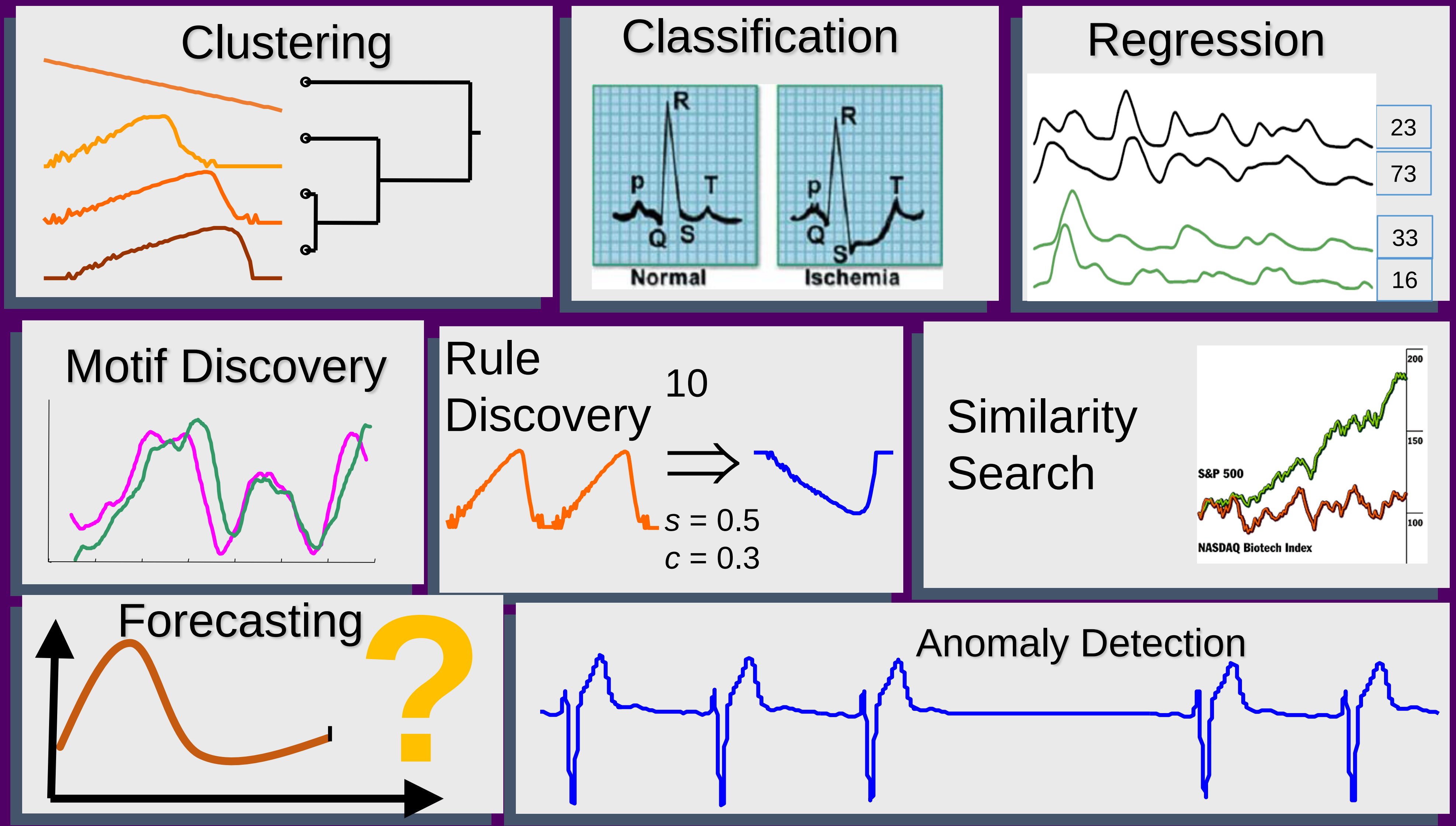
Time Series Machine Learning

with the aeon toolkit

Tony Bagnall
A.J.Bagnall@soton.ac.uk

School of Electronics and Computer Science
University of Southampton

Time Series Machine Learning Tasks



aeon: a toolkit for machine learning with time series

forecasting	classification	clustering	regression	transformations	distances	anomaly detection

aeon

Algorithm research

University of East Anglia
UCL
UCD Dublin
bcam basque center for applied mathematics
University of Oxford
Monash University
Universidad de Córdoba
USP Universidade de São Paulo
Humboldt-Universität zu Berlin

