned lu-shapelets are shown in blue. Subsequently, we vary shapelet number from 1 to 6 with a fixed lu-shapelets length as 10. These lu-shapelets are shown in red. The results show that when increasing the length of lu-shapelets, there was a heavy overlap in lu-shapelets. When varying the lu-shapelets number, the lu-shapelets changed, but they still overlaped. These observations confirm the previous conclusions that, in practice, the length and the number of lu-shapelets does not need to be large values.

6.7 Running Time and Convergence Curve

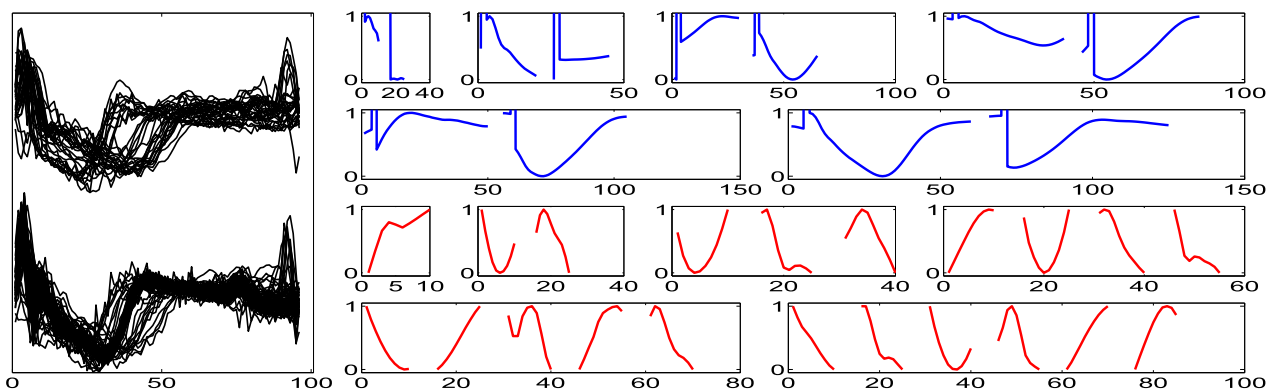
This section explores the running time variation with respect to dataset size and length of original time series and the convergence curves of the USSL.



