OCLTS: One-Class Learning Time-Series Shapelets

Akihiro Yamaguchi[1], Takeichiro Nishikawa[1]
[1] System Engineering Lab., Corporate R&D Center, Toshiba Corporation, Japan
akihiro5.yamaguchi@toshiba.co.jp

***Abstract*—Time-series shapelets are time-series segments effective for classifying time-series instances. In recent years, simultaneously learning both a classifier and the shapelets has been studied because they provide not only interpretable results but also superior classification performance. However, in some applications such as anomaly detection, class distributions are highly imbalanced between majority and minority classes. In particular, it is important to detect unseen features, which do not appear during training, in a minority class if those features are discriminative. Our aim is to learn both a classifier and shapelets using only training instances for the majority class without the minority class. We propose a method called One-Class Learning Time-series Shapelets (OCLTS). OCLTS efficiently and simultaneously optimizes both the shapelets and a nonlinear classifier based on One-Class Support Vector Machine by a stochastic subgradient descent algorithm. Experimental results show the method's effectiveness for interpretability and imbalanced binary classification.**

***Keywords*-Time-series shapelets; One-class support vector machine; Imbalanced binary classification**

I. Introduction

The Internet of Things (IoT) has spurred the development of time-series classification technology by machine learning in fields such as medicine, disaster response, and manufacturing. In particular, studies on the discovery of shapelets (time-series segments effective for classifying time-series instances) by learning classifiers have attracted considerable interest [1], [2]. The idea is that discriminative features appear only in a few segments, not throughout an entire time series. The advantages of shapelet-based methods are (1) quick prediction of test instances once learning both of the shapelets and the classifier is complete; (2) high classification performance in experiments; and (3) interpretability for domain experts.

In recent years, several methods for learning shapelets have been proposed [2], [3], [4] although the study of shapelets started from search-based methods, which rank all segments in time-series instances and repeatedly extract discriminative segments as shapelets [1]. These learning-based methods learn both shapelets and classifiers from rough initial shapelets by using gradient descent algorithms. The advantages of the learning-based methods are (1) reducing algorithmic complexity; (2) improving classification performance in experiments; and (3) learning shapelets robust to noise in the original time series.

On the other hand, a highly imbalanced binary class distribution between majority and minority classes is an important classification problem [5], [6], because minority classes are rare and some features may not appear in the training instances. For example, anomalies should be detected even if they are rare relative to normal instances and are unseen during training. One approach for dealing with this problem is one-class learning, and the One-Class Support Vector Machine (OCSVM) is a well-known method [7].

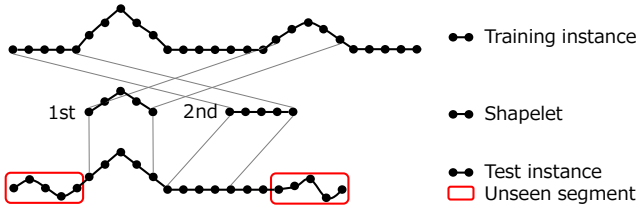

Figure 1. Best-matching approach in existing shapelet-based methods. It is difficult to match unseen discriminative segments, which do not appear during training, and to capture the features.

However, one-class learning has not yet been incorporated into shapelet-based methods to solve this problem. All of the existing shapelet-based methods use feature vectors based on best-matching distances between shapelets and the time-series instance. This conventional approach is inappropriate if an insufficient amount of training data for the minority class is collected for identifying discriminative features.

Example: Fig. 1 shows the problem with the best-matching approach. Each instance of the majority class is a time-series instance consisting of two time-series segments (horizontal and peak segments). During a test, the time-series instance of the minority class has unseen but discriminative segments that are not found during training (the two parts surrounded by red boxes). In this case, shapelets might not match the segments. As a result, the discriminative difference is not reflected in the feature vectors and is thus ignored.

Let us consider anomaly detection that needs to find abnormal time-series segments. If abnormal segments appear during a test and do not appear during training, the shapelets of the best-matching approach cannot match the abnormal segments, as in the red boxes in Fig. 1. As a result, existing shapelet-based methods fail to detect the anomaly.

We propose a method called One-Class Learning Time-series Shapelets (OCLTS) to simultaneously learn a one-class classifier and shapelets by using training data of only the majority class. In OCLTS, we formulate an integrated

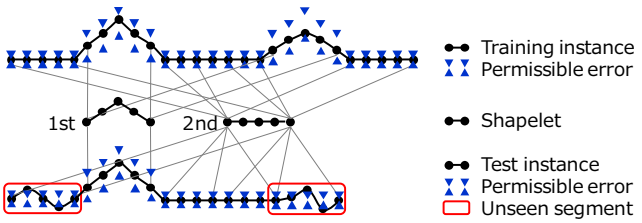

Figure 2. The permissible errors approach in OCLTS. This approach can cover unseen discriminative segments by measuring permissible errors for segments that cover the time series, and can capture the features.

optimization problem to learn both of the shapelets and the classifier, and find the solution by using a stochastic subgradient descent algorithm.

In addition, OCLTS uses feature vectors based on permissible errors for segments that cover time series when shapelets approximate the time series. The permissible errors approach detects when some of the minority time-series segments deviate from the permissible error, such as the two parts surrounded by red boxes in Fig. 2. As a result, OCLTS can detect discriminative segments even if they do not appear during training and can overcome the problems with the best-matching approach.

The main contributions are summarized as follows:

- We propose OCLTS, which
  (a) learns shapelets and a one-class classifier simultaneously from only the majority class;
  (b) learns a nonlinear classifier based on kernelized OCSVM;
  (c) reduces algorithmic complexity by using a stochastic subgradient descent algorithm; and
  (d) uses feature vectors suitable for one-class classification based on permissible errors.
- We confirm the superiority of OCLTS compared to state-of-the-art shapelet-based methods [2] for imbalanced classification, and demonstrate the interpretability of OCLTS on both synthetic and real-world datasets.

This paper is the extended version of our previous work [8], and is organized as follows. Section II provides an overview of related work. In Section III, we introduce the preliminaries and problem setting. In Section IV, we propose OCLTS. After introducing the experiments in Section V, we evaluate OCLTS in Section VI. Our conclusions are provided in Section VII.

