THE EUROPEAN CONFERENCE ON MACHINE LEARNING & PRINCIPLES AND PRACTICE OF KNOWLEDGE DISCOVERY IN DATABASES (ECML/PKDD 2017)

# Cost Sensitive Time-series Classification


Shoumik Roychoudhury, Mohamed Ghalwash, Zoran Obradovic

Presented by: Martin Pavlovski

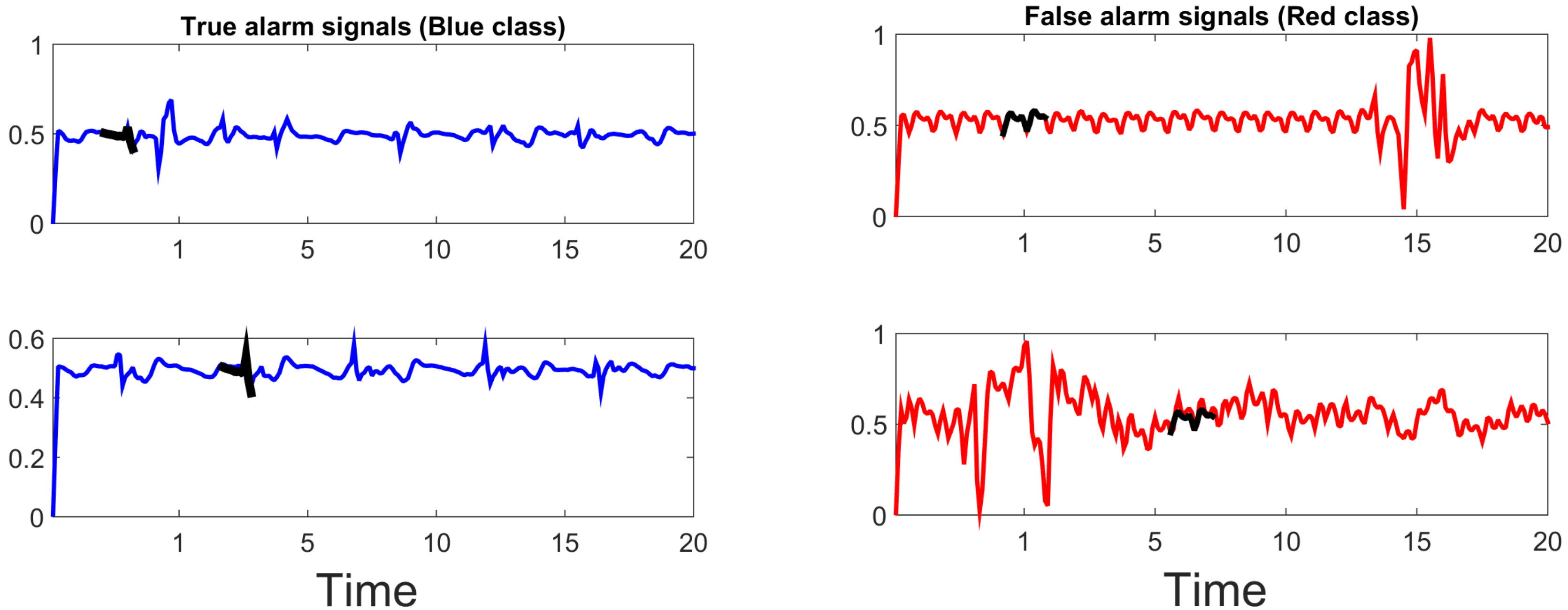
Prepared By Shoumik (shoumik.rc@temple.edu)