Scientific research

EPSRC Reference:	EP/W030756/2
Title:	aeon: a toolkit for machine learning with time series

Industry application

Daimler
Shell
GSK

Data science community

PyData
Google Summer of Code
Outreachy
NumFOCUS OPEN CODE = BETTER SCIENCE

Introduction to TSML

Lecture 1 (today)

- background and overview of tasks
- Time series classification

Lecture 2 (tomorrow)

- Classification and regression
- Time series clustering

Some pictures are taken with permission from talks given by Eamonn Keogh

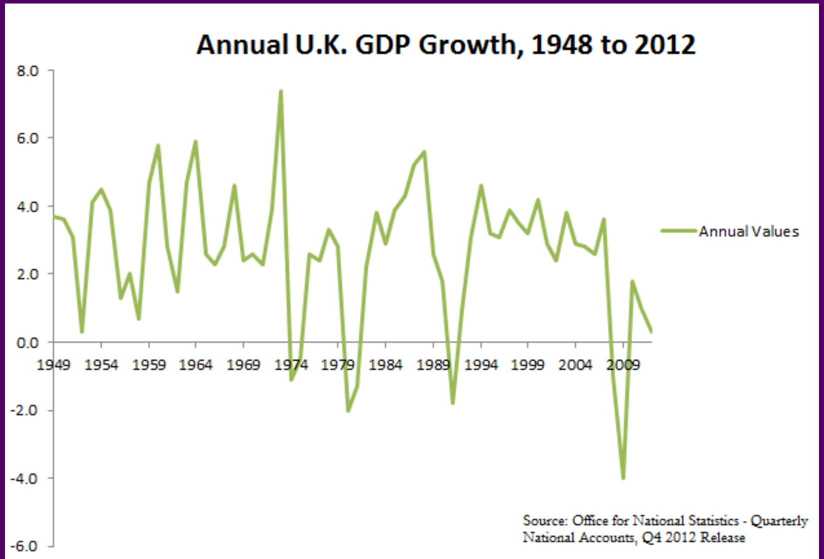
Some slides provided by aeon core developers Matthew Middlehurst, Chris Holder and Ali Fawaz

What is a time series?

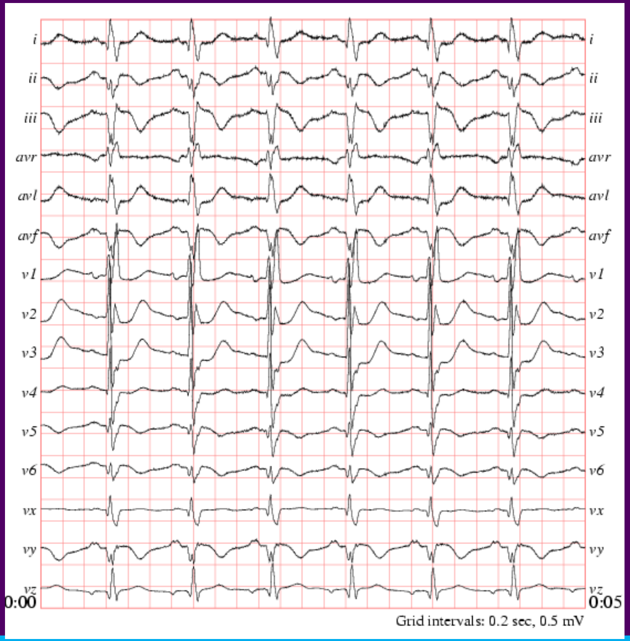
A time series is an ordered list of observations of real valued variable

If each observation is a scalar, we call it a univariate time series

If each observation is a vector of observations we call it a multivariate time series



The data does not need be ordered in time (sometimes called a data series)



Time Series Machine Learning Repository
www.timeseriesclassification.com

Introduced in 2002 by Eamonn Keogh and expanded several times since, the archive datasets have been used in thousands of papers