II. Related Work

In this section, we present related work for two aspects of shapelet-based methods and one-class classification.

A. Time-Series Shapelets

Time-series shapelets have mainly been proposed as time-series segments that are discriminative for predicting class labels if given all of the class labels during training. The first method discovers shapelets by repeatedly searching time-series segments while learning a decision tree [1]. Training in this kind of search-based method is time-consuming although the prediction is fast. For that reason, speed-up techniques such as reusing computations, pruning the search space, and hardware (GPU)-based optimizations have been proposed [9], [10], [11]. In another aspect, shapelet-based methods have been extended from classification to clustering where no class labels are given during training [12].

Instead of exhaustively searching for shapelets from time-series segments, a learning-based method has been proposed [2]. This method simultaneously learns shapelets and classifiers based on logistic regression by using a gradient descent algorithm, and improves classification performance while reducing algorithmic complexity for training. In addition, learning-based methods that focus on higher speed [4] and unsupervised clustering [3], [13] have been proposed. Learning-based methods can provide shapelets that can be interpreted as excluding noise of original time series [3], [13].

However, no existing shapelet-based methods adopt one-class classification. We noticed a similar work [14]. Although that method can deal with imbalanced class distributions by cost-sensitive classification, it is not one-class learning and requires all class labels based on the best-matching approach. Therefore, no existing shapelet-based methods are suitable if discriminative features appear outside of the best-matching segments, as shown in Fig. 1.

B. One-Class Classification

In domains such as medical diagnostics and monitoring, fraud or intrusion detection, and risk management, the ratio of minority to majority classes can have drastic imbalances such as 1:100, 1:1,000, 1:10,000 (or sometimes even more) [5], [6]. In such cases, standard classifiers tend to ignore minority classes. To solve this problem, methods based on re-sampling, cost-sensitive learning, and one-class learning have been studied [5].

One-class learning is effective when few minority training instances are collected or when the the distribution of classes is drastically imbalanced (minority ratio of 3% or less) [15]. OCSVM [7] is a representative one-class learning method that adapts Support Vector Machine (SVM) for one-class learning. OCSVM can efficiently learn a nonlinear classifier using the kernel trick in the same manner as SVM. However, it is difficult for most existing one-class learning methods to deal with time-series structure such as the ordering of attributes and phase shift. Although a one-class learning method that can deal with time-series has been proposed in recent years [16], it cannot discover time-series segments that are effective for classification.

Shapelets have attracted attention for interpretable classification and clustering. However, to the best of our knowledge,

no existing methods based on one-class learning can learn discriminative time-series segments.

III. Preliminaries and Problem Setting

In this section, we show the formulation of OCSVM, which we partially use in the formulation of OCLTS, and describe the variables learned by OLCTS.

A. OCSVM: One-Class Support Vector Machine

OCSVM can deal with a nonlinear one-class classification using a kernel trick, which maps feature vectors through a kernel function to a higher (potentially infinite) dimensional space that is thus more discriminative.

Given I' majority training instances, OCSVM maps the ith feature vector $\mathbf{X'}_i$ to a high-dimensional feature space by function $\phi(\mathbf{X'}_i)$ for $i = 1, 2, \cdots, I'$, and separates the regions of majority and minority classes by the hyperplane in $\mathbf{w}$ so as to maximize the margin in the high-dimensional space. Pegasos [17] is the algorithm used to learn SVM, and it can be naturally applied to OCSVM in the same manner as in [18]. The optimization problem is formulated in

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}}\, \frac{\lambda}{2}||\mathbf{w}||^2 + \frac{1}{I'}\sum_{i=1}^{I'} \mathbf{h}_i \qquad (1)\\ &\mathbf{h}_i = \max\{0, 1 - \langle \mathbf{w}, \phi(\mathbf{X'_i})\rangle\} \end{aligned}$$

where λ is a meta-parameter and loss function $\mathbf{h}_i$ is hinge loss.

The representer theorem allows us to express the optimal solution as a linear combination such as $\mathbf{w} = \sum_{i=1}^{I'} \mathbf{a'_i}\phi(\mathbf{X'_i})$, where $\mathbf{a'}$ is an I'-dimensional vector [19]. For this reason, we can rewrite the primary problem (1) using kernel function $\mathcal{K}$ without rewriting it as a dual problem:

$$\begin{aligned} &\underset{\mathbf{a'}}{\text{minimize}}\, \frac{\lambda}{2}\sum_{i,i'=1}^{I'} \mathbf{a'}_{i'}\mathbf{a'}_i\mathcal{K}(\mathbf{X'}_{i'}, \mathbf{X'}_i) + \frac{1}{I'}\sum_{i=1}^{I'} \mathbf{h}_i \qquad (2)\\ &\mathbf{h}_i = \max\left\{0, 1 - \sum_{i'=1}^{I'} \mathbf{a'}_{i'}\mathcal{K}(\mathbf{X'}_{i'}, \mathbf{X'}_i)\right\} \end{aligned}$$

where $\mathbf{X}_i$ and $\mathbf{a}_i$ are a support vector and the weight factor in case of $\mathbf{a}_i \neq 0$, respectively.

Equation (2) is formulated without explicitly using function ϕ, and the kernel trick can be used. Kernelized Pegasos is the algorithm used to solve the optimization problems in (2) by a stochastic subgradient descent algorithm with high classification performance [17].

B. Problem Setting

Our goal is to learn shapelets $\mathbf{S} \in \mathbb{R}^{K\times L}$ and a one-class classifier from time-series instances $\mathbf{T} \in \mathbb{R}^{I\times Q}$ of a majority class, where I training instances of time-series length Q, number K of shapelets, and shapelet length L are given. We denote the jth value of the ith time series $\mathbf{T}_i$ as $\mathbf{T}_{i,j}$, and the lth value of the kth shapelet $\mathbf{S}_k$ as $\mathbf{S}_{k,l}$. There are a total of $J := Q - L + 1$ segments of length L for each time-series instance. We measure the distance between the jth time-series segment $(\mathbf{T}_{i,j}, \mathbf{T}_{i,j+1}, \cdots, \mathbf{T}_{i,j+L-1})$ of $\mathbf{T}_i$ and $\mathbf{S}_k$ by the Euclidean distance given in

$$\mathbf{D}_{i,k,j} = \frac{1}{L}\sum_{l=1}^{L}(\mathbf{T}_{i,j+l-1} - \mathbf{S}_{k,l})^2. \qquad (3)$$

In this study, shapelets are learned to determine $\mathbf{S}$ that is effective for one-class classification and robust to noise.

