Cost Sensitive Time-Series Classification

Shoumik Roychoudhury[1], Mohamed Ghalwash[1,2,3], and Zoran Obradovic[1(✉)]

[1] Center for Data Analytics and Biomedical Informatics, Temple University,
Philadelphia, PA, USA
{shoumik.rc,mohamed.ghalwash,zoran.obradovic}@temple.edu
[2] IBM T. J. Watson Research Center, Cambridge, MA, USA
[3] Faculty of Science, Ain Shams University, Cairo, Egypt

Abstract. This paper investigates the problem of highly imbalanced time-series classification using shapelets, short patterns that best characterize the target time-series, which are highly discriminative. The current state-of-the-art approach learns generalized shapelets along with weights of the classification hyperplane via a classical cost-insensitive loss function. Cost-insensitive loss functions tend to treat different misclassification errors equally and thus, models are usually biased towards examples of majority class. The rare class (which will be referred to as positive class) is usually the important class and a false negative is always costlier than a false positive. Traditional 0–1 loss functions fail to differentiate between these two types of misclassification errors. In this paper, the generalized shapelets learning framework is extended and a cost-sensitive learning model is proposed. Instead of incorporating the misclassification cost as a prior knowledge, as was done by other published methods, we formulate a constrained optimization problem to *learn* the unknown misclassification costs along with the shapelets and their weights. First, we demonstrated the effectiveness of the proposed method on two case studies, with the objective to detect true alarms from life threatening cardiac arrhythmia dataset from Physionets MIMIC II repository. The results show improved true alarm detection rates over the current state-of-the-art method. Next, we compared to the state-of-the-art learning shapelet method on 16 balanced dataset from UCR time-series repository. The results show evidence that the proposed method outperforms the state-of-the-art method. Finally, we performed extensive experiments across additional 18 imbalanced time-series datasets. The results provide evidence that the proposed method achieves comparable results with the state-of-the-art sampling/non-sampling based approaches for highly imbalanced time-series datasets. However, our method is highly interpretable which is an advantage over many other methods.

Keywords: Cost sensitive · Time-series classification · Shapelets

1 Introduction

Research on time-series classification has garnered importance among practitioners in the data mining community. A major reason behind the ever increasing

M. Ceci et al. (Eds.): ECML PKDD 2017, Part II, LNAI 10535, pp. 495–511, 2017.
https://doi.org/10.1007/978-3-319-71246-8_30

interest among data-miners is the plethora of time-series data available from a wide range of real-life domains. Temporal ordered data from areas such as financial forecasting, medical diagnosis, weather prediction etc. provide classification challenges more akin to real-world scenarios. Thus, building more robust time-series classification models is imperative.

One of the key sources of performance degradation in the field of time-series classification is the class imbalance problem [18] where the minority class (we call it the positive class) is outnumbered by abundant negative class instances. Models built using standard classification algorithms on such imbalanced datasets, which generally have minimum classification error as a criterion for classifier design often, are biased towards the majority class; and therefore, have higher misclassification error for the minority class examples. Moreover, in real-world scenarios such as object detection, medical diagnosis etc., the positive class is usually the more important class and false negatives are always costlier than false positives. Traditional 0–1 loss function classifiers fail to differentiate between these two types of errors and final outcomes are naturally biased towards the abundant negative class. Thus, a cost-sensitive classifier is preferred when dealing with datasets where examples from different classes carry different misclassification costs.

Recently, in the realm of time-series classification, Grabocka et al. [10] proposed a novel framework known as Learning Time-series Shapelets (LTS) to directly learn generalized short time-series subsequences known as shapelets [23] along with weights of a classifier hyperplane to differentiate temporal instances in a binary classification framework. Shapelets are local discriminative patterns (or subsequences) that can be used to characterize the target class, for determining the time-series class membership. Shapelets have been proven to have high predictive powers as they provide local variation information within the time-series as well as high interpretability of predictions due to easier visualizations. LTS formulates an optimization problem where a cost-insensitive 0–1 logistic loss function is minimized in order to learn generalized shapelets. The minimum Euclidean distances of the learned shapelets to the time-series can be used to linearly separate the time-series examples from different classes.

However, LTS uses cost-insensitive loss function that treats false positive and false negative errors equally, which limits its applicability on balanced datasets. In this paper, we propose a cost-sensitive time-series classification framework (henceforth known as CS-LTS) by extending the LTS model. A cost-sensitive logistic loss function is minimized to enhance the modeling capability of LTS. The cost-sensitive logistic loss function uses variable misclassification costs for false positive and false negative errors. Generally, these misclassification cost values are available from the cost matrix provided by domain experts which is often a cumbersome procedure. Instead of using fixed cost parameters, this paper *learns* the variable misclassification costs from the training data via a constrained optimization problem. Thus, the main contribution of this paper is summarized as the following.

1. The proposed method learns the misclassification costs from the training data thus nullifying the need for predetermination of cost values for misclassification errors. To the best of our knowledge, the proposed model is the first algorithmic approach to solve highly imbalanced time-series classification problem.
2. A constrained optimization problem is proposed which jointly learns shapelets (highly interpretable patterns), their weights, and most importantly misclassification costs, while other cost-sensitive approaches mainly consider misclassification costs are given a priori.
3. The effectiveness of the method is demonstrated on life-threatening cardiac arrhythmia dataset from Physionets MIMIC II repository showing improved true alarm detection rates over the current state-of-the-art method for false alarm suppression.
4. Finally, the method is evaluated extensively on 34 real-world time series datasets with varied degree of imbalances and compared to a large set of baseline methods previously proposed in the realm of imbalance time-series classification problems.

In Fig. 1(a), we show all time series examples for the blue and red classes. The blue class has only 3 time series, while the red class has 10 time series. Since LTS does not handle imbalance dataset, the learned hyperplane is very biased. This is clear from Fig. 1(b) that shows the distance between the two learned shapelets using LTS and the training time series. CS-LTS learns a hyperplane that is aware about the imbalance in the data, as shown in Fig. 1(c).