A large proportion donated by the TSML group at UEA/Southampton

Expanded in 2018 to 128 datasets
Multivariate Archive introduced in 2019 with 30 datasets

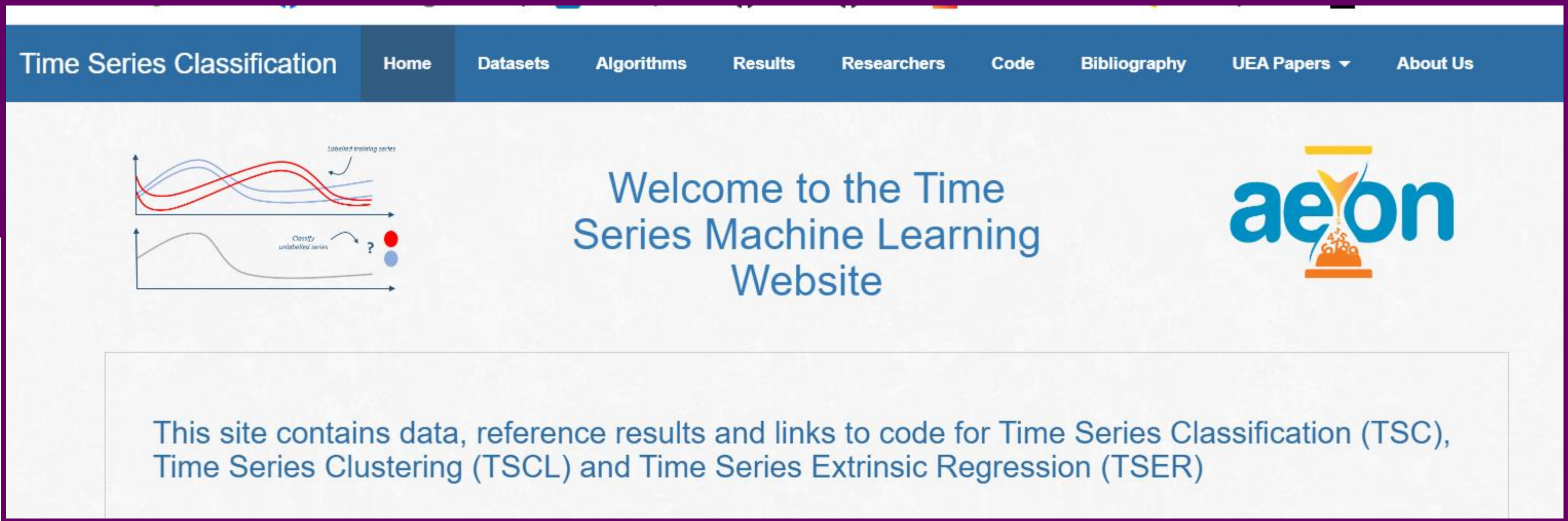
2002: 22

2015: 85
2018: 128
now > 200

University of East Anglia

University of Southampton

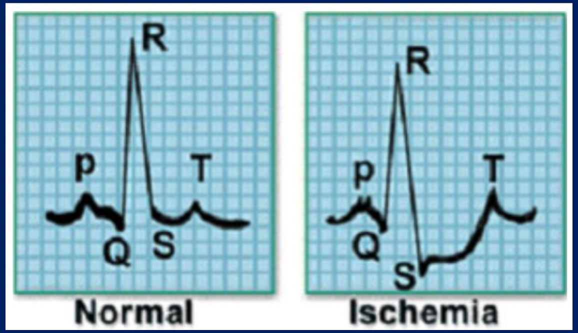
UNIVERSITY OF CALIFORNIA
UC RIVERSIDE

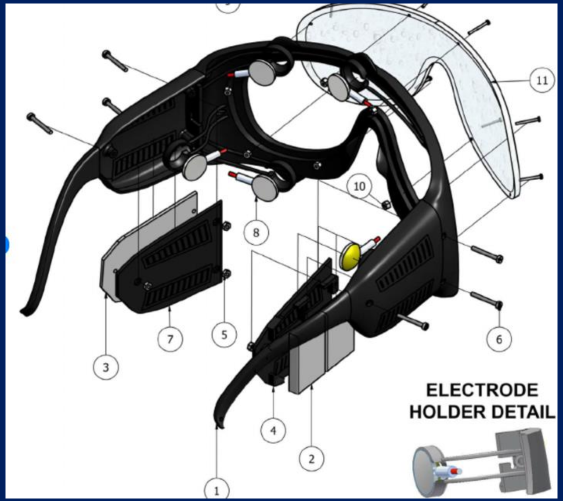


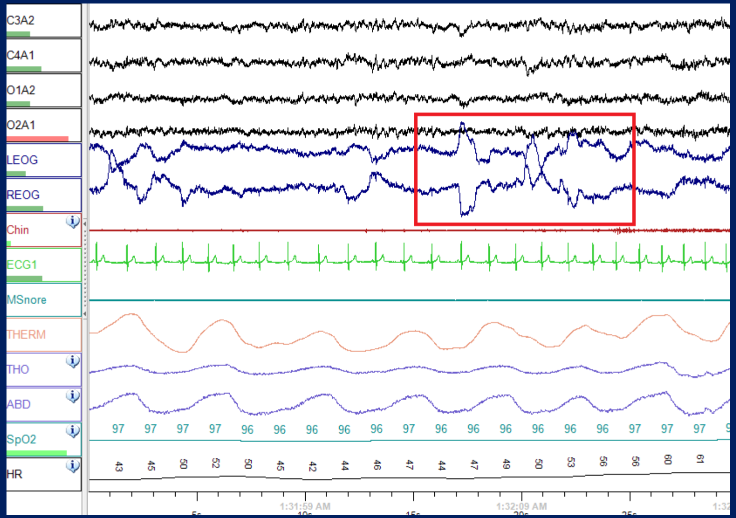
We would like to thank everyone who donates and helps maintain these archives