Given a feature vector $\mathbf{x}$ of a test instance, the discriminant function based on OCSVM is

$$f(\mathbf{x}) = \sum_{i=1}^{I} \mathbf{a}_i\mathcal{K}(\mathbf{X}_i, \mathbf{x}) \qquad (4)$$

where $\mathbf{X}_i$ is the feature vector of $\mathbf{T}_i$, $\mathcal{K}$ is the kernel function, and $\mathbf{a}$ is an I-dimensional vector. Eq. (4) has a high value if $\mathbf{x}$ tends to belong to the majority class. In this study, a classifier is learned to determine model parameters $\mathbf{a}$.

IV. One-Class Learning Time-Series Shapelets

In this section, we show the learning process using an illustrative example in order to provide an intuitive understanding of the basic idea. Next, we describe how the time series is approximated by shapelets, and define a feature vector based on permissible errors for the approximation. After that, we formulate the objective function, and derive gradients for optimizing it. Finally, we present a learning algorithm using the gradients, and summarize its algorithmic complexity.

A. Illustration of the Mechanism

As shown in Fig. 3, OCLTS learns both shapelets $\mathbf{S}$ that approximate $\mathbf{T}$ and OCSVM-based model parameters $\mathbf{a}$ simultaneously by a gradient descent algorithm. OCLTS can learn nonlinear classifiers by using a kernel trick through formulation through a kernel function. We reduce the algorithmic complexity of OCLTS by formulation based on both a subgradient descent algorithm and a stochastic gradient descent algorithm. The feature vector $\mathbf{X}_i$ of $\mathbf{T}_i$ is a K-dimensional vector, and is based on the permissible errors when $\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_K$ fit the segments of $\mathbf{T}_i$.

In the example in Fig. 3, we show the learning processes of $\mathbf{S}$ and $\mathbf{a}$ where $(K, L) = (2, 5)$. $\mathbf{T}$ is a synthetic dataset consisting of a horizontal segment and a peak segment with a small amount of Gaussian noise. Minority class instances have as an unseen discriminative feature a more gradual peak segment than the majority class instances. The top, middle, and bottom rows of Fig. 3 are at the 1st, 20th, and 500th iteration of the gradient descent algorithm, respectively. The leftmost column shows growth of $\mathbf{S}$. The center column

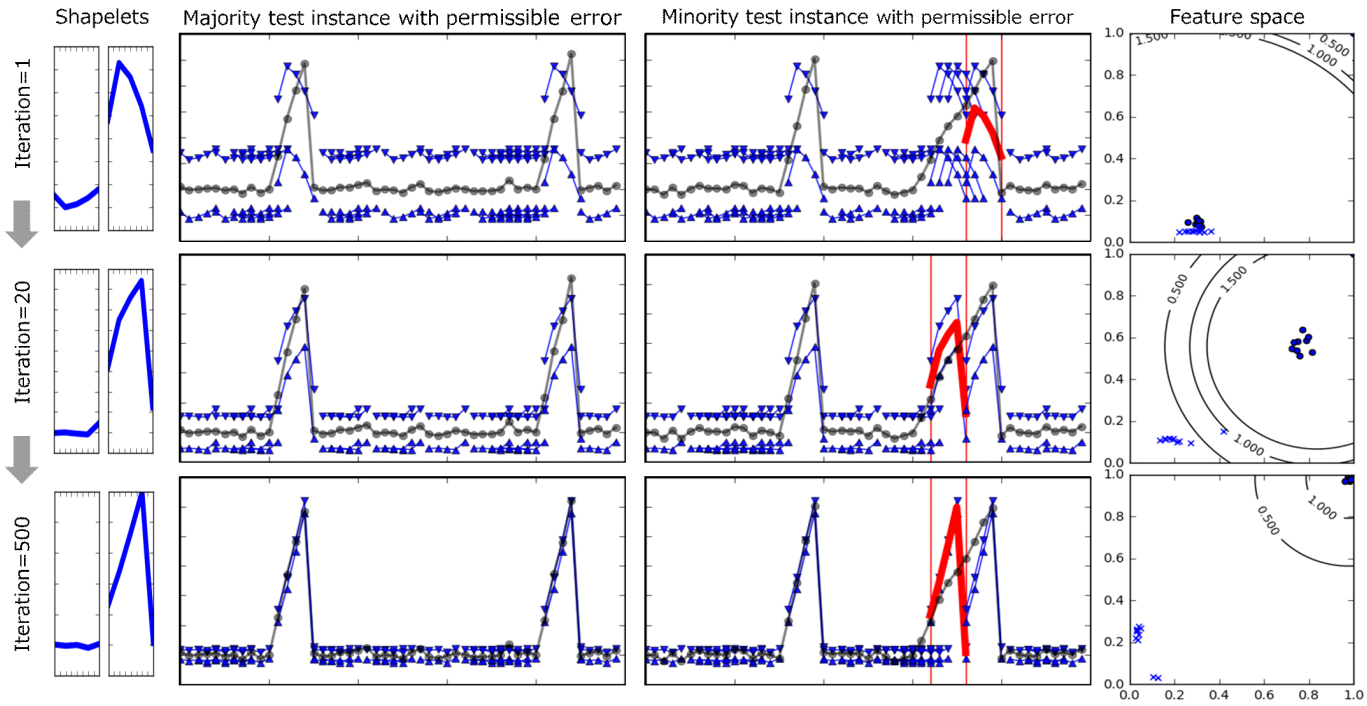

Figure 3. Illustration of the learning process. Shapelets grow so that they approximate the time series and the feature vectors are more separable.

shows test instances of majority and minority classes and the approximation by $\mathbf{S}$. For the kth shapelet $\mathbf{S}_k$, we show the maximum range (blue triangles) of the permissible errors when $\mathbf{S}$ approximates $\mathbf{T}$[1]. In addition, we show the shapelets (bold red) fitting the segments farthest away from the ranges and the position (two red vertical lines). The rightmost column shows the feature space based on OCSVM with a Radial Basis Function (RBF) kernel. The contour represents the values of (4). Test instances of majority and minority classes (circle and cross marks) are shown, respectively. We find that both the feature vectors and the maximum ranges of permissible errors are updated together as $\mathbf{S}$ grows.