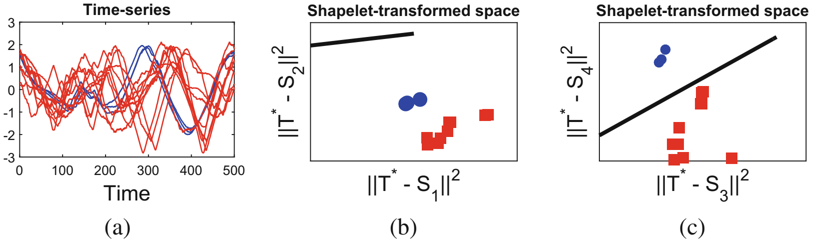

Fig. 1. An illustration of the proposed CS-LTS model (c) compared to LTS (b) using 2 shapelets learned on an imbalanced version of BirdChicken dataset (a). (Color figure online)

Next, we present a short literature review for time-series classification using shapelets and cost-sensitive time-series classification.

2 Related Work

Time-Series Classification via Shapelets. In the field of time-series classification, the concept of shapelets have received a lot of attention [8,10,16,23,24].

Shapelets are local discriminative patterns (or subsequences) that characterize the target class and maximally discriminate instances of time-series from various classes. Discovering the most discriminative subsequences is crucial for the success of time-series classification using shapelets. The primary approach, based on search-based techniques, proposed by Ye and Keogh [23], exhaustively search for all possible subsequences and a decision tree was constructed based on information gain criterion. The information gain accuracy was ranked based on the minimum distance of the candidate subsequences to the entire time-series training set. Hills et al. [15] perceived this minimum distance of the set of shapelets to a time-series dataset as a data transformation to a shapelet-transformed space where standard classifiers could be used to achieve high classification accuracy using the shapelet-transformed data as predictors. Recently, Grabocka et al. [10] proposed a novel framework known as Learning Time-series Shapelets (LTS) to jointly learn generalized shapelets along with weights of a logistic regression model using the minimum Euclidean distances of shapelets to time-series dataset as predictors. The method discovered optimal shapelets and reported statistically significant improvements in accuracy compared to other shapelet-based time-series classification models. However, a major drawback is low true positive rate in case of highly imbalanced time-series datasets. The logistic loss used in the LTS framework is a cost-insensitive loss function which treats false positive and false negative misclassifications errors equally. Classification models built using such loss functions suffer from the class imbalance problem.

Cost-Sensitive Classification. Classification techniques for handling imbalanced data-sets can broadly be divided into two kinds of approaches, data-level approaches [2–5,12,13,17] and algorithmic-level [22] approaches. Data-level methods are sampling techniques that act as a pre-processing steps prior to the learning algorithm to balance the imbalanced datasets either through oversampling of the minority class or under sampling of the majority class or combination of both. Algorithmic-level approaches directly manipulate the learning algorithm by incorporating a predefined misclassification cost for each class to the loss function. These methods have reported excellent performance with good theoretical guarantees [14]; however, predetermination of optimal class misclassification cost or data-space weighting is required which can vary on a case-by-case basis among different datasets and also require domain expertise.

In this paper, an algorithmic approach is followed to directly manipulate the learning procedure by minimizing a cost-sensitive logistic loss function. An additive asymmetric learning function is fitted to the training data. In addition to learning the shapelets and weight parameters of the classification hyperplane, the cost parameters are also estimated from the training data. A constrained optimization problem is formulated that is optimized to jointly learn shapelets, weights of the classification hyperplane and misclassification cost parameters nullifying the need for predetermination of cost values for misclassification errors.

3 Model Description

Preliminaries: A binary class time-series dataset composed of I training examples denoted as $\mathbf{T} \in \mathbb{R}^{I \times Q}$ is considered where, each T_i $(1 \leq i \leq I)$ is of length Q and the label for each time-series instance is a nominal variable $Y \in \{0,1\}^I$. Candidate shapelets are segments of length L from a time-series starting from j-th time point inside the i^{th} time-series. The objective is to learn k shapelets $\mathbf{S}$, each of length L, that are most discriminative in order to characterize the target class. The shapelets are denoted as $S \in \mathbb{R}^{K \times L}$.

The minimum distance $M_{i,k}$ between the i^{th} series T_i and the k^{th} shapelet S_k is the distance between the segment and time-series. This is defined as

$$M_{i,k} = \min_{j=1,\ldots,J} \frac{1}{L} \sum_{l=1}^{L} (T_{i,j+l-1} - S_{k,l})^2 \tag{1}$$

Given a set of I time-series training examples and K shapelets, a shapelet-transformed matrix [15] $\mathbf{M} \in \mathbb{R}^{I \times K}$ can be constructed which is composed of minimum distances $M_{i,k}$ between the i^{th} series T_i and the k^{th} shapelet S_k. The minimum distance M matrix is a representation in the shapelet transformed space and acts as predictors for each target time-series. However, the function in Eq. (3) is not continuous and thus non-differentiable. Grabocka et al. [10] defined a soft-minimum function (shown in Eq. (2)), which is an approximation for $M_{i,k}$.

$$M_{i,k} \approx \hat{M}_{i,k} = \frac{\sum_{j=1}^{J} D_{i,k,j} \exp(\alpha D_{i,k,j})}{\sum_{\bar{j}=1}^{J} \exp(\alpha D_{i,k,\bar{j}})} \tag{2}$$

where $D_{i,k,j}$ is defined as the distance between the j^{th} segment of series i and the k^{th} shapelet given by the formula

$$D_{i,k,j} = \frac{1}{L} \sum_{l=1}^{L} (T_{i,j+l-1} - S_{k,l})^2 \tag{3}$$

Learning Model: A linear learning model (shown in Eq. (4)) was proposed by [10] using the minimum distances M as predictors in the transformed shapelet space.

$$\hat{Y}_i = W_0 + \sum_{k=1}^{K} M_{i,k} W_k \quad \forall i \in \{1, \ldots, I\} \tag{4}$$

The learning function (Eq. (4)) is extended by incorporating C_{FN} and C_{FP} for false negative and false positive misclassifications cost respectively. The new asymmetric learning model is defined as Eq. (5).

$$Z_i = \frac{1}{C_{FN} + C_{FP}} ln \frac{\sigma(\hat{Y}) C_{FN}}{1 - \sigma(\hat{Y}) C_{FP}} = \frac{1}{C_{FN} + C_{FP}} (\hat{Y} + ln \frac{C_{FN}}{C_{FP}}) \tag{5}$$

$\sigma()$ is the logistic function and $\sigma(\hat{Y})$ represents the posterior probability of $P(Y = 1 \,|\, X)$.