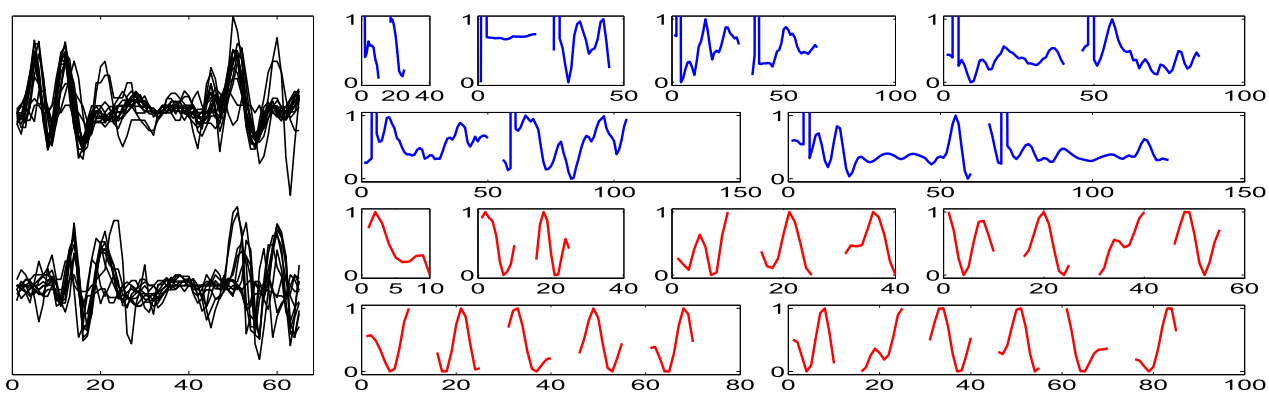
(a) Coffee



(b) DiatomSizeReduction

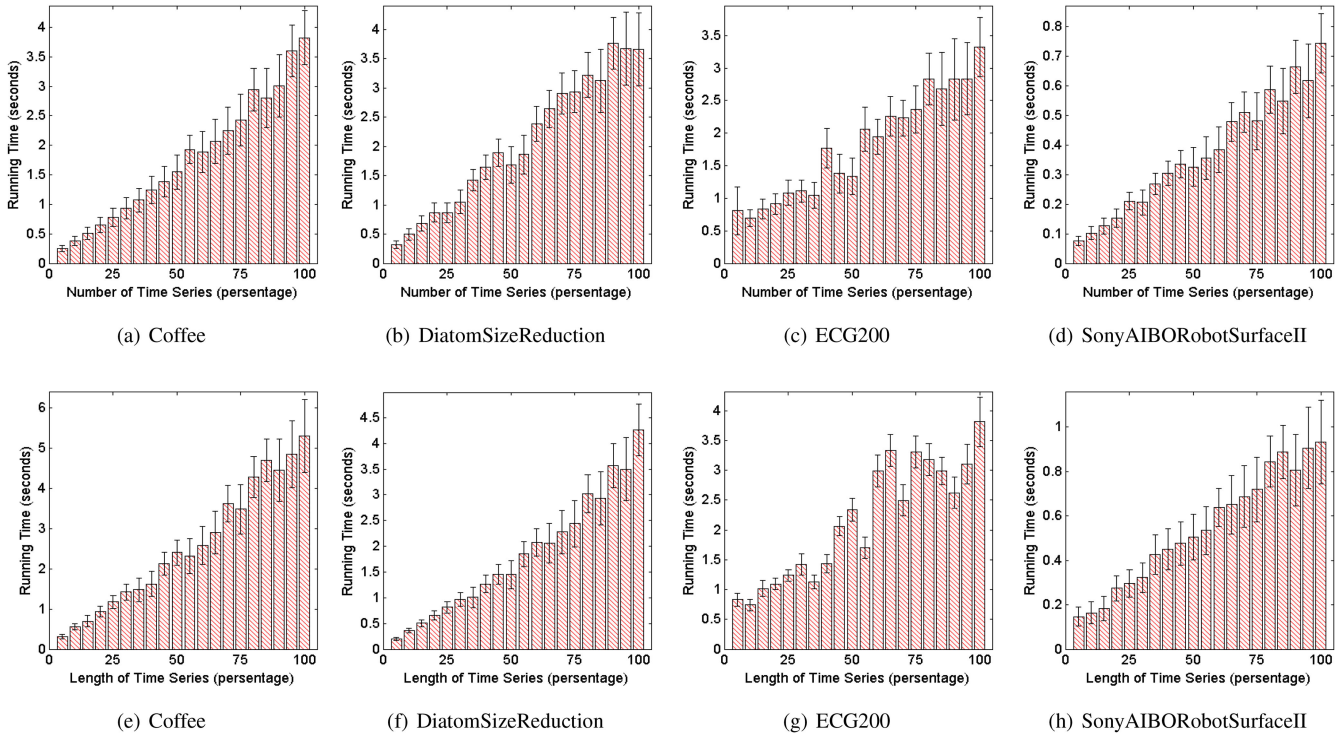


(c) ECG200



(d) SonyAIBORobotSurfaceII

Fig. 6. An illustration of the shapelets learned by USSL on the four real-world datasets. The left shows some original time series examples. Time series from the same class are shown together. The right shows the obtained lu-shapelets. lu-shapelets in blue are learned by increasing the length from 10 to 60 with a step 10. lu-shapelets in red are learned by increasing the number from 1 to 6 with a step of 1. We see when increasing the length of lu-shapelets, there is a heavy overlap in lu-shapelets. When varying the lu-shapelets number, the lu-shapelets change but they still overlap. These observations confirm that in practice the length and the number of lu-shapelets do not need to be large values.