In the following, we show how learning both shapelets $\mathbf{S}$ and model parameters $\mathbf{a}$ progresses with each iteration:

- At iteration 1 (top row of Fig. 3)

$\mathbf{S}$ cannot approximate $\mathbf{T}$ appropriately, and the permissible errors are large in not only the minority class but also the majority class. In the feature space, feature vectors of both classes are not separable.

- At iteration 20 (middle row of Fig. 3)

$\mathbf{S}$ is close to the horizontal and peak segments constituting $\mathbf{T}$. As a result, the maximum ranges of permissible errors have narrowed. At the same time, the feature vectors of different classes are separable in the feature space.

- At iteration 500 (bottom row of Figure3)

The algorithm approaches convergence. $\mathbf{S}$ almost completely coincides with the horizontal and peak segments, and the maximum ranges have narrowed further. Also, the feature vectors have become more separable. Learning of shapelets $\mathbf{S}$ and model parameters $\mathbf{a}$ is therefore complete.

B. Feature Vectors Based on Permissible Errors

Roughly speaking, OCLTS approximates $\mathbf{T}_i$ as if shapelets covered the whole of $\mathbf{T}_i$, as shown in Fig. 2. In the approximation, OCLTS finds the shapelets $\mathbf{S}_{k_1}, \mathbf{S}_{k_2}, \cdots, \mathbf{S}_{k_{N_i}}$ most similar to the segments of $\mathbf{T}_i$ and the positions $j_1, j_2, \cdots, j_{N_i}$ by allowing overlap while sliding the positions, where N_i is the number of slides in $\mathbf{T}_i$ (i.e., the number of the shapelets fitting to approximate $\mathbf{T}_i$). For $n = 1, 2, \cdots, N_i$, (k_n, j_n) is expressed as

$$(k_n, j_n) = \begin{cases} \underset{k=1,2,\cdots,K,\ j=0}{\arg\min} \{\mathbf{D}_{i,k,j}\} & (n = 1) \\ \underset{k=1,2,\cdots,K,\ j_{n-1}<j\le j_{n-1}+L}{\arg\min} \{\mathbf{D}_{i,k,j}\} & (1 < n < N_i) \\ \underset{k=1,2,\cdots,K,\ j=Q-L+1}{\arg\min} \{\mathbf{D}_{i,k,j}\} & (n = N_i). \end{cases} \quad (5)$$

We denote a set composed of (k_n, j_n) in (5) as

$$P_i = \{(k_n, j_n)\}_{n=1}^{N_i}. \quad (6)$$

OCLTS calculates $(k_n, j_n) \in P_i$ by incrementing n by one from $n = 1$ to $j_n = Q - L + 1$. With each increment of n, j_n can be incremented between 1 and L under $j_n \le Q - L + 1$ from j_{n-1}. N_i is given as the n that satisfies $j_n = Q - L + 1$.

[1]For the kth shapelet $\mathbf{S}_k$, we measure the maximum range as $\max_{1\le i\le I}\{\mathbf{X}_{i,k}\}$, where $\mathbf{X}_i$ is the feature vector of $\mathbf{T}_i$.

We explain how to calculate (5) by using the example in Fig. 2. Given time series $\mathbf{T}_1$ of $Q = 27$ and shapelets $\mathbf{S}$ of $(K, L) = (2, 5)$, $(k_n, j_n) \in P_1$ is calculated as $(k_1, j_1) = (2, 1)$, $(k_2, j_2) = (1, 6)$, $(k_3, j_3) = (2, 11)$, $(k_4, j_4) = (2, 13)$, $(k_5, j_5) = (1, 18)$, and $(k_6, j_6) = (2, 23)$. The calculation finishes because $j_6 = Q - L + 1$. As a result, $\mathbf{T}_1$ is approximated by six shapelets (i.e., $N_1 = 6$).

The feature vector $\mathbf{X}_i$ of $\mathbf{T}_i$ is based on the permissible errors when shapelets $\mathbf{S}$ approximate $\mathbf{T}_i$. If $\mathbf{S}_k$ is used in the approximation of $\mathbf{T}_i$ at least once, k is included in $\{k_n\}_{n=1}^{N_i}$ of (5). In this case, the kth permissible error is defined as the maximum distance between $\mathbf{S}_k$ and the segments fitted to $\mathbf{S}_k$. Otherwise, $\mathbf{S}_k$ is never used in the approximation of $\mathbf{T}_i$. This means that $\mathbf{T}_i$ can be better approximated by using other shapelets than by using $\mathbf{S}_k$. In that case, we assume that $\mathbf{S}_k$ approximates the segments where $\mathbf{S}_k$ is best-matching. As a result, the kth permissible error is substituted as the minimum distance between $\mathbf{S}_k$ and the segments of $\mathbf{T}_i$, as in the existing best-matching approach. In summary, the permissible errors in both cases are used as a feature vector and are expressed as follows:

$$\mathbf{X}_{i,k} = \begin{cases} \max_{(k,j)\in P_i} \mathbf{D}_{i,k,j} & (k \in \{k_n\}_{n=1}^{N_i}) \\ \min_{j=1,2,\cdots,J} \mathbf{D}_{i,k,j} & (k \notin \{k_n\}_{n=1}^{N_i}), \end{cases} \tag{7}$$
$$i = 1, 2, \cdots, I,\ k = 1, 2, \cdots, K,\ P_i = \{(k_n, j_n)\}_{n=1}^{N_i}.$$

In the example in Fig. 2, $\mathbf{S}_1$ and $\mathbf{S}_2$ fit into the training instance twice and four times, respectively. For this reason, the first case in (7) is used for both $k = 1$ and $k = 2$.