Additionally, a cost-sensitive loss function (Eq. (6)) is proposed which is a differential cost-weighted logistic loss between the actual targets Y and the estimated targets Z.

$$\mathcal{L}(Y, Z) = -Y ln\sigma(C_{FN}Z) - (1 - Y) ln(1 - \sigma(C_{FP}Z)) \tag{6}$$

A regularized cost-sensitive logistic loss function defined by Eq. (7) is the regularized objective function denoted by $\mathcal{F}$.

$$\operatorname*{argmin}_{S,W,C} \mathcal{F}(S, W, C) = \operatorname*{argmin}_{S,W,C} \sum_{i=1}^{I} \mathcal{L}(Y_i, Z_i) + \lambda_W \|W\|^2 \tag{7}$$

where $C \in \{C_{FN}, C_{FP}\}$. The problem is formulated as a constrained optimization problem since the misclassification costs should always be positive. The misclassification cost denotes the loss incurred when a wrong prediction occurs. The constraints ensure both costs are positive and also the fact that cost of false negative is at least θ times greater than cost of false positive. These conditions ensure the loss function to be penalized more in the event of an error in the positive class than an error in the negative class.

$$\begin{aligned} &\operatorname*{argmin}_{S,W,C} \mathcal{F}(S, W, C) \\ \text{subject to } &C_{FN} > 0,\ C_{FP} > 0 \\ &C_{FN} > \theta C_{FP} \end{aligned} \tag{8}$$

Similar to [10], a Stochastic gradient descent (henceforth SGD) approach is adopted to solve the optimization problem. The SGD algorithm optimizes the parameters to minimize the loss function by updating through per instance of the training data. Thus, the per-instance decomposed objective function $\mathcal{F}_i$ (denoted by Eq. (9)) shows the division of Eq. (7) into per-instance losses for each time-series.

$$\mathcal{F}_i = \mathcal{L}(Y_i, Z_i) + \frac{\lambda_W}{I} \sum_{k=1}^{K} W_k^2 \tag{9}$$

The objective of the learning algorithm is to learn the optimal shapelet S_k, the weights W for the hyperplane and the misclassification costs C which minimizes the loss function (Eq. (7)).

The SGD algorithm requires definitions of gradients of the objective function with respect to shapelets, hyperplane weights and misclassification costs. Eq. (10) shows the point gradient of objective function for the i^{th} time-series with respect to shapelet S_k.

$$\frac{\partial \mathcal{F}_i}{\partial S_{k,l}} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \hat{Y}_i} \frac{\partial \hat{Y}_i}{\partial \hat{M}_{i,k}} \sum_{j=1}^{J} \frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}} \frac{\partial D_{i,k,j}}{\partial S_{k,l}} \tag{10}$$

Furthermore, the gradient of the cost-sensitive loss function with respect to the learning function Z_i is defined in Eq. (11). Also the gradient of the cost-sensitive learning function with respect to the estimated target variable $\hat{Y}_i$ is shown in Eq. (12)

$$\frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} = (1 - Y_i)\sigma(C_{FP}Z_i)C_{FP} - Y_i(1 - \sigma(C_{FN}Z_i))C_{FN} \tag{11}$$

$$\frac{\partial Z_i}{\partial \hat{Y}_i} = \frac{1}{C_{FN} + C_{FP}} \tag{12}$$

Equation (13) shows the gradient of the estimated target variable with respect to the minimum distance. The gradient of the over all minimum distance with respect to the segment distance and the gradient of the segment distance with respect to a shapelet point is defined by Eqs. (14) and (15) respectively.

$$\frac{\partial \hat{Y}_i}{\partial \hat{M}_{i,k}} = W_k \tag{13}$$

$$\frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}} = \frac{\exp(\alpha D_{i,k,j}(1 + \alpha(D_{i,k,j} - \hat{M}_{i,k}))}{\sum_{\bar{j}=1}^{J} \exp\left(\alpha D_{i,k,\bar{j}}\right)} \tag{14}$$

$$\frac{\partial D_{i,k,j}}{\partial S_{k,l}} = \frac{2}{L}(S_{k,l} - T_{i,j+l-1}) \tag{15}$$

The hyperplane weights W are learned by minimizing the objective function 7 via SGD. The gradients for updating the weights W_k is shown in Eqs. (16) and (17) shows the gradient for update of the bias term W_0.

$$\frac{\partial \mathcal{F}_i}{\partial W_k} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \hat{Y}_i} \hat{M}_{i,k} + \frac{2\lambda_W}{I} W_k \tag{16}$$

$$\frac{\partial \mathcal{F}_i}{\partial W_0} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \hat{Y}_i} \tag{17}$$

The learning procedure for estimating the misclassification cost values in the proposed framework is a constrained optimization problem because we need to guarantee that $C_{FN} > 0$, $C_{FP} > 0$ and $C_{FN} > \theta C_{FP}$, where $\theta \in \mathbb{Z}$. However, Stochastic Gradient Descent algorithm can only be applied to solve unconstrained optimization problems. Thus, we convert the constrained optimization into an unconstrained optimization similar to [19] and apply SGD algorithm to solve the optimization problem for learning the optimal misclassification costs.

$$C_{FN} = \theta C_{FP} + \mathcal{D} \tag{18}$$

The false negative misclassification cost (C_{FN}) is first written in terms of false positive misclassification cost as shown in Eq. (18) and replaced in Eq. (6) changing the optimization problem to Eq. (19).