Center for Data Analytics and Biomedical Informatics

TEMPLE UNIVERSITY

# Motivating application

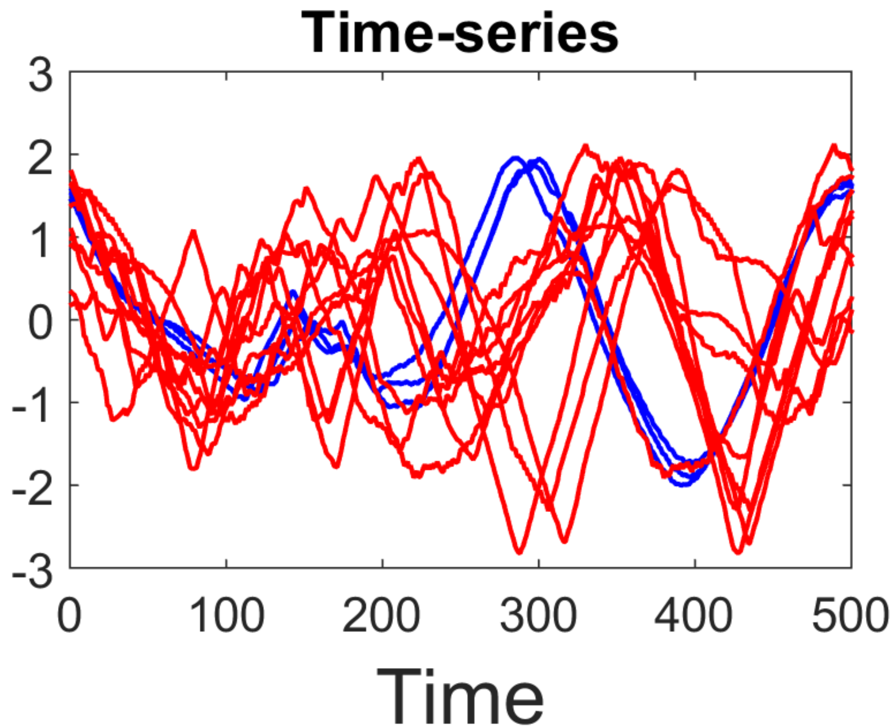
- Suppression of false Cardiac Arrhythmia alarms in ICU patients.




- High percentages of ICU bedside monitor cardiac false alarms.
- Blue signals (Positive class) are ECG II signals of true cardiac arrhythmia alarms.
- Red signals (Negative class) are ECG II signals of false cardiac arrhythmia alarms.
- Benefits: Improve *alarm fatigue* among caregivers inside ICU.

# Introduction

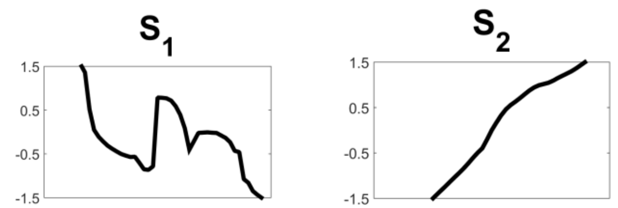
- One of the key sources of performance degradation in the field of time-series classification is the **class imbalance** problem.



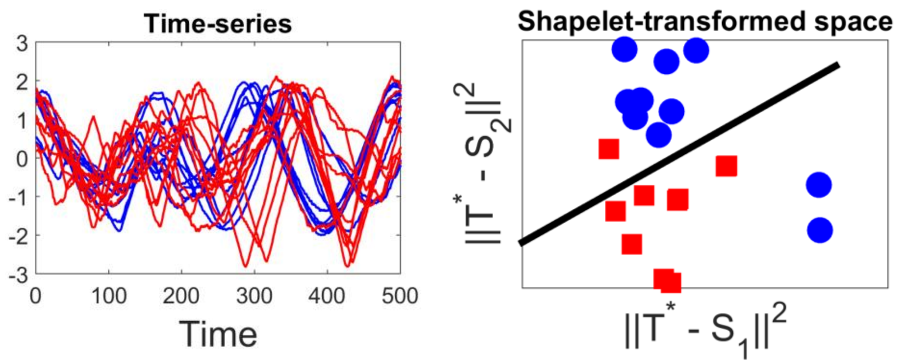

- Imbalanced **BirdChicken** dataset from UCR Time Series Classification Archive.
- The minority class (Positive class) is outnumbered by abundant majority negative class instances.