C. Objective Function

We formulate objective function $\mathbf{F}_{\mathbf{S},\mathbf{a}}$ to optimize both model parameters $\mathbf{a}$ and shapelets $\mathbf{S}$ simultaneously in

$$\underset{\mathbf{S},\mathbf{a}}{\text{minimize}}\, \mathbf{F}_{\mathbf{S},\mathbf{a}} \tag{8}$$
$$\mathbf{F}_{\mathbf{S},\mathbf{a}} = \frac{\lambda_1}{2} \sum_{i,i'=1}^{I} \mathbf{a}_{i'} \mathbf{a}_i \mathcal{K}(\mathbf{X}_{i'}, \mathbf{X}_i) + \frac{1}{I} \sum_{i=1}^{I} \mathbf{h}_i - \lambda_2 \sum_{i=1}^{I} \frac{\mathbf{b}_i}{N_i},$$
$$\mathbf{h}_i = \max \left\{ 0, 1 - \sum_{i'=1}^{I} \mathbf{a}_{i'} \mathcal{K}(\mathbf{X}_{i'}, \mathbf{X}_i) \right\},$$
$$\mathbf{b}_i = \sum_{(k_n, j_n)\in P_i,\ \mathbf{h}_i = 0} e^{-\gamma \mathbf{D}_{i,k_n,j_n}}$$

where λ_1, λ_2, and γ are meta-parameters. The formulation related to OCSVM is based on that of Pegasos [17], and the first and second terms of $\mathbf{F}_{\mathbf{S},\mathbf{a}}$ are the same as those in the formulation of (2). We use hinge loss $\mathbf{h}_i$ as the loss function of the SVM. The third term of $\mathbf{F}_{\mathbf{S},\mathbf{a}}$ corresponds to the optimization for $\mathbf{S}$ to get closer to the fitted time-series segments.

D. Updating Shapelets

To solve the optimization problem of (8), OCLTS needs to calculate the gradients of the objective function $\mathbf{F}_{\mathbf{S},\mathbf{a}}$ with respect to shapelets $\mathbf{S}$. By updating $\mathbf{S}$ in the negative direction of the gradients, (8) becomes closer to the optimum solution. The gradient of $\mathbf{F}_{\mathbf{S},\mathbf{a}}$ with respect to $\mathbf{S}_{k,l}$ is expressed using the chain rule of derivatives as

$$\frac{\partial \mathbf{F}_{\mathbf{S},\mathbf{a}}}{\partial \mathbf{S}_{k,l}} = \frac{\lambda_1}{2} \sum_{i,i'=1}^{I} \mathbf{a}_{i'} \mathbf{a}_i \frac{\partial \mathcal{K}(\mathbf{X}_{i'}, \mathbf{X}_i)}{\partial \mathbf{X}_i} \frac{\partial \mathbf{X}_i}{\partial \mathbf{S}_{k,l}} + \frac{1}{I} \sum_{i=1}^{I} \frac{\partial \mathbf{h}_i}{\partial \mathbf{X}_i} \frac{\partial \mathbf{X}_i}{\partial \mathbf{S}_{k,l}} - \lambda_2 \sum_{i=1}^{I} \frac{1}{N_i} \frac{\partial \mathbf{b}_i}{\partial \mathbf{S}_{k,l}}. \tag{9}$$

In the following, we derive each gradient decomposed by the chain rule in (9).

From the subdifferential of (7) and the differential of (3), we derive

$$\frac{\partial \mathbf{X}_i}{\partial \mathbf{S}_{k,l}} = \sum_{j=1}^{J} \frac{\partial \mathbf{X}_i}{\partial \mathbf{D}_{i,k,j}} \frac{\partial \mathbf{D}_{i,k,j}}{\partial \mathbf{S}_{k,l}} = \frac{2\left(\mathbf{S}_{k,l} - \mathbf{T}_{i,j'+l-1}\right)}{L}, \tag{10}$$
$$j' = \begin{cases} \underset{j=1,2,\cdots,J}{\arg\max} \{\mathbf{D}_{i,k,j} \mid (k,j) \in P_i\} & (k \in \{k_n\}_{n=1}^{N_i}) \\ \underset{j=1,2,\cdots,J}{\arg\min} \{\mathbf{D}_{i,k,j}\} & (k \notin \{k_n\}_{n=1}^{N_i}), \end{cases}$$
$$P_i = \{(k_n, j_n)\}_{n=1}^{N_i}.$$

From the subdifferential of $\mathbf{b}_i$ in (8), we derive

$$\frac{\partial \mathbf{b}_i}{\partial \mathbf{S}_{k,l}} = \begin{cases} -\frac{2\gamma}{L} \sum_{(k,j)\in P_i} (\mathbf{S}_{k,l} - \mathbf{T}_{i,j+l-1}) e^{-\gamma \mathbf{D}_{i,k,j}} & (\mathbf{h}_i = 0) \\ 0 & (\mathbf{h}_i > 0). \end{cases} \tag{11}$$

In the following derivation, we specify kernel function $\mathcal{K}$ for the RBF kernel[2]:

$$\mathcal{K}(\mathbf{X}_{i'}, \mathbf{X}_i) = e^{-\gamma ||\mathbf{X}_i - \mathbf{X}_{i'}||^2}. \tag{12}$$

From differentiating (12) with respect to $\mathbf{X}_i$, we derive

$$\frac{\partial \mathcal{K}(\mathbf{X}_{i'}, \mathbf{X}_i)}{\partial \mathbf{X}_i} = -2\gamma (\mathbf{X}_i - \mathbf{X}_{i'}) e^{-\gamma ||\mathbf{X}_i - \mathbf{X}_{i'}||^2}. \tag{13}$$

From the subdifferential of $\mathbf{h}_i$ in (8) and the differential of (12), we derive

$$\frac{\partial \mathbf{h}_i}{\partial \mathbf{X}_i} = \begin{cases} 2\gamma \sum_{i'=1}^{I} \mathbf{a}_{i'} (\mathbf{X}_i - \mathbf{X}_{i'}) e^{-\gamma ||\mathbf{X}_i - \mathbf{X}_{i'}||^2} & (\mathbf{h}_i > 0) \\ \text{zero vector of length } K & (\mathbf{h}_i = 0). \end{cases} \tag{14}$$

E. Updating One-Class Classifier

To solve the optimization problem of (8) by a gradient descent algorithm, OCLTS needs to calculate the gradients of the objective function $\mathbf{F}_{\mathbf{S},\mathbf{a}}$ with respect to model parameters $\mathbf{a}$ based on OCSVM. We apply Kernelized Pegasos [17], which is a stochastic subgradient decent algorithm for learning model parameters of the SVM by directly minimizing

[2]We can use other kernel function such as a polynomial kernel or a sigmoid kernel instead of the RBF kernel if we can derive $\partial \mathcal{K}(\mathbf{X}_{i'}, \mathbf{X}_i)/\partial \mathbf{X}_i$.

the primal problem using a kernel function to deal with the nonlinear classifier.