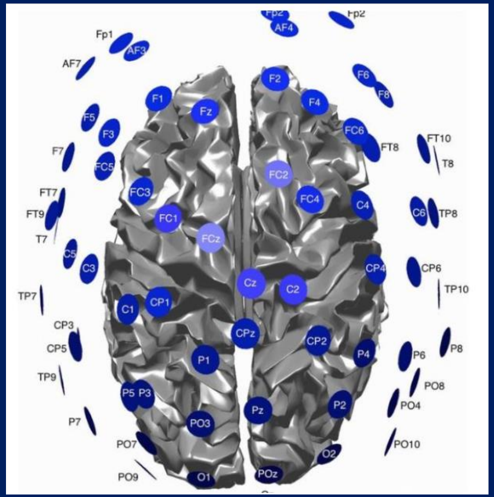
Biomedical signals

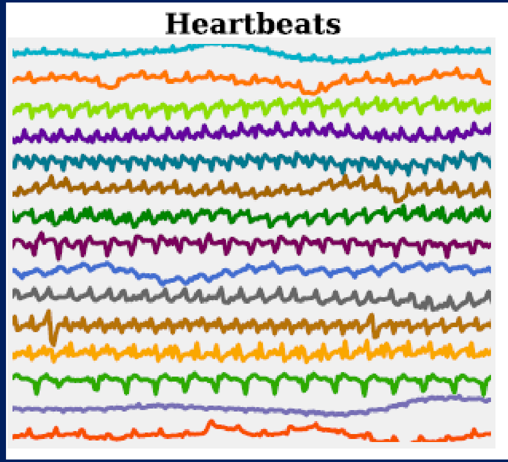
ECG, EEG, MEG, EOG

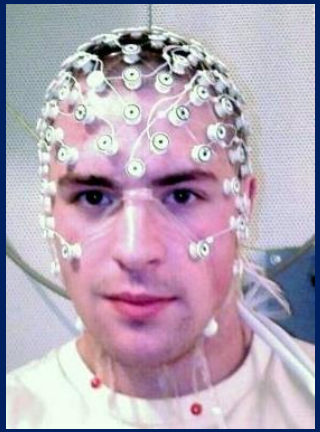


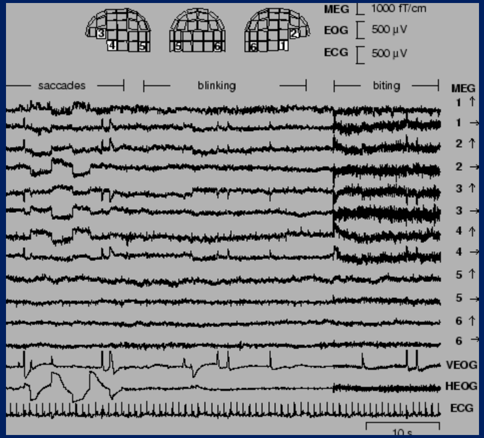


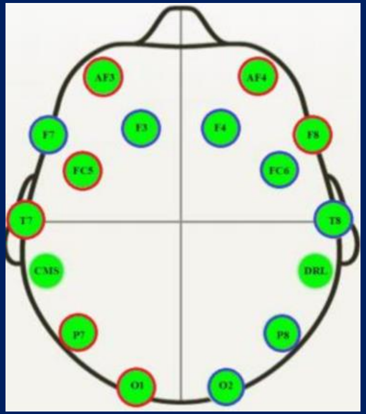






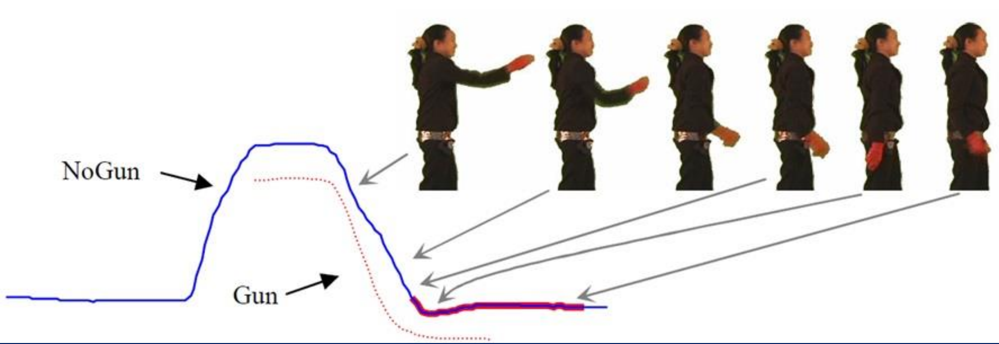


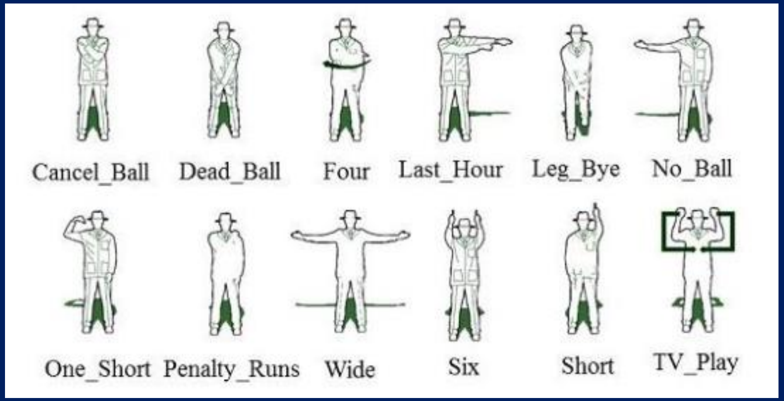


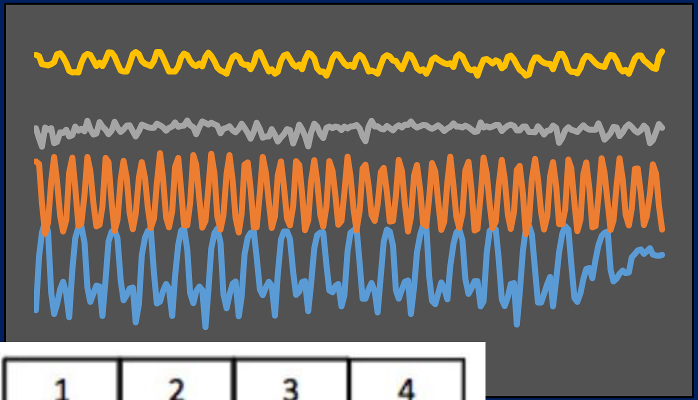


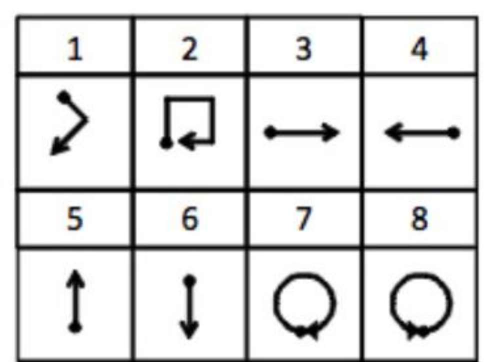
Human Activity Recognition

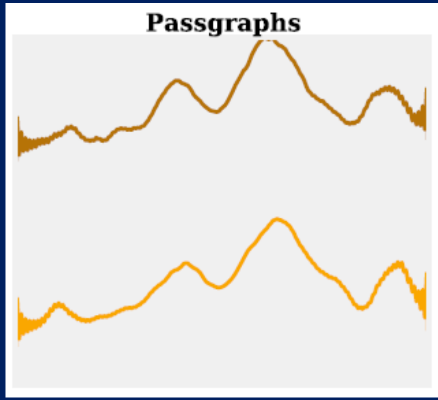
Gestures (Uwave), Cricket hand signals, Gun Point, Asphalt road condition, inline skating

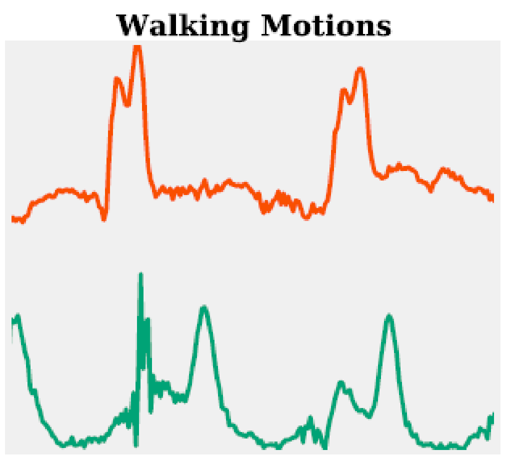


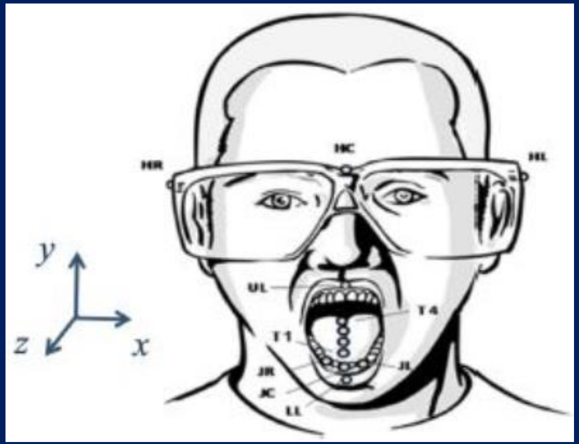








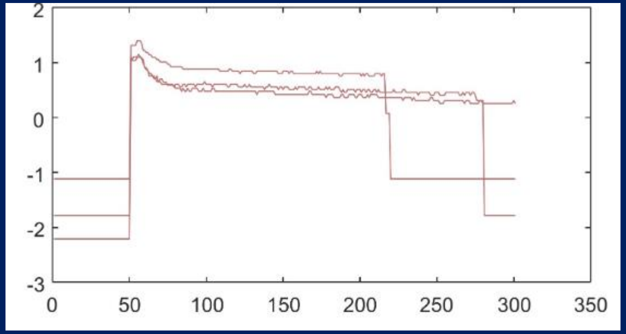


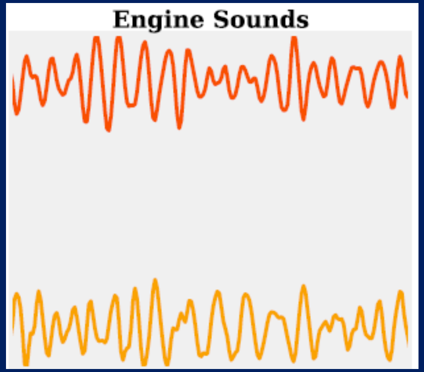


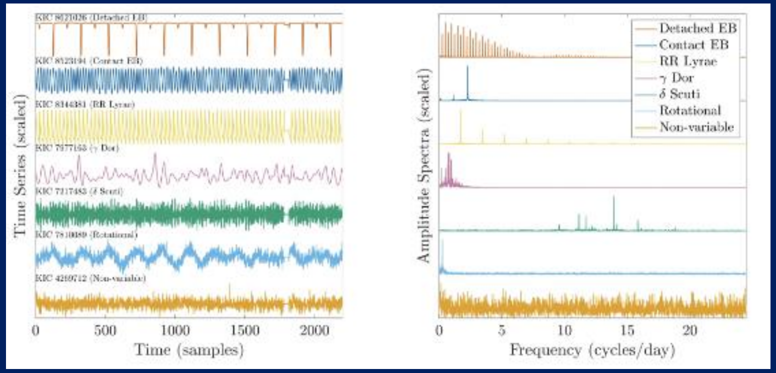
Sensor Data

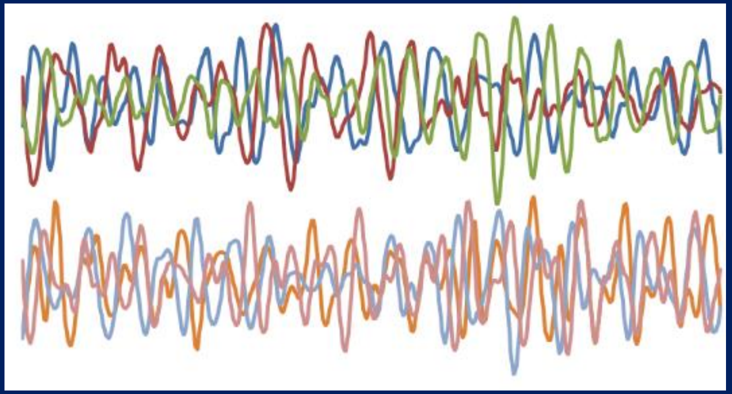
Car engines, light curves, lightning, electric devices

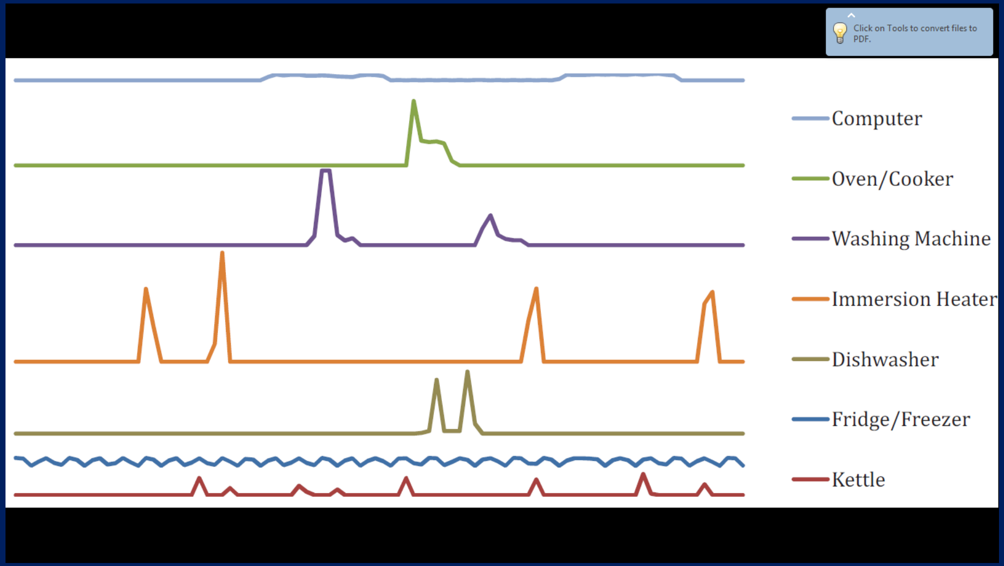
Insect wing beats, Car Engines, Phonemes (sound), Worm Motion

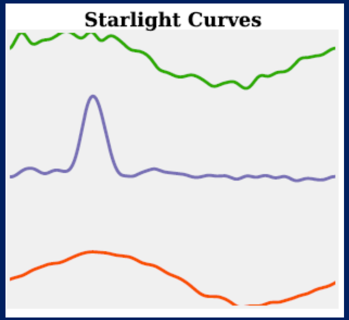












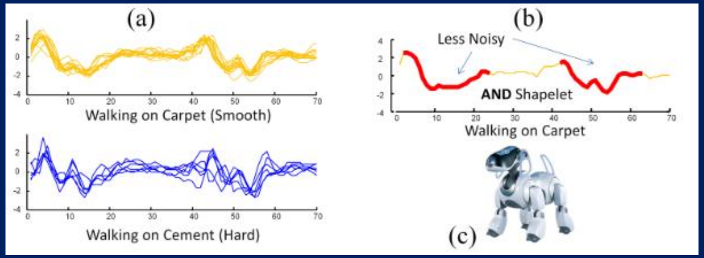
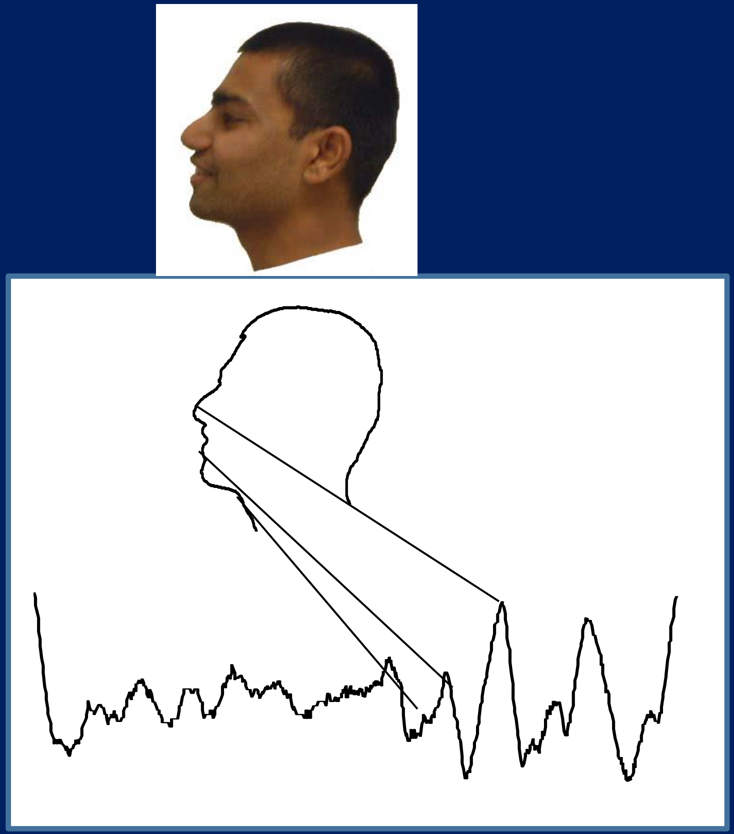
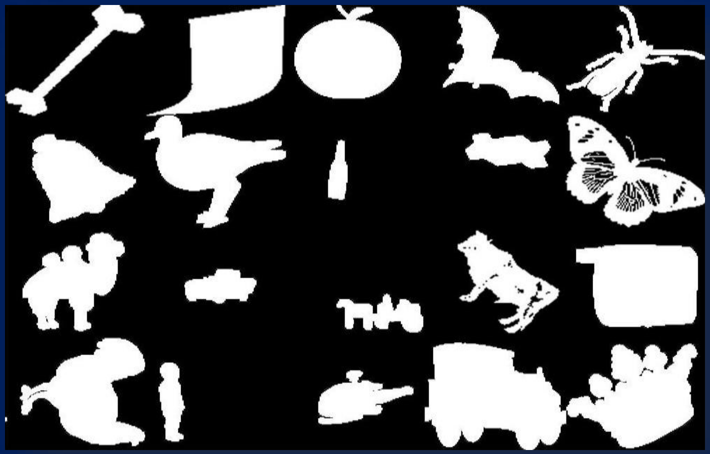