# Time-series classification

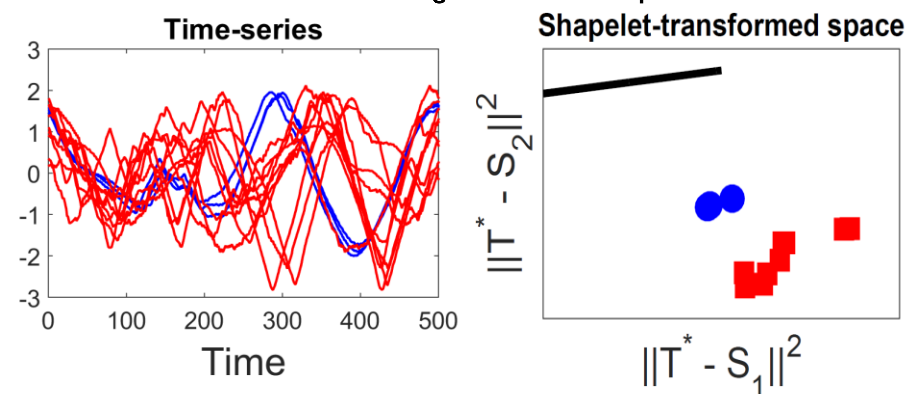
- Learning time-series shapelets (LTS)*
- Shapelets are short time-series sub-sequences ($S_1, S_2$)
- T* is the time-series dataset.




Learned generalized shapelets via LTS




Example 1:Balanced time-series dataset




Example 2:Imbalanced time-series dataset

- In LTS, a cost-insensitive **0-1 logistic loss function** is minimized in order to learn generalized shapelets.
- Traditional Logistic Loss function: $\mathcal{L}(Y,\hat{Y}) = -Y ln\sigma(\hat{Y}) - (1-Y)ln(1-\sigma(\hat{Y}))$
- Linear Model: $\hat{Y}_i = W_0 + \sum_{k=1}^{K} M_{i,k} W_k \quad \forall i \in \{1, ..., I\}$ where $M_{i,k} = \min_{j=1,\dots,J} \frac{1}{L} \sum_{l=1}^{L} (T_{i,j+l-1} - S_{k,l})^2$
- The minimum **Euclidean distances** $M_{i,k}$ of the **learned shapelets** $S_1, S_2$ to the time-series is used as features to linearly separate the examples in the **shapelet-transformed space.**

* Grabocka et al. (KDD 2014)

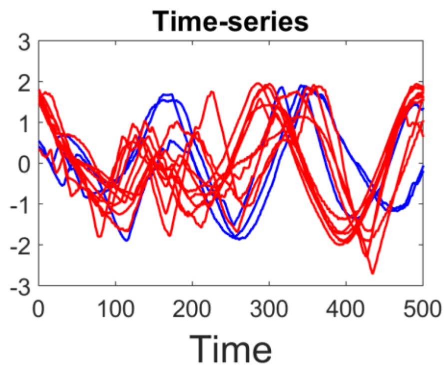
# Cost-sensitive classification

- Caveats of imbalanced data
  - Minimum classification error criterion based classification models generate biased models towards the majority class.
  - Higher misclassification error for minority class examples.
  - 0-1 loss function (e.g Logistic loss) classifier fail to differentiate between various misclassification costs (since classifiers are cost – insensitive).

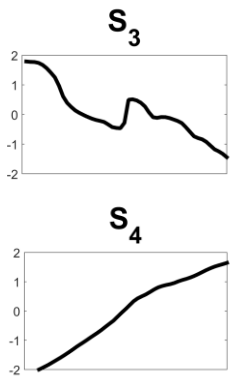
- Objective
  - Develop a cost-sensitive time-series classification framework
    - Specifically using **differentially weighted loss function** having variable **misclassification cost** for false positive and false negative errors.
    - Learning **interpretable temporal patterns** (shapelets).
    - **Learning misclassification cost** from data.
  - Jointly learn hyperplane parameters $W_k, W_0$ , shapelets $S$ and misclassification cost $C_{FP}, C_{FN}$ via constrained optimization.