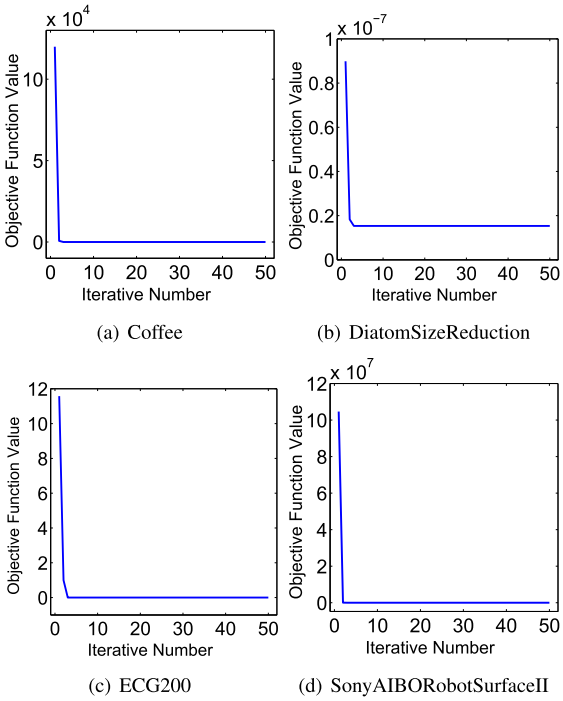
(a) Coffee (b) DiatomSizeReduction (c) ECG200 (d) SonyAIBORobotSurfaceII

(e) Coffee (f) DiatomSizeReduction (g) ECG200 (h) SonyAIBORobotSurfaceII

Fig. 7. Running time with respect to different number and length of time series.

We ranged the length and number of training time series from 5 to 100 percent, respectively, of the original datasets, with a step of 5 percent in the first two experiments, while the related other parameters were fixed.

The reported running time was an average of 10 repeated experiments. Fig. 7 shows the experimental results. We can see running time increased approximately linearly with the length or the number of training time series, which means it scales well to large datasets.



(a) Coffee (b) DiatomSizeReduction

(c) ECG200 (d) SonyAIBORobotSurfaceII

Fig. 8. Convergence curves of USSL over Coffee, DiatomSizeReduction, ECG200 and SonyAIBORobotSurfaceII datasets.

In the third experiment, we explored USSL's convergence curves. As shown in Algorithm 1, the algorithm is an iterative process. The convergence curves of the USSL's iterative process over the four datasets are presented in Fig. 8. We can observe that USSL is effective and converges quickly.

7 Conclusions

The paper proposes a new optimization model for unsupervised salient subsequence learning in time series, called USSL, by integrating the strengths of shapelet learning, spectral analysis, pseudo-class labels, and least-squares minimization. The proposed model can automatically learn the salient shapelets to help time series clustering. Results for both real-world and synthetic datasets show USSL can learn meaningful unsupervised shapelets with promising performance compared to the state-of-the-art unsupervised time series learning methods.

Acknowledgments

This work was supported in part by Australia's ARC Discovery Project (DP140100545, DP140102206) and Australia's ARC Linkage project (LP150100671, LP160100630), in part by the MQNS under Grant 9201701203, in part by the Collaborative Research Project (CRP) between Macquarie University and Data61 on dynamic graph mining.

References

[1] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning time-series shapelets," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 392–401.

[2] E. J. Ruiz, V. Hristidis, C. Castillo, A. Gionis, and A. Jaimes, "Correlating financial time series with micro-blogging activity," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 513–522.

[3] S. J. Taylor, *Modeling Financial Time Series*. Singapore, World Scientific Publishing, 2007.

[4] S. Hirano and S. Tsumoto, "Cluster analysis of time-series medical data based on the trajectory representation and multiscale comparison techniques," in *Proc. 6th Int. Conf. Data Mining*, 2006, pp. 896–901.

[5] E. Keogh, J. Lin, A. W. Fu, and H. VanHerle, "Finding unusual medical time-series subsequences: Algorithms and applications," *IEEE Trans. Inf. Technol. Biomed.*, vol. 10, no. 3, pp. 429–439, Jul. 2006.