Algorithm 1. Cost-sensitive learning time-series shapelets

```
procedure CS-LTS
Input: $T \in \mathcal{R}^{I \times Q}$, Number of shapelets $K$, length of a shapelet $L$, Regularization parameter $\lambda_W$, Learning rate $\eta$, maxIter
Initialize: Shapelets $S \in \mathbb{R}^{K \times L}$, classification hyperplane weights $W \in \mathbb{R}^K$, Bias $W_0 \in \mathbb{R}$, Misclassification cost $C_{FP} \in \mathbb{R}$, $\theta \in \mathbb{Z}$, $\mathcal{D} \in \mathbb{R}$
    for iterations = $\mathbb{N}_1^{maxIter}$ do
        for $i = 1, ..., I$ do
            for $k = 1, ..., K$ do
                $W_k^{new} \leftarrow W_k^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial W_k}$
                for $l = 1, ..., L$ do
                    $S_{k,l}^{new} \leftarrow S_{k,l}^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial S_{k,l}}$
            $W_0^{new} \leftarrow W_0^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial W_0}$
            $\log C_{FP}^{new} \leftarrow \log C_{FP}^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial \log C_{FP}}$
            $\mathcal{D}^{new} \leftarrow \mathcal{D}^{old} - \eta \frac{\partial \mathcal{F}_i}{\partial \mathcal{D}}$
    Return $S, W, W_0, C_{FP}$
```

$$\begin{aligned} &\underset{S,W,C_{FP},\mathcal{D}}{\text{argmin}} \;\; \mathcal{F}(S, W, C_{FP}, \mathcal{D}) \\ &\text{subject to } C_{FP} > 0 \end{aligned} \tag{19}$$

$\mathcal{D}$ is a regularization term for the misclassification cost. The objective function is then minimized with respect to $\log C_{FP}$ instead of C_{FP}. As a result, the new optimization problem becomes unconstrained. Derivatives of objective function with respect to $\log C_{FP}$ and $\mathcal{D}$ in gradient descent are computed as:

$$\frac{\partial \mathcal{F}_i}{\partial \log c_{FP}} = c_{FP} \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial c_{FP}} \tag{20}$$

$$\frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial c_{FP}} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial c_{FP}} \tag{21}$$

$$\frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial \mathcal{D}} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial \mathcal{D}} \tag{22}$$

The steps of the proposed cost-sensitive time-series classification method (CS-LTS, henceforth) are shown in Algorithm 1. The pseudocode shows that the procedure updates all K shapelets and the weights W, W_0, false positive cost C_{FP} and parameter $\mathcal{D}$ by a learning rate η.

4 Experimental Evaluation

In this section, we evaluate the effectiveness of the proposed method on different setting represented by different datasets. The objective function in Eq. (7) is a non-convex function with respect to parameters and solving it via SGD requires a good initialization of the parameters. The initialization step is very important in this scenario as it influences whether the optimization reaches the region of global minimum.

Model Parameter Initializations: Shapelets were initialized using K-means centroids of all segments similar to [10]. First we set the minimum length (L_{min}) of a shapelet to be 10% of the length of the time-series examples. Then the total number of shapelets was computed as L_{min} multiplied by number of training time series. The number of shapelets used as input for the optimization function was determined using $K = log(total\ number\ of\ segments)$. Three scales $\{L_{min}, 2 \times L_{min}, 3 \times L_{min}\}$ of subsequence lengths were investigated.

The weight parameters W_k and W_0 were initialized randomly around 0. C_{FP} was initially set to 1. The values for θ and initial value of $\mathcal{D}$ were determined through a grid search approach using internal cross-validations over the training data. The values for θ were searched from the set $\{1, 5, 10, 25, 50, 100\}$ and the initial values for $\mathcal{D}$ was chosen from $\{0.001, 0.01, 0.1, 10, 100, 1000\}$. The best parameter value was identified via internal cross-validation on training data. Once the best parameter value was identified, the methods were trained on the entire training set using the best chosen parameters, and the learned model was tested on the test set which was completely separate from the training procedure. The learning rate η was initialized to a small value of 0.01. The $maxIter$ for the optimization was set to 5000 iterations.

Evaluation Measures: We report F_β score for $\beta \in \{1, 2, 3\}$ since this is a commonly used performance metric for imbalanced learning. These are simple functions of the precision and recall. The traditional F-score or F_1 score is the harmonic mean of precision and recall that is considered a balanced measure between precision and recall. For $\beta > 1$ the evaluation metric rewards higher true positive rates. We also consider the sensitivity and specificity evaluation metrics, as the objective is to achieve lower false negative with minimum increase in false positive rates.

4.1 Cost Sensitive Cardiac Arrhythmia Alarms Detection

In this set of experiments, we demonstrate the effectiveness of the proposed method on two cost-sensitive applications from PhysioNets MIMIC II version 3 repository [9,21]. The objective is to detect true alarms while suppressing false alarms, where missing true alarms (positive class) is more severe than missing false alarms (negative class), since missing true alarm could lead to serious consequences and risk patients' lives.

The database is a multi-parameter ICU repository containing patients' records of up to eight signals from bedside monitors in Intensive Care Units (ICU). The extracted datasets contain human-annotated true and false cardiac arrhythmia alarms. We extracted a subset of patients' records that contained signal from lead ECG II, because it was identified as the sensor that contained the least number of missing values across the patients. For each alarm event, a 20-s window prior to the alarm event was extracted similar to [20].

We partition the dataset into four distinct cross-validation datasets, where we train the model on 3 folds and test on the fourth one. In addition to the cross validation experiment, we repeat the entire process of cross-validation for 10 independent trials (each trial has 4 distinct partitions on true alarm instances) which results in 40 different combination of training data. The mean and standard deviation of the evaluation metrics is then reported.