# Time-series classification

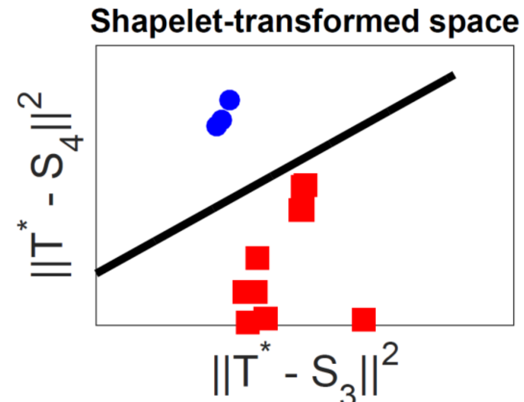
- Cost-sensitive time-series classification (CS-LTS)




Example 2:Imbalanced time-series dataset




Learned Shapelets via CS-LTS




Minimum Euclidean Distance of shapelet to time-series T*

- Cost-sensitive extension : $Z_i = \frac{1}{C_{FN}+C_{FP}} ln \frac{\sigma(\hat{Y})C_{FN}}{1-\sigma(\hat{Y})C_{FP}} = \frac{1}{C_{FN}+C_{FP}}(\hat{Y} + ln\frac{C_{FN}}{C_{FP}})$

- A **cost-sensitive logistic loss function** is minimized to enhance the modeling capability of LTS.

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

- Regularized objective function: $\underset{S,W,C}{\operatorname{argmin}} \mathcal{F}(S,W,C) = \underset{S,W,C}{\operatorname{argmin}} \sum_{i=1}^{I} \mathcal{L}(Y_i, Z_i) + \lambda_W \|W\|^2$

# CS-LTS

- Constrained optimization problem: $$\underset{S,W,C}{\operatorname{argmin}} \mathcal{F}(S,W,C)$$ $$\text{subject to } C_{FN} > 0,\ C_{FP} > 0$$ $$C_{FN} > \theta C_{FP}$$
- Objective of the model is to learn *S, W, C* that minimize *F.*
- **Stochastic Gradient descent** algorithm used to solve the optimization problem.
- Estimation the misclassification cost values is **a constrained optimization problem.** $C_{FN} > 0$, $C_{FP} > 0$ and $C_{FN} > \theta C_{FP}$, where $\theta \in \mathbb{Z}$.
- Convert the constrained optimization into an unconstrained optimization.

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

- Revised Objective function: $$\underset{S,W,C_{FP},\mathcal{D}}{\operatorname{argmin}} \mathcal{F}(S,W,C_{FP},\mathcal{D})$$ $$\text{subject to } C_{FP} > 0$$
- Gradients for false positive error: $\frac{\partial \mathcal{F}_i}{\partial \log c_{FP}} = c_{FP} \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial c_{FP}}$, $\frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial c_{FP}} = \frac{\partial \mathcal{L}(Y_i, Z_i)}{\partial Z_i} \frac{\partial Z_i}{\partial c_{FP}}$

# Contribution

- **Learns the misclassification costs**. No need to predetermine the cost values for misclassification errors.

- A **constrained optimization** problem is proposed which **jointly learns shapelets** (highly interpretable patterns), **their weights** (classification hyperplane parameters), and most importantly **misclassification costs**.

- Method effectiveness demonstrated on **life-threatening cardiac arrhythmia** dataset showing **improved true alarm detection rates** over the current state-of-the-art method for false alarm suppression.

- Evaluated extensively on **34 real-world time series datasets** with varied degree of imbalances and compared to a large set of **baseline methods**.

# Cardiac Arrhythmia Alarms Detection

- ECG lead II data
  - Two critical arrhythmia alarm datasets from MIMIC II version 3 repository.
    - Ventricular tachycardia (VTACH)
    - False alarm suppression challenge 2015 (CHALLENGE)

| Dataset | Total alarms | True alarms(%) | False alarms(%) |
|---|---|---|---|
| VTACH | 629 | 227(36.09%) | 402(63.91%) |
| CHALLENGE | 750 | 250(33.33%) | 500(66.66%) |