In the objective function of (8), the first and second terms are equal to the formulation of Pegasos (i.e., (2)), and the third term does not depend on $\mathbf{a}$. Therefore, the update of $\mathbf{a}$ is the same as in Pegasos, regardless of $\mathbf{S}$. We show how Pegasos is integrated into our algorithm in Section IV-F.

F. Algorithm

Algorithm 1 shows the algorithm of OCLTS. In line 1, $\mathbf{S}$ and $\mathbf{a}$ are initialized. Initialization of $\mathbf{S}$ generates K rough shapelets as centroids by applying the k-means++ clustering algorithm [20] to segments of length L in $\mathbf{T}$, as in [2], [3]. In lines 2–13, the stochastic subgradient descent algorithm is iterated. For $\mathbf{T}_i$ chosen at random, a feature vector $\mathbf{X}_i$ is calculated in (7) in line 4. In line 7, $\partial \mathbf{F}_{\mathbf{S},\mathbf{a}} / \partial \mathbf{S}_{k,l}$ is calculated as described in Section IV-D, and shapelets $\mathbf{S}$ are updated in the negative direction of $\partial \mathbf{F}_{\mathbf{S},\mathbf{a}} / \partial \mathbf{S}_{k,l}$. In lines 10–14, model parameters $\mathbf{a}$ are updated. From the discussion in Section IV-E, the update of $\mathbf{a}$ is the same as in the Kernelized Pegasos algorithm [17].

Algorithm 1 OCLTS

Require: Majority class training time-series instances: $\mathbf{T} \in \mathbb{R}^{I \times Q}$, Number and length of shapelets: (K, L), Number of iterations: M, Parameters: $\lambda_1, \lambda_2, \gamma$

Ensure: Shapelets: $\mathbf{S} \in \mathbb{R}^{K \times L}$, Model parameters: $\mathbf{a} \in \mathbb{R}^{I}$

1: Initialize $\mathbf{S}$ and set $\mathbf{a} = \mathbf{0}$.
2: **for** $m = 1, 2, \cdots, M$ **do**
3: Choose $i \in \{1, 2, \cdots, I\}$ uniformly at random.
4: Calculate feature vector $\mathbf{X}_i$ of $\mathbf{T}_i$ based on $\mathbf{S}$.
5: **for** $k = 1, 2, \cdots, K$ **do**
6: **for** $l = 1, 2, \cdots, L$ **do**
7: $\mathbf{S}_{k,l} \leftarrow \mathbf{S}_{k,l} - \frac{1}{2\lambda_1} \frac{\partial \mathbf{F}_{\mathbf{S},\mathbf{a}}}{\partial \mathbf{S}_{k,l}}$
8: **end for**
9: **end for**
10: **if** $\frac{1}{\lambda_1 m} \sum_{i'=1}^{I} \mathbf{a}_{i'} \mathcal{K}(\mathbf{X}_{i'}, \mathbf{X}_i) < 1$ **then**
11: $\mathbf{a}_i \leftarrow \mathbf{a}_i + 1$
12: **end if**
13: **end for**
14: $\mathbf{a} \leftarrow \frac{1}{\lambda_1 M} \mathbf{a}$
15: **return** $\mathbf{S}, \mathbf{a}$

G. Algorithmic Complexity

Changing the search-based method [1] to the learning-based method named LTS [2] reduced algorithmic complexity from $O(I^2 Q^3)$ to $O(IQ^2 M')$, where M' is the number of iterations in the gradient descent algorithm[3]. However, because LTS differentiates the minimum function after approximating it by a soft-minimum function, the algorithmic complexity includes Q^2. In addition, because LTS sequentially calculates the gradients of every training instance, the algorithmic complexity includes I.

In contrast, the algorithmic complexity of OCLTS is $O(IQM)$, despite the classifier being nonlinear. The reason for excluding Q^2 is that we calculate the subdifferential for the differential of the minimum function. The reason for including I is to learn the nonlinear classifier. In detail, $\partial \mathbf{F}_{\mathbf{S},\mathbf{a}} / \partial \mathbf{S}_{k,l}$ is calculated from the i' loop in line 10 of Algorithm 1 and from the i' loops in (9) and (14). Since $\mathbf{T}_i$ is chosen at random by the stochastic gradient descent algorithm, the algorithmic complexity does not include I^2. Although M of OCLTS may be larger than M' of LTS, we confirmed that the classification performance is sufficient with $M = 500$ for various datasets.

[3]In the same manner as [2], we omit K and L from the algorithmic complexity because they are small values.

V. Experiment Setting

This section describes the experiment setting for the evaluation in Section VI.

A. Classification Performance Metrics

We use the Area Under the Curve (AUC) and F-measure because they are appropriate metrics for imbalanced classification [6]. We performed three-fold cross-validation, where we evenly separate training data, test data, and validation data, replace them with each other, and repeat the cross-validation 10 times. We measured the metrics as the mean. The optimal meta-parameter setting of each fold is selected based on the highest AUC from the validation data.

B. Datasets

We use synthetic datasets and real-world datasets and separate each dataset into two classes.

We generate a synthetic time-series dataset consisting of (a) a horizontal segment with a value of 0 and (b) a horizontal segment with a value of 1. We add Gaussian noise with mean zero and standard deviation 0.5 to (a). We prepare two classes, namely, Class I and Class II. We add Gaussian noise with mean zero and standard deviation 0.5 to part of (b) in Class II, although we do not add noise to Class I. The length of each time-series instance is 100. Although OCLTS uses training instances of only Class I, we generate thirty instances in both classes.