[6] Y. Cai and R. Ng, "Indexing spatio-temporal trajectories with Chebyshev polynomials," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2004, pp. 599–610.

[7] P. Héas and M. Datcu, "Modeling trajectory of dynamic clusters in image time-series for spatio-temporal reasoning," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 7, pp. 1635–1647, Jul. 2005.

[8] D. Gong, G. Medioni, and X. Zhao, "Structured time series analysis for human action segmentation and recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1414–1427, Jul. 2014.

[9] F. Zhou and F. De La Torre, "Generalized canonical time warping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 279–294, Feb. 2016.

[10] J. Zakaria, A. Mueen, and E. Keogh, "Clustering time series using unsupervised-shapelets," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 785–794.

[11] M. G. Baydogan, G. Runger, and E. Tuv, "A bag-of-features framework to classify time series," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2796–2802, Nov. 2013.

[12] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 1855–1870.

[13] L. Ye and E. Keogh, "Time series shapelets: A new primitive for data mining," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 947–956.

[14] A. Mueen, E. Keogh, and N. Young, "Logical-shapelets: An expressive primitive for time series classification," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 1154–1162.

[15] T. Rakthanmanon and E. Keogh, "Fast shapelets: A scalable algorithm for discovering time series shapelets," in *Proc. SIAM Int. Conf. Data Mining*, 2013, pp. 668–676.

[16] K.-W. Chang, B. Deka, W.-M. W. Hwu, and D. Roth, "Efficient pattern-based time series classification on GPU," in *Proc. IEEE 12th Int. Conf. Data Mining*, 2012, pp. 131–140.

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

[18] A. Bagnall, J. Lines, J. Hills, and A. Bostrom, "Time-series classification with COTE: The collective of transformation-based ensembles," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2522–2535, Sep. 2015.

[19] P. K. Vemulapalli, V. Monga, and S. N. Brennan, "Robust extrema features for time-series data analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1464–1479, Jun. 2013.

[20] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1734–1747, Sep. 2016.

[21] A. Romero, P. Radeva, and C. Gatta, "Meta-parameter free unsupervised sparse feature learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 8, pp. 1716–1722, Aug. 2015.

[22] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Proc. 18th Int. Conf. Neural Inf. Process. Syst.*, 2005, pp. 507–514.

[23] Z. Zhao and H. Liu, "Spectral feature selection for supervised and unsupervised learning," in *Proc. 24th Int. Conf. Mach. Learn.*, 2007, pp. 1151–1157.

[24] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 333–342.

[25] Z. Zhao, L. Wang, and H. Liu, "Efficient spectral feature selection with minimum redundancy," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 673–678.

[26] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "l2, 1-norm regularized discriminative feature selection for unsupervised learning," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1589–1594.

[27] Z. Li, Y. Yang, J. Liu, X. Zhou, H. Lu, et al., "Unsupervised feature selection using nonnegative spectral analysis," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 1026–1032.

[28] L. Shi, L. Du, and Y.-D. Shen, "Robust spectral learning for unsupervised feature selection," in *Proc. IEEE Int. Conf. Data Mining*, 2014, pp. 977–982.

[29] Z. Li, J. Liu, J. Tang, and H. Lu, "Robust structured subspace learning for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2085–2098, Oct. 2015.

[30] Z. Li and J. Tang, "Unsupervised feature selection via nonnegative spectral analysis and redundancy control," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5343–5355, Dec. 2015.

[31] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu, "Clustering-guided sparse structural learning for unsupervised feature selection," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2138–2150, Sep. 2014.

[32] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint l2, 1-norms minimization," in *Proc. 23rd Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.

[33] I. J. Goodfellow, A. Courville, and Y. Bengio, "Scaling up spike-and-slab models for unsupervised feature learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1902–1914, Aug. 2013.

[34] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1178–1192, May 2013.