The two datasets selected are VTACH and CHALLENGE. VTACH consists of true and false Ventricular Tachycardia alarms from the ICU patients. CHALLENGE dataset is a mixture of different true and false arrhythmia alarms. The alarms categories are Asystole, Extreme Bradycardia, Extreme Tachycardia, Ventricular Tachycardia and Ventricular Flutter/Fibrillation. This dataset was presented at a competition in 2015 organized by PhysioNet to encourage the development of algorithms to reduce the incidence of false alarms in the Intensive Care Unit (ICU).

Achieving high true alarm detection rate (TAD) or high sensitivity is important when suppressing high false alarm rates from bedside monitors in ICU. High false alarm rates cause desensitization among care providers, thus risking patients' lives [7]. The objective of the prediction task is to provide high false alarms suppression (FAS) rates (achieve high specificity) while keeping TAD (sensitivity) high. In the two datasets, (Fig. 2) CS-LTS (circle) achieves higher TAD (Y-axis) than LTS (diamond) and the current state-of-the-art baseline BEHAR [1] (star) in the field of critical alarm detection. FAS (X-axis) is better for LTS (diamond) on both datasets compared to CS-LTS (circle). However, improving TAD by decreasing FAS is acceptable as missing true alarms may result in patient fatality. CS-LTS (circle) beats BEHAR (star) in terms of true alarm detection rate on both the datasets. In terms of false alarm suppression, CS-LTS achieves comparable performance on VTACH dataset. BEHAR (star) achieves 100% FAS for CHALLENGE dataset, however, true alarm detection rate is 0%. Figure 3 shows the comparison of F_β scores for VTACH and CHALLENGE datasets. In both datasets CS-LTS outperforms LTS with respect to $\beta = 2$ and $\beta = 3$. This proves that CS-LTS improves the TAD score on both datasets when compared to LTS.

4.2 Balanced Time Series Datasets

In this set of experiments, we highlight that the proposed model attains comparable or better classification accuracy when compared to state-of-the-art LTS on balanced datasets. So, incorporating cost sensitive learning does not hurt the optimization algorithm because it automatically learns the cost sensitive parameters. This is very useful if the intrinsic sensitivity of the data is not known a priori.

Sixteen binary-class datasets were selected from UCR time-series repository [6]. In order to ensure fair comparison with LTS, the default train and test splits were used. Ten independent runs (with different initialization for both

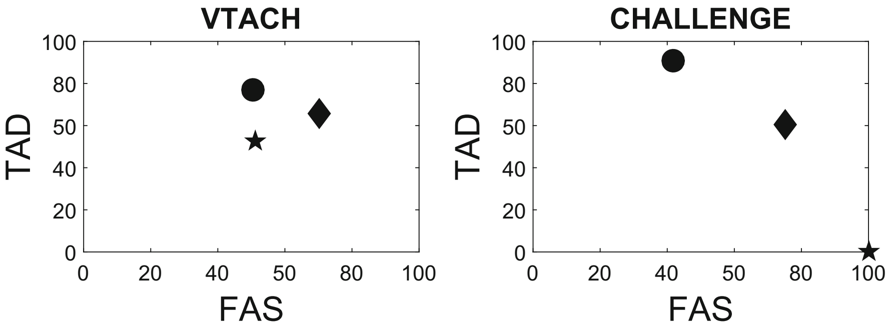

Fig. 2. CS-LTS[●] vs. LTS[♦] vs. BEHAR[★] in terms of true alarm detection (TAD) and false alarm suppression (FAS) rates over 2 critical alarm datasets. CS-LTS achieves higher TAD on both datasets compared to LTS and BEHAR.

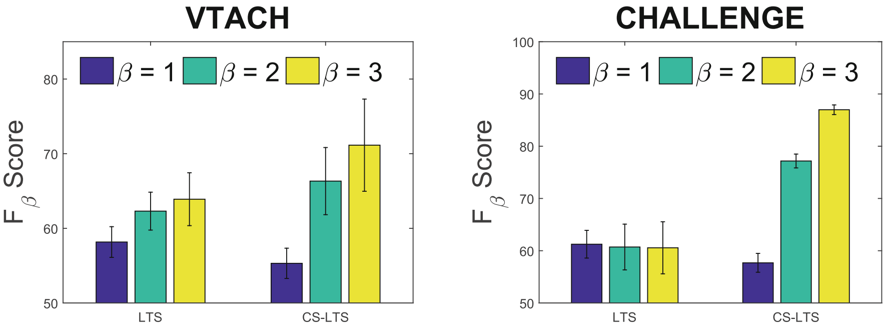

Fig. 3. Comparison of CS-LTS vs. LTS in terms F_1, F_2 and F_3 scores over 2 false alarm suppression datasets.

LTS and CS-LTS) were conducted and the average and standard deviation of the evaluation metric are reported.

The results of comparing CS-LTS to LTS on the 16 datasets are shown in Fig. 4. It is observed that CS-LTS outperforms or comparable to LTS on all 16 datasets. This set of experiments highlights the fact that the CS-LTS model provides a good alternative to LTS as it can handle balanced datasets quite effectively. The proposed method attains higher sensitivity with little loss of specificity when compared to LTS.

4.3 Imbalanced Time Series Datasets

In order to highlight the advantage of cost-sensitive learning over cost-insensitive learning, in this set of experiments, we extensively evaluate the model on 18 highly imbalanced datasets and compare it with LTS and different over-sampling and under-sampling methods. The imbalanced time series datasets were

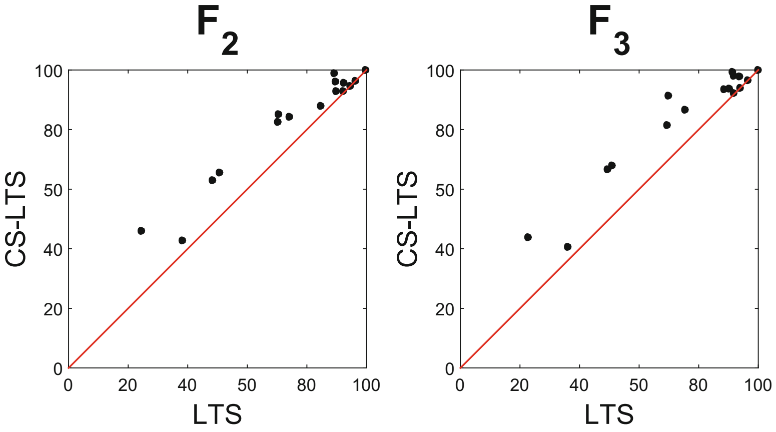

Fig. 4. F_2 and F_3 scores between CS-LTS and LTS for 16 balanced time-series datasets. (Left) In terms of F_2 score CS-LTS outperforms or is comparable to LTS in all 16 datasets. (Right) In terms of F_3 score CS-LTS outperforms or is comparable to LTS in all 16 datasets.