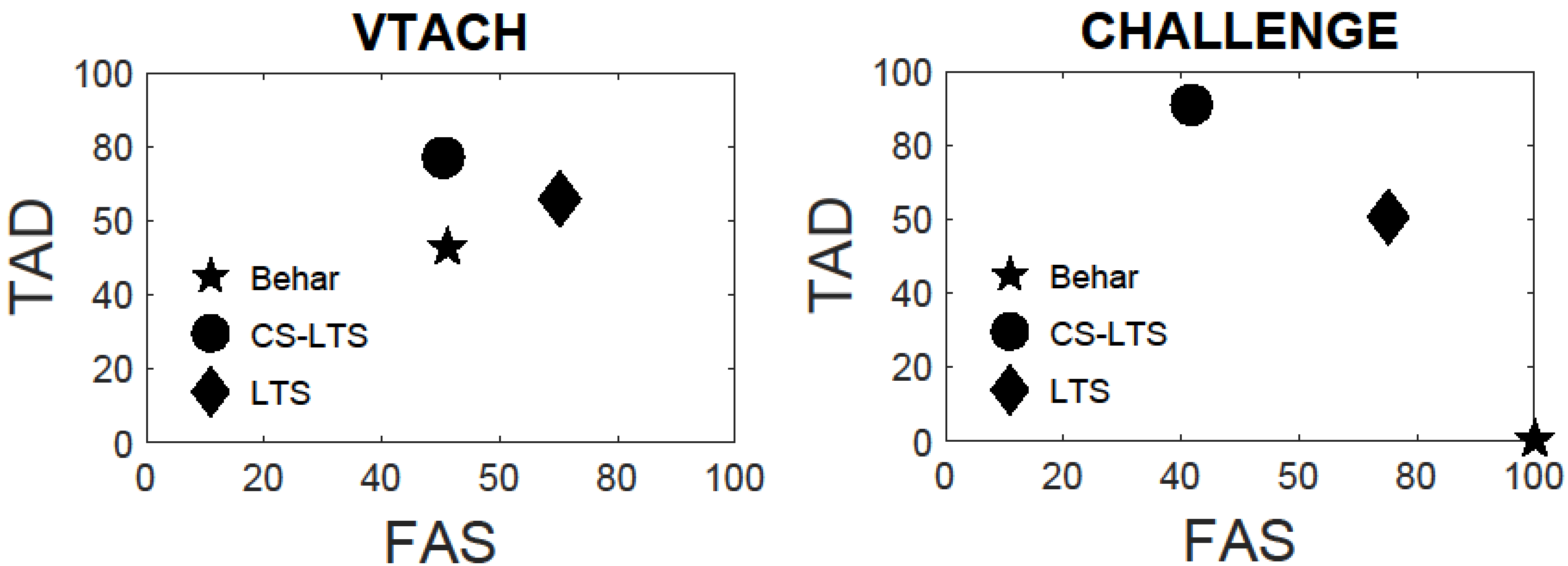
- Setup
  - For each alarm event, a 20-second window prior to the alarm event was extracted.
  - Dataset was partitioned into four distinct cross-validation datasets, where we train the model on 3 folds and test on the fourth one.
  - The entire process of cross-validation for 10 independent trials (each trial has 4 distinct partitions on true alarm instances) which results in 40 different combination of training data.

# Cardiac Arrhythmia Alarms Detection

- Evaluation measures
  - True alarm detection rate(TAD) = sensitivity
  - False alarm suppression rate(FAS) = specificity
  - $F_\beta = \frac{(1+\beta^2) * true\ positives}{(1+\beta^2) * true\ positive + \beta^2 * false\ negative + false\ positive}$
  - Baseline Methods
    - Behar et al (2013) black box method using feature extraction and SVM
    - Learning time-series shapelets (LTS) Grabocka et al. (2014)

# Cost Sensitive Cardiac Arrhythmia Alarms Detection

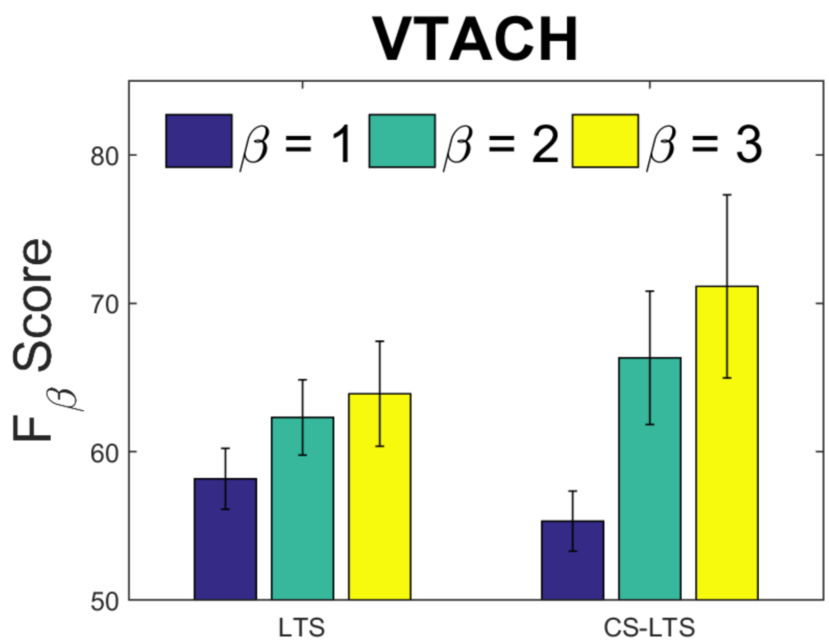
- Agenda: Achieve high FAS (X-axis) while keeping near 100% TAD (Y-axis).
- Increasing value in X-axis indicates high false alarm suppression and increasing value in Y-axis indicate high true alarm detection.
- The marking indicate performance of each model for both TAD and FAS together.