[35] M. Qian and C. Zhai, "Robust unsupervised feature selection," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1621–1627.

[36] J. Zakaria, A. Mueen, E. Keogh, and N. Young, "Accelerating the discovery of unsupervised-shapelets," *Data Mining Knowl. Discovery*, vol. 30, no. 1, pp. 243–281, 2016.

[37] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering–A decade review," *Inf. Syst.*, vol. 53, pp. 16–38, 2015.

[38] L. Ulanova, N. Begum, and E. Keogh, "Scalable clustering of time series with U-shapelets," in *Proc. SIAM Int. Conf. Data Mining*, 2015, pp. 900–908.

[39] Y. Li, M. L. Yiu, Z. Gong, et al., "Efficient discovery of longest-lasting correlation in sequence databases," *VLDB J.*, vol. 25, pp. 1–24, 2016.

[40] J. Frank, S. Mannor, J. Pineau, and D. Precup, "Time series analysis using geometric template matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 740–754, Mar. 2013.

[41] H. A. Dau and N. B. E. Keogh, "Semi-supervision dramatically improves time series clustering under dynamic time warping," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, 2016, pp. 978–987.

[42] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh, "Searching and mining trillions of time series subsequences under dynamic time warping," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 262–270.

[43] M. Wistuba, J. Grabocka, and L. Schmidt-Thieme, "Ultra-fast shapelets for time series classification," *CoRR*, vol. abs/1503.05018, 2015. [Online]. Available: http://arxiv.org/abs/1503.05018

[44] P. Esling and C. Agon, "Time-series data mining," *ACM Comput. Surv.*, vol. 45, no. 1, 2012, Art. no. 12.

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

[46] J. Lines, L. M. Davis, J. Hills, and A. Bagnall, "A shapelet transform for time series classification," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 289–297.

[47] Q. Zhang, J. Wu, H. Yang, Y. Tian, and C. Zhang, "Unsupervised feature learning from time series," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 2322–2328.

[48] L. Ye and E. Keogh, "Time series shapelets: A novel technique that allows accurate, interpretable and fast classification," *Data Mining Knowl. Discovery*, vol. 22, no. 1–2, pp. 149–182, 2011.

[49] Q. He, F. Zhuang, T. Shang, Z. Shi, et al., "Fast time series classification based on infrequent shapelets," in *Proc. 11th Int. Conf. Mach. Learn. Appl.*, 2012, pp. 215–219.

[50] Y. Hao, Y. Chen, J. Zakaria, B. Hu, T. Rakthanmanon, and E. Keogh, "Towards never-ending learning from time series streams," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 874–882.

[51] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.

[52] W. E. Donath and A. J. Hoffman, "Lower bounds for the partitioning of graphs," *IBM J. Res. Develop.*, vol. 17, no. 5, pp. 420–425, 1973.

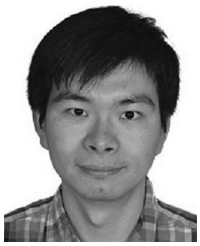
[53] J. Tang and H. Liu, "An unsupervised feature selection framework for social media data," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 12, pp. 2914–2927, Dec. 2014.

[54] C. T. Kelley, *Iterative Methods for Optimization*. Philadelphia, PA, USA: SIAM, 1999.

[55] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[56] M. S. Cetin, A. Mueen, and V. D. Calhoun, "Shapelet ensemble for multi-dimensional time series," in *Proc. SIAM Int. Conf. Data Mining*, 2015, pp. 307–315.
[57] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Statistical Assoc.*, vol. 66, no. 336, pp. 846–850, 1971.
[58] H. Zhang, T. B. Ho, Y. Zhang, and M.-S. Lin, "Unsupervised feature extraction for time series clustering using orthogonal wavelet transform," *Informatica*, vol. 30, no. 3, pp. 305–319, 2006.
[59] S. C. Madeira, M. C. Teixeira, I. Sa-Correia , and A. L. Oliveira, "Identification of regulatory modules in time series gene expression data using a linear time biclustering algorithm," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 7, no. 1, pp. 153–165, Jan.–Mar. 2010.
[60] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, no. 2–3, pp. 107–145, 2001.
[61] J. Yang and J. Leskovec, "Patterns of temporal variation in online media," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 177–186.
[62] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognit.*, vol. 44, no. 3, pp. 678–693, 2011.
[63] S. Shariat and V. Pavlovic, "Isotonic CCA for sequence alignment and activity recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2572–2578.