Table 1. Imbalanced datasets constructed from UCR Repository [6] where $*$ is the index of the original class that is assumed as the positive class

Dataset	Training			Test		Length
	#Positive	#Negative	IM ratio	#Positive	#Negative	
FaceAll*	80–150	1000	6.7–12.5	91–123	977–1079	131
SLeaf*	35	450	12.9	40	600	128
TwoPatterns*	200	180	9	1001–1106	1894–1999	128
Wafer*	200	380–3000	1.9–15	562–6220	392–3402	152
Yoga*	200	800–900	4–4.5	1300–1570	730–870	426

constructed by Cao et al. [4] from 5 multi-class datasets from the UCR time-series repository and the details are shown in Table 1.

The main advantage of CS-LTS over LTS is its superior performance in case of imbalanced datasets. In Fig. 5, it is shown that CS-LTS comfortably outperforms LTS on all 18 imbalanced datasets in terms of both F_1 and F_2 scores.

Moreover, in comparison to the state-of-the-art methods for imbalanced time-series classification, CS-LTS is very competitive. As shown in Table 2 in terms of F_1 score. The best method per dataset is shown in bold. The proposed CS-LTS method attains the highest number of absolute wins (5.86 wins) where a point is awarded to a method if it attains the highest F_1 score among the rest of the baseline methods for that particular dataset. In case of draws, the point is split into equal fractions and awarded to each method having the highest F_1 for a particular dataset.

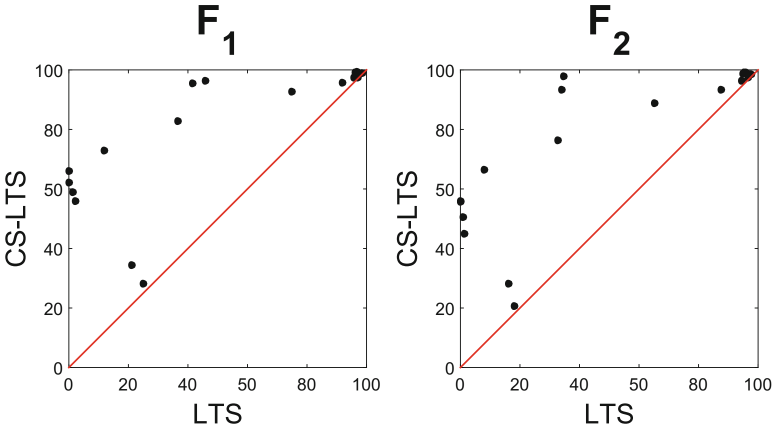

Fig. 5. F_1 and F_2 score between CS-LTS and LTS for 18 imbalanced time-series datasets. (Left) In terms of F_1 score CS-LTS achieves very high accuracy compared to LTS on 15 datasets and is comparable to LTS in 3. (Right) In terms of F_2 score CS-LTS outperforms or is comparable to LTS in all 18 datasets.

5 Discussion

Amongst the baselines, SPO [2], SMOTE [5], BORSMOTE [12], ADASYN [13], DB [11] and MoGT [4] are over-sampling techniques which mostly act as a preprocessing technique to over sample the rare class examples in order to construct balanced datasets. Easy [17] and Balanced [17] are under-sampling methods which reduces the number of examples from the majority class via under-sampling the majority class to balance the datasets.

From Table 2, we can infer that CS-LTS beats LTS and Easy across all datasets except 1 dataset (TwoPatterns3) in case of LTS which is a draw. Comparing with other baseline methods we see that CS-LTS has achieved similar accuracy as baseline methods on more than one datasets (such as wafer0 and wafer1). CS-LTS achieves comparable results with almost all of the over-sampling methods except for sleaf1 and TwoPatterns3 dataset. Results of CS-LTS on Sleaf1 and TwoPatterns3 certainly outperform LTS by huge margins; however, due to overlapping data-points in the feature space, it is hard for a linear model to achieve high classification accuracy in these two datasets. Compared to under-sampling methods (Easy and Balanced), CS-LTS is better than these baseline methods on most of the datasets. Another comparable method is the 1-Nearest Neighbor method (1-NN) which is known to be a good classifier for time-series classification problems. However, 1-NN computationally suffers from high dimensionality, hence it is time consuming compared to our method. Moreover, CS-LTS is an easier-to-interpret method as compared to 1-NN which makes it more desirable to domain experts. CS-LTS is an algorithmic approach to solve the imbalanced time-series classification problem whereas the state-of-the-art methods in this field are data manipulation methods that use over-sampling

Table 2. Comparison of mean F_1 scores for various baseline methods against proposed method. CS-LTS achieves highest absolute wins.