We use real-world benchmark time-series datasets from the UCR Time Series Classification Archive [21]. For each dataset, we assume the class with the smallest instances as the minority class and the remaining classes as the majority class before removing training instances of the minority class at random, so that the ratio of minority to majority classes becomes close to 1:100.

C. Comparison Method

For comparison, we selected LTS [2], which is a state-of-the-art shapelet-based method, and achieves high classification performance if classes are not imbalanced [2], [4], [22].

Meta-parameter settings common to both OCLTS and LTS are the iterations M of the gradient descent algorithm and the number K and length L of shapelets. We set $M = 500$, and set L as 10% of the length of the time series. Because we know that the number of segments constituting the time-series instances is two in the case of the synthetic dataset, we set $K = 2$. In the case of other datasets, we vary $K \in \{5, 20, 80\}$ in the validation data. For the remaining meta-parameters of OCLTS, we vary $\gamma \in \{0.1, 1, 10, 100\}$ in the validation data under $\lambda_1 = 0.1$ and $\lambda_2 = 0.5$. For those of LTS, we vary $\lambda_W \in \{0.01, 0.1, 1\}$ and $\eta \in \{0.01, 0.1, 1.0\}$ in the validation data, respectively.

VI. Experimental Results

In this section, we show the effectiveness of OCLTS from the aspect of interpretability and classification performance on synthetic and real-world datasets.

A. Noisy Synthetic Data

First, we find that OCLTS learns shapelets that are robust to noise as shown in Fig. 4 (left), even if each training time-series instance has a lot of noise as shown in Fig. 4 (center). Fig. 4 (right) shows the maximum ranges (blue triangles) of permissible errors, the shapelet (bold red horizontal line) farthest away from the ranges, and the part of the time series (surrounded by two red vertical lines), respectively, on a test instance of Class II. The horizontal segment with a value close to 0 is noisy and the range is large. Nevertheless, the learned shapelet includes little noise.

Next, we find that OCLTS can discover the discriminative part (surrounded by two red vertical lines) in Fig. 4 (right). The classification performance of OCLTS is AUC 1.00 and F-measure 0.97. In contrast, the performance of LTS is AUC 0.70 and F-measure 0.67, despite the fact that sufficient and evenly distributed training instances of both classes were given. This is because it is difficult for shapelets to match the discriminative part (surrounded by two red vertical lines) and the difference is unlikely to appear in the best-matching approach. Therefore, we confirmed the case where the feature vectors based on the permissible errors approach are effective despite a balanced class distribution.

B. Imbalanced Classification on Real-World Datasets

Table I shows the imbalanced classification performance on test data in the benchmark datasets. Although the optimal meta-parameter settings were selected and changed in each fold of cross-validation, we show the meta-parameter settings for the best AUC in the validation data. We applied SMOTE [23], which is a well-known over-sampling method, to training data of LTS in order to reduce the sparseness of the minority class at the data level. The superiority of OCLTS, which uses training data from only the majority class, can be seen in Table I.

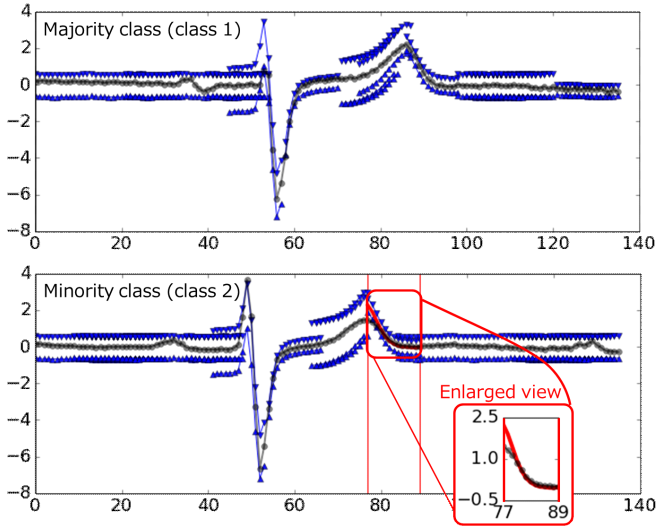

Figure 5. Discovery of a t-wave, which is the only medically significant difference in the ECG data. OCLTS can discover it by one-class learning.

C. Interpretability Using ECG Dataset

ECG Five Days is a real-world electrocardiography (ECG) dataset in the UCR time series. It has linear drift (in a medical context, this is called a wandering baseline), and the time series from the two different classes are very similar, at least globally. We learned the shapelets by OCLTS using training data of Class 1. Figure 5 (top) shows a test time-series instance in class 1 (i.e., majority class) and the maximum ranges (blue triangles) of permissible errors. Figure 5 (bottom) shows a test time-series instance in class 2 (i.e., minority class), the maximum ranges (blue triangles) of permissible errors, the shapelet (bold red) farthest away from the ranges, and the part (surrounded by two red vertical lines), respectively. This discriminative part is known as a t-wave, which is the only medically significant difference between the two classes. Although the existing shapelet-based method can discover this part [11] under supervised binary classification, OCLTS can also discover the same part despite using a one-class classification.

VII. Conclusion

We proposed OCLTS for learning shapelets and a classifier based on OCSVM from training data of only one class. Algorithmic complexity is reduced to a greater extent by OCLTS than by LTS [2] by using a stochastic subgradient descent algorithm, despite OCLTS learning using a nonlinear classifier. Using synthetic and real-world datasets, we confirmed that OCLTS learns shapelets robust

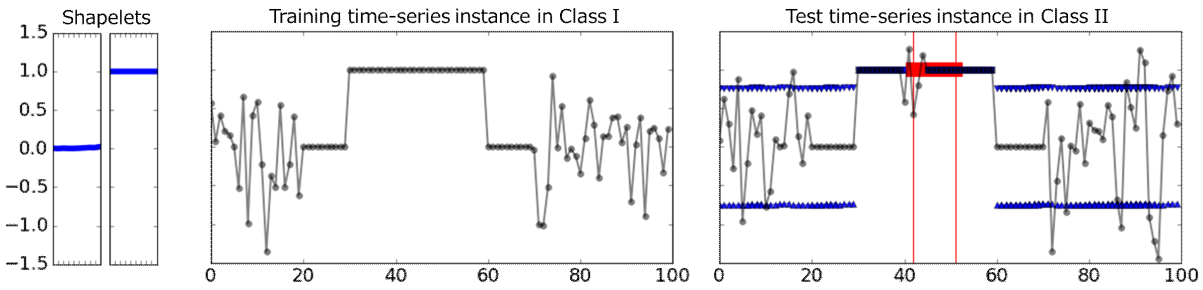

Figure 4. Discovery of shapelets in noisy synthetic data. OCLTS can discover the shapelets robust to noise.