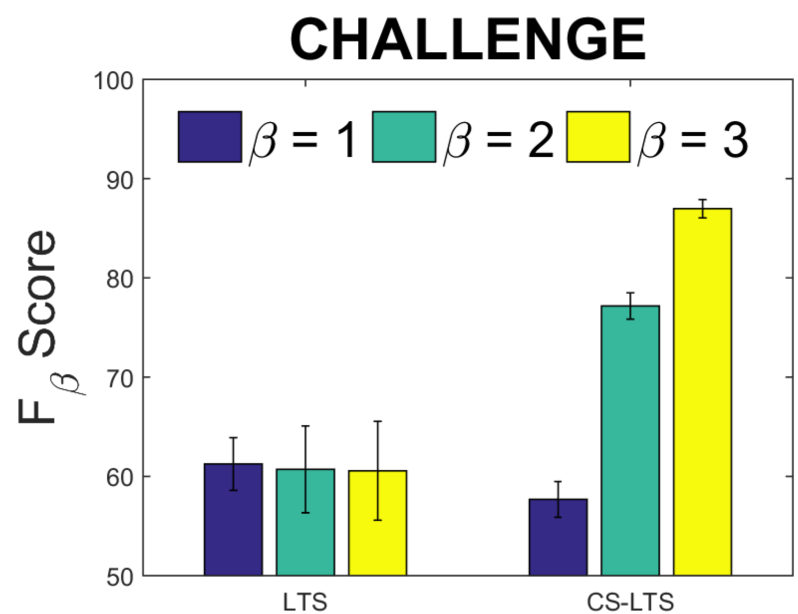



- Upper right hand corner in the figure is ideal result.(100 % FAS and 100 % TAD).
- Proposed method CS-LTS **(Circle)** outperform all baseline methods in terms of TAD in both the datasets.
- Baseline methods are better in terms of FAS however they make lot of false negative errors.

# Cost Sensitive Cardiac Arrhythmia Alarms Detection

- Agenda: True positives, false negatives and false positives should not have equal weights.
- Higher true positive means lesser missed true alarms. False negative errors are more costlier than false positives. False negative might result in patient death (missed true alarm).
- False negative errors are penalized more using $F_\beta$. For example, $\beta = 2, 3$ penalizes false negative error more and awards true positive more than $\beta = 1$ which represents harmonic mean.