Dataset	SPO [2]	Repeat	SMOTE [5]	BORSMOTE [12]	ADASYN [13]	DB [11]	1MoGT [4]	2MoGT [4]	1 NN	Easy [17]	Balanced [17]	LTS [10]	CS-LTS
FaceAll1	96 (0.9)	94 (0.0)	95 (0.6)	95 (0.5)	95 (0.5)	95 (0.8)	96 (0.5)	97 (0.5)	98 (0.0)	67 (5.9)	86 (2.4)	98 (0.4)	**99 (0.2)**
FaceAll2	93 (1.0)	83 (0.0)	88 (0.5)	88 (0.7)	88 (0.8)	92 (0.4)	90 (0.5)	86 (0.8)	83 (0.0)	76 (3.2)	93 (1.3)	93 (0.4)	**95 (0.4)**
FaceAll3	95 (0.6)	**97 (0.0)**	96 (0.6)	**97 (0.2)**	96 (0.4)	91 (0.4)	95 (0.6)	94 (0.6)	**97 (0.0)**	60 (6.6)	73 (2.7)	90 (2.6)	92 (0.4)
FaceAll4	94 (0.5)	96 (0.0)	95 (0.6)	96 (0.5)	96 (0.5)	90 (1.0)	95 (0.5)	95 (0.5)	96 (0.0)	72 (3.0)	87 (2.7)	94 (0.3)	**98 (0.1)**
FaceAll5	96 (0.4)	**97 (0.0)**	**97 (0.1)**	**97 (0.1)**	**97 (0.2)**	95 (0.3)	**97 (0.2)**	95 (0.3)	95 (0.0)	85 (2.5)	92 (1.1)	95 (0.4)	**97 (0.1)**
SLeaf1	83 (0.8)	81 (0.0)	79 (1.4)	79 (1.6)	79 (1.6)	81 (1.6)	**87 (2.1)**	83 (1.7)	57 (0.0)	54 (5.1)	50 (4.4)	4 (20.2)	49 (1.8)
SLeaf2	96 (1.0)	94 (0.0)	95 (0.7)	96 (0.0)	96 (0.4)	96 (0.0)	**98 (0.7)**	95 (0.3)	91 (0.0)	85 (6.7)	87 (3.9)	96 (1.5)	**98 (0.5)**
SLeaf3	**88 (1.6)**	83 (0.0)	83 (1.0)	83 (1.1)	83 (1.1)	82 (0.5)	84 (0.7)	84 (1.4)	66 (0.0)	66 (5.6)	54 (6.6)	0.0 (0.0)	84 (1.6)
SLeaf4	**93 (1.0)**	61 (0.0)	72 (2.4)	71 (0.7)	73 (0.4)	89 (1.5)	83 (2.9)	88 (1.9)	68 (0.0)	56 (7.9)	66 (4.8)	66 (36.7)	88 (0.0)
SLeaf5	**90 (1.1)**	88 (0.0)	89 (0.7)	89 (0.7)	89 (0.6)	87 (0.8)	89 (1.0)	89 (0.8)	71 (0.0)	59 (8.3)	52 (5.2)	36 (3.9)	82 (1.4)
TwoPatterns1	92 (0.3)	71 (0.0)	77 (0.2)	77 (0.2)	78 (0.3)	89 (0.2)	84 (0.6)	84 (0.6)	92 (0.0)	95 (4.0)	75 (1.6)	96 (1.4)	**99 (1.4)**
TwoPattern2	78 (0.7)	65 (0.0)	68 (0.3)	68 (0.1)	68 (0.2)	73 (0.2)	75 (0.5)	81 (0.6)	**89 (0.0)**	31 (2.4)	68 (1.2)	51 (1.7)	72 (3.4)
Twopattern3	86 (0.3)	65 (0.0)	70 (0.4)	71 (0.5)	71 (0.7)	57 (0.2)	82 (0.6)	89 (0.6)	**91 (0.0)**	36 (3.0)	69 (0.9)	5 (13.1)	51 (11.3)
TwoPattern4	90 (0.5)	68 (0.0)	73 (0.2)	73 (0.2)	73 (0.2)	73 (0.2)	82 (0.7)	87 (0.4)	87 (0.0)	35 (2.5)	71 (1.5)	96 (1.4)	**99 (1.2)**
Wafer0	**99 (0.0)**	**99 (0.0)**	**99 (0.0)**	**99 (0.0)**	**99 (0.0)**	**99 (0.0)**	**99 (0.0)**	**99 (0.0)**	**99 (0.0)**	93 (1.1)	**99 (0.1)**	98 (0.6)	**99 (0.0)**
Wafer1	**99 (0.1)**	**99 (0.0)**	**99 (0.1)**	**99 (0.0)**	**99 (0.1)**	**99 (0.1)**	**99 (0.1)**	**99 (0.1)**	98 (0.0)	93 (0.8)	98 (0.6)	97 (1.6)	**99 (0.1)**
Yoga1	89 (0.2)	88 (0.0)	**90 (0.1)**	**90 (0.2)**	**90 (0.2)**	88 (0.0)	88 (0.1)	88 (0.2)	83 (0.0)	59 (2.5)	85 (0.6)	24 (1.7)	70 (0.9)
Yoga2	**91 (0.2)**	90 (0.0)	**91 (0.1)**	**91 (0.1)**	**91 (0.1)**	**91 (0.0)**	90 (0.1)	**91 (0.1)**	86 (0.0)	61 (2.5)	87 (0.7)	5 (0.0)	81 (1.7)
Absolute Wins	3.36	0.69	0.85	1.18	0.85	0.36	1.86	0.36	2.42	0	0.09	0	5.86

and under-sampling techniques, which act as a preprocessing step to solve the high imbalance time-series classification problem. Figure 6 shows the critical difference diagram amongst all the baseline methods and CS-LTS.

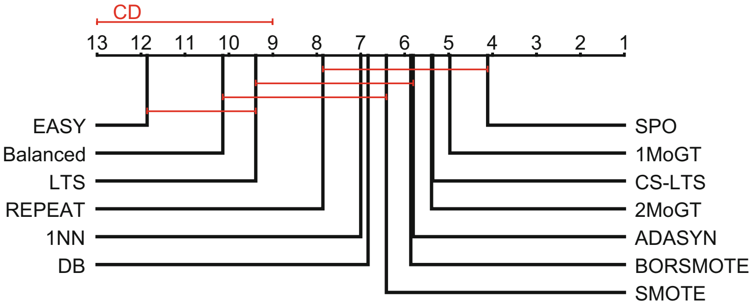

Fig. 6. Critical difference diagram showing average rank of CS-LTS against all baseline methods on 18 imbalanced datasets.