Qin Zhang received the master's degree from the University of Chinese Academy of Sciences, China, in 2014. She is working toward the doctoral degree in the Centre for Artificial Intelligence (CAI), University of Technology Sydney, Australia. She is currently with the Data Centre of Social Network Group (SNG), Tencent, China Her main research interests include sequence data learning and network analysis by using various deep learning and optimisation methods. She has served as a reviewer (sub-reviewer) for KDD, NIPS, ICDM, IJCAI, AAAI and SDM.

Jia Wu (M'16) received the PhD degree in computer science from the University of Technology Sydney, Australia. He is currently a lecturer with the Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, Australia. His research focuses on data mining and machine learning. Since 2009, he has published more than 100 top-tier journals (such as the *IEEE Transactions on Knowledge and Data Engineering*, the *IEEE Transactions on Cybernetics*, the *ACM Transactions on Knowledge Discovery Data, Pattern Recognition*) and conference papers (such as *International Joint Conference on Artificial Intelligence* (IJCAI), *AAAI Conference* on *Artificial Intelligence* (AAAI), IEEE *International Conference* on *Data Mining* (ICDM), IEEE *International Conference* on *Data Engineering* (ICDE), SIAM *International Conference* on *Data Mining* (SDM), ACM *Conference* on *Information and Knowledge Management* (CIKM)) in these areas. He was the recipient of SDM'18 Best Paper Award in Data Science Track, IJCNN'17 Best Student Paper Award, and ICDM14 Best Paper Candidate Award. He is the Associate Editor of *ACM Transactions* on *Knowledge Discovery from Data*. He is a member of the IEEE.



Peng Zhang received the PhD degree from the Chinese Academy of Sciences, Beijing, China, in 2009. Since then, he has been with one national engineering laboratory in China, two universities in the USA, and University of Technology Sydney, Ultimo, NSW, Australia. He is currently a senior data analyst with the Ant Financial Services Group, Hangzhou, China. His current research interests include machine learning and data mining. He is an associate editor for two Springer journals: the *Journal of Big Data*, and the *Annals of Data Science*. To date, he has published more than 100 research papers, including TKDE, KDD, ICDM, SDM, IJCAI, AAAI, CIKM, WWW.

Guodong Long received the PhD degree from the University of Technology Sydney (UTS), Australia, in 2014. He is currently a senior lecturer with the Research Centre for Artificial Intelligence (CAI) in UTS. His research interests include data mining, machine learning, artificial intelligence, social network analytics and healthcare informatics.

Chengqi Zhang (SM'95) received the PhD degree from the University of Queensland, Brisbane, Australia, in 1991 and the DSc degree (higher doctorate) from Deakin University, Geelong, Australia, in 2002. Since December 2001, he has been a professor of Information Technology with the University of Technology Sydney (UTS), Australia, where he was director of the UTS Priority Investment Research Centre for Quantum Computation and Intelligent Systems from 2008-2016. He is the Associate Vice President of UTS since 2017. He has published more than 300 research papers, including several in first-class international journals, such as the *Artificial Intelligence*, IEEE, and ACM Transactions. He has published six monographs and edited 16 books and has attracted 13 Australian Research Council grants. His research interests mainly focus on data mining and its applications. He has been serving as an associate editor for three international journals, including the *IEEE Transactions on Knowledge and Data Engineering* (2005-2008). He was general co-chair of KDD 2015 in Sydney and the Local Arrangements chair of IJCAI-2017 in Melbourne, and a fellow of the Australian Computer Society and a Senior Member of the IEEE.