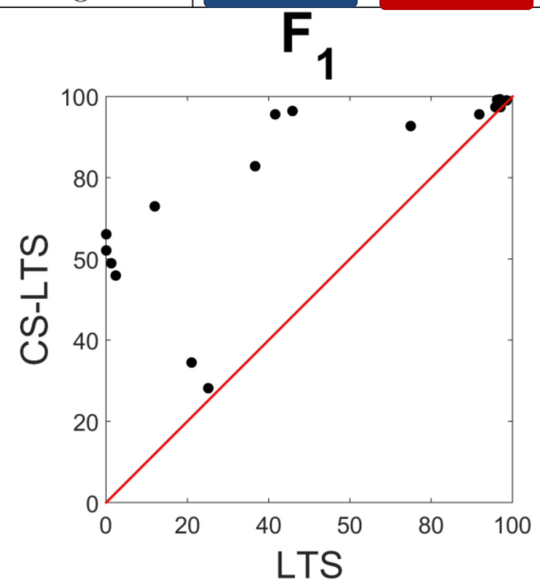



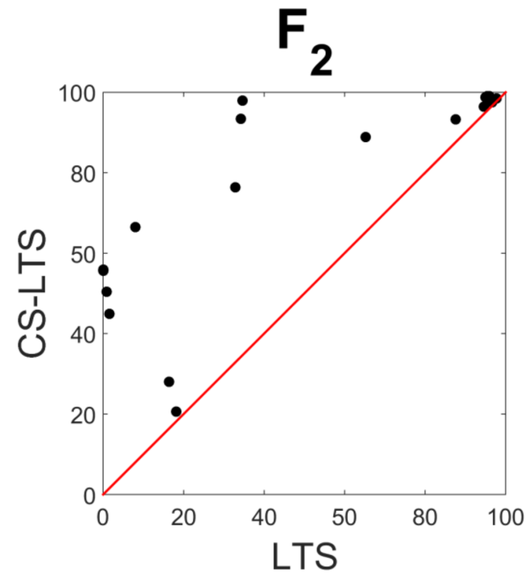

- CS-LTS method outperforms LTS for $\beta = 2, 3$.

# Imbalanced time-series datasets

- Agenda: Advantage of cost-sensitive learning over cost-insensitive learning.
- 18 highly imbalanced datasets generated from 5 multi-class datasets from UCR archives.

| 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 |



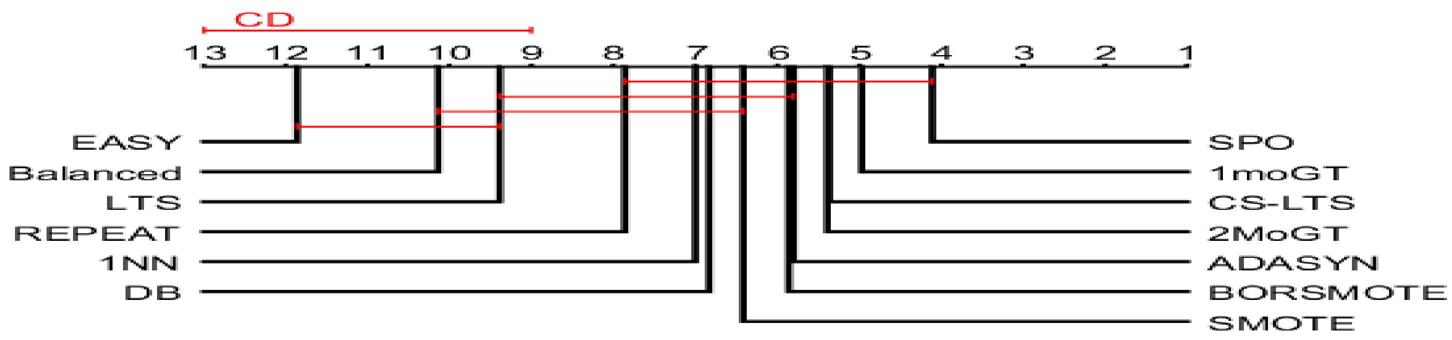

- CS-LTS outperforms or comparable LTS in terms of $F_1$ and $F_2$ on 18 highly imbalanced datasets.

# Baseline comparisons

- Proposed CS-LTS compared with 12 baseline methods (over 10 iterations).
- CS-LTS method attains the highest number of absolute wins (5.86 wins).
- 1 Point is awarded to a method with highest F1 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 F1 for a particular dataset.