Table I
COMPARING PERFORMANCE FOR IMBALANCED BINARY CLASSIFICATION

Dataset	OCLTS				LTS				
	AUC	F-measure	K	γ	AUC	F-measure	K	λ_W	η
ECG5000	**0.99**	**1.00**	80	100	0.98	0.99	5	1	1
ECGFiveDays	**0.98**	**0.90**	80	10	0.55	0.71	5	0.1	1
DiatomSizeReduction	**1.00**	**1.00**	5	100	0.92	0.97	5	0.1	0.1
DistalPhalanxOutlineCorrect	**0.78**	**0.82**	80	10	0.77	0.73	5	0.01	1
MiddlePhalanxOutlineAgeGroup	0.93	0.95	80	100	0.93	0.95	5	1	1
MiddlePhalanxTW	**0.95**	0.96	80	1	0.92	0.96	80	0.01	1
ProximalPhalanxOutlineAgeGroup	**0.96**	**0.96**	80	100	0.93	0.66	5	1	1
ProximalPhalanxOutlineCorrect	0.73	**0.81**	80	1	**0.85**	0.49	20	1	0.1
Haptics	**0.84**	0.89	5	100	0.77	0.89	20	1	1
Earthquakes	**0.82**	**0.91**	80	100	0.76	0.89	5	0.01	0.1
Lighting2	**0.79**	**0.74**	5	100	0.77	0.69	5	0.01	1

to noise and achieves performance superior to LTS in imbalanced binary classification. For the ECG Five Days dataset, OCLTS discovered discriminative shapelets that the supervised shapelet-based method [11] also discovered using both class labels during training, despite OCLTS using only one-class learning.

REFERENCES

[1] L. Ye and E. Keogh, "Time Series Shapelets: A New Primitive for Data Mining," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '09. ACM, 2009, pp. 947–956.

[2] J. Grabocka, N. Schilling, M. Wistuba, and L. Schmidt-Thieme, "Learning Time-series Shapelets," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '14. ACM, 2014, pp. 392–401.

[3] Q. Zhang, J. Wu, H. Yang, Y. Tian, and C. Zhang, "Unsupervised Feature Learning from Time Series," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, pp. 2322–2328.

[4] L. Hou, J. T. Kwok, and J. M. Zurada, "Efficient learning of timeseries shapelets," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI'16. AAAI Press, 2016, pp. 1209–1215.

[5] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special Issue on Learning from Imbalanced Data Sets," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 1–6, Jun. 2004.

[6] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Trans. on Knowl. and Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.

[7] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the Support of a High-Dimensional Distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[8] Y. Akihiro and N. Takeichiro, "One-Class Learning Time-Series Shapelets," in *IEEE International Conference on Big Data*, ser. Big Data '13. IEEE Computer Society, 2018, pp. 2365–2372.

[9] A. Mueen, E. Keogh, and N. Young, "Logical-shapelets: An Expressive Primitive for Time Series Classification," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '11. ACM, 2011, pp. 1154–1162.

[10] K.-W. Chang, B. Deka, W. mei W. Hwu, and D. Roth, "Efficient Pattern-Based Time Series Classification on GPU." in *Proceedings of the Fifth International Conference on Data Mining*, ser. ICDM '05. IEEE Computer Society, 2012, pp. 131–140.

[11] E. J. Keogh and T. Rakthanmanon, "Fast Shapelets: A Scalable Algorithm for Discovering Time Series Shapelets," in *Proceedings of the 13th SIAM International Conference on Data Mining*, 2013, pp. 668–676.

[12] J. Zakaria, A. Mueen, and E. Keogh, "Clustering Time Series Using Unsupervised-Shapelets," in *Proceedings of the 12th International Conference on Data Mining*, ser. ICDM '12. IEEE Computer Society, 2012, pp. 785–794.

[13] Q. Zhang, J. Wu, P. Zhang, G. Long, and C. Zhang, "Salient Subsequence Learning for Time Series Clustering," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, june 2018.

[14] S. Roychoudhury, M. F. Ghalwash, and Z. Obradovic, "Cost Sensitive Time-Series Classification," in *ECML/PKDD*, ser. Lecture Notes in Computer Science, vol. 10535. Springer, 2017, pp. 495–511.

[15] B. Raskutti and A. Kowalczyk, "Extreme Re-balancing for SVMs: A Case Study," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 60–69, Jun. 2004.

[16] Y. Song, Z. Wen, C.-Y. Lin, and R. Davis, "One-class Conditional Random Fields for Sequential Anomaly Detection," in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, ser. IJCAI '13. AAAI Press, 2013, pp. 1685–1691.

[17] S. Shalev-Shwartz, Y. Singer, N. Srebro, and A. Cotter, "Pegasos: Primal Estimated Sub-gradient Solver for SVM," *Math. Program.*, vol. 127, no. 1, pp. 3–30, Mar. 2011.

[18] C. LEE, "Pegasos Algorithm for One-Class Support Vector Machine," *IEICE Transactions on Information and Systems*, vol. E96.D, no. 5, pp. 1223–1226, 2013.

[19] G. S. Kimeldorf and G. Wahba, "Some Results on Tchebycheffian Spline Functions," *Journal of Mathematical Analysis and Applications*, vol. 33, no. 1, pp. 82–95, 1971.

[20] D. Arthur and S. Vassilvitskii, "K-means++: The Advantages of Careful Seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[21] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, "The UCR Time Series Classification Archive," July 2015, www.cs.ucr.edu/~eamonn/time_series_data/.

[22] X. Li and J. Lin, "Evolving Separating References for Time Series Classification," in *SDM*. SIAM, May 2018, pp. 243–251.

[23] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Int. Res.*, vol. 16, no. 1, pp. 321–357, Jun. 2002.