6 Conclusion

In this paper, we adapt the novel perspective of learning generalized shapelets for time-series classification via a logistic loss minimization, and extend the time-series classification framework to a cost-sensitive framework that can handle highly imbalanced time-series datasets. In contrast to the baseline model, whose prediction accuracy is biased towards the abundant negative class, the proposed CS-LTS does not suffer from class imbalance problem. Extensive experiments on 36 real-world time-series datasets reveal the proposed method is a good alternative to the baseline model. It can handle both balanced and imbalanced time-series datasets and achieve better or comparable results against the current state-of-the-art methods. In future, we plan to extend the cost-sensitive learning framework for multivariate time-series datasets in order to improve the performance of the model.

Acknowledgments. This research was supported in part by DARPA grant No. FA9550-12-1-0406 negotiated by AFOSR, the National Science Foundation grants NSF-SES-1447670, NSF-IIS-1636772 and Temple University Data Science Targeted Funding Program. Simulations were performed on the OwlsNest HPC cluster at Temple University, which is supported in part by the National Science Foundation through NSF grant CNS-1625061 and Pennsylvania Department of Health CURE grant.

References

1. Behar, J., Oster, J., Li, Q., Clifford, G.: ECG signal quality during arrhythmia and its application to false alarm reduction. IEEE Trans. Biomed. Eng. **60**(6), 1660–1666 (2013)

2. Cao, H., Li, X., Woon, D.Y., Ng, S.: SPO: structure preserving oversampling for imbalanced time series classification. In: 11th IEEE International Conference on Data Mining, ICDM 2011, Vancouver, BC, Canada, 11–14 December 2011, pp. 1008–1013 (2011)
3. Cao, H., Li, X., Woon, D.Y., Ng, S.: Integrated oversampling for imbalanced time series classification. IEEE Trans. Knowl. Data Eng. **25**(12), 2809–2822 (2013)
4. Cao, H., Tan, V.Y.F., Pang, J.Z.F.: A parsimonious mixture of Gaussian trees model for oversampling in imbalanced and multimodal time-series classification. IEEE Trans. Neural Netw. Learn. Syst. **25**(12), 2226–2239 (2014)
5. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. J. Artif. Int. Res. **16**(1), 321–357 (2002)
6. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The UCR time series classification archive, July 2015
7. Drew, B.J., Harris, P., Zgre-Hemsey, J.K., Mammone, T., Schindler, D., Salas-Boni, R., Bai, Y., Tinoco, A., Ding, Q., Hu, X.: Insights into the problem of alarm fatigue with physiologic monitor devices: a comprehensive observational study of consecutive intensive care unit patients. PLoS ONE **9**(10), e110274 (2014)
8. Ghalwash, M., Radosavljevic, V., Obradovic, Z.: Utilizing temporal patterns for estimating uncertainty in interpretable early decision making. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 402–411 (2014)
9. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. Circulation **101**(23), e215–e220 (2000)
10. Grabocka, J., Schilling, N., Wistuba, M., Schmidt-Thieme, L.: Learning time-series shapelets. In: Proceedings of 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2014, pp. 392–401. ACM (2014)
11. Guo, H., Viktor, H.L.: Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach. SIGKDD Explor. Newsl. **6**(1), 30–39 (2004)
12. Han, H., Wang, W.-Y., Mao, B.-H.: Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In: Huang, D.-S., Zhang, X.-P., Huang, G.-B. (eds.) ICIC 2005. LNCS, vol. 3644, pp. 878–887. Springer, Heidelberg (2005). https://doi.org/10.1007/11538059_91
13. He, H., Bai, Y., Garcia, E.A., Li, S.: ADASYN: adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of International Joint Conference on Neural Networks, IJCNN 2008, Part of the IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, China, 1–6 June 2008, pp. 1322–1328 (2008)
14. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9), 1263–1284 (2009)
15. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data Min. Knowl. Discov. **28**(4), 851–881 (2014)
16. Hou, L., Kwok, J.T., Zurada, J.M.: Efficient learning of timeseries shapelets. In: Proceedings of 30th AAAI Conference on Artificial Intelligence, 12–17 February 2016, Phoenix, Arizona, USA, pp. 1209–1215 (2016)
17. Liu, X.Y., Wu, J., Zhou, Z.H.: Exploratory undersampling for class-imbalance learning. Trans. Sys. Man Cyber. Part B **39**(2), 539–550 (2009)

18. Lpez, V., Fernndez, A., Garca, S., Palade, V., Herrera, F.: An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics. Inf. Sci. **250**, 113–141 (2013)
19. Radosavljevic, V., Vucetic, S., Obradovic, Z.: Continuous conditional random fields for regression in remote sensing. In: Proceedings of 2010 Conference on ECAI 2010: 19th European Conference on Artificial Intelligence, pp. 809–814. IOS Press, Amsterdam (2010)
20. Roychoudhury, S., Ghalwash, M.F., Obradovic, Z.: False alarm suppression in early prediction of cardiac arrhythmia. In: 2015 IEEE 15th International Conference on Bioinformatics and Bioengineering (BIBE), pp. 1–6, November 2015
21. Saeed, M., Villarroel, M., Reisner, A., Clifford, G., Lehman, L.W., Moody, G., Heldt, T., Kyaw, T., Moody, B., Mark, R.: Multiparameter intelligent monitoring in intensive care II: a public-access intensive care unit database. Crit. Care Med. **39**(5), 952–960 (2011)
22. Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recogn. **40**(12), 3358–3378 (2007)
23. Ye, L., Keogh, E.: Time series shapelets: a new primitive for data mining. In: Proceedings of 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2009, pp. 947–956. ACM, New York (2009)
24. Zhang, Q., Wu, J., Yang, H., Tian, Y., Zhang, C.: Unsupervised feature learning from time series. In: Proceedings of 25th International Joint Conference on Artificial Intelligence, IJCAI 2016, 9–15 July 2016, New York, NY, USA, pp. 2322–2328 (2016)