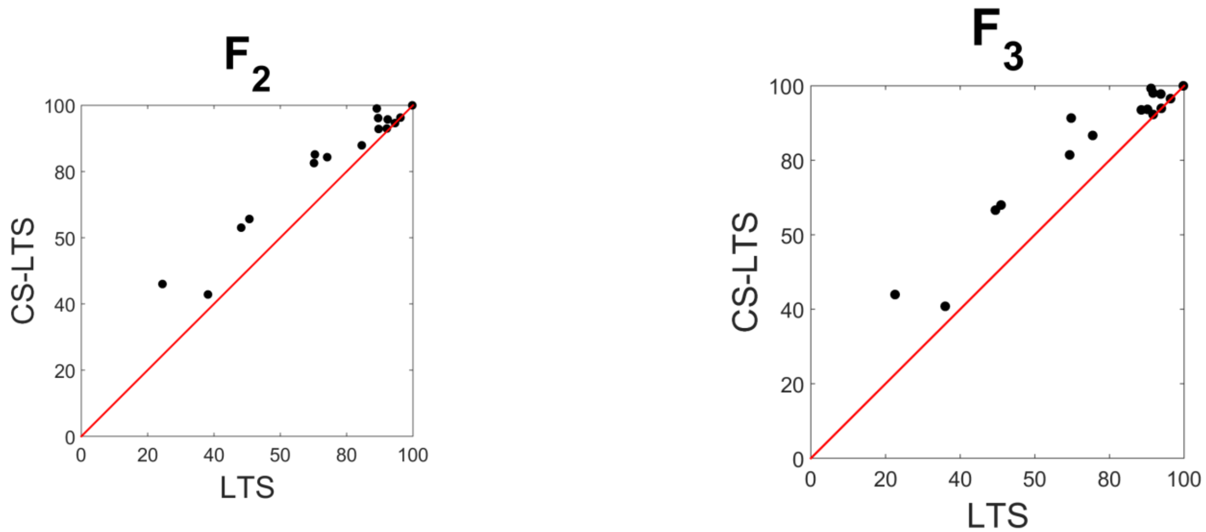
| Method | SPO | Repeat | SMOTE | BORSMOTE | ADASYN | DB | 1MoGT | 2MoGT | 1 NN | Easy | Balanced | LTS | CS-LTS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Absolute Wins | 3.36 | 0.69 | 0.85 | 1.18 | 0.85 | 0.36 | 1.86 | 0.36 | 2.42 | 0 | 0.09 | 0 | 5.86 |

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



# Balanced time-series datasets

- CS-LTS attains comparable or better classification accuracy when compared to LTS on balanced datasets from UCR Time-series dataset archive.
- 16 binary-class datasets were selected from UCR time-series repository. Default train/test splits were used.




- CS-LTS outperforms or comparable to LTS on all 16 datasets.
- CS-LTS model provides a good alternative to LTS as it can handle balanced datasets quite effectively.
- CS-LTS attains higher sensitivity with little loss of specificity when compared to LTS.

# Summary

- We extend the novel perspective of learning generalized shapelets for time-series classification via a logistic loss minimization.

- We extend the time-series classification framework to a cost-sensitive time-series classification framework that can handle **highly imbalanced** time-series datasets.

- 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.
- Future work
  - We plan to extend the cost-sensitive learning framework for multivariate time-series datasets in order to handle more datasets akin to real-world.

# Thank you

Further questions: shoumik.rc@temple.edu

# Additional Slides

# Learning Algorithm

- Gradients are computed as partial derivatives of the per-instance function.

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

- Following equation shows the point gradient of objective function for the $i^{th}$ time-series with respect to shapelet $S_k$

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

- The gradients of the weights of the hyperplane are

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

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

# CS-LTS

- Regularized objective function: $\underset{S,W,C}{\operatorname{argmin}} \mathcal{F}(S,W,C) = \underset{S,W,C}{\operatorname{argmin}} \sum_{i=1}^{I} \mathcal{L}(Y_i, Z_i) + \lambda_W \|W\|^2$
- Constrained optimization problem:

$$\underset{S,W,C}{\operatorname{argmin}} \mathcal{F}(S,W,C)$$

$$\text{subject to } C_{FN} > 0,\ C_{FP} > 0$$

$$C_{FN} > \theta C_{FP}$$

- Objective of the model is to learn *S, W, C* that minimize *F*
- Stochastic Gradient descent algorithm used to solve the optimization problem

**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** $S, W, W_0, C_{FP}$

# Learning Algorithm

- 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 $\theta \in \mathbb{Z}$.
- Convert the constrained optimization into an unconstrained optimization

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

- Revised Objective function: $$\underset{S,W,C_{FP},\mathcal{D}}{\operatorname{argmin}} \; \mathcal{F}(S, W, C_{FP}, \mathcal{D})$$ $$\text{subject to } C_{FP} > 0$$

- Gradients for false positive error:

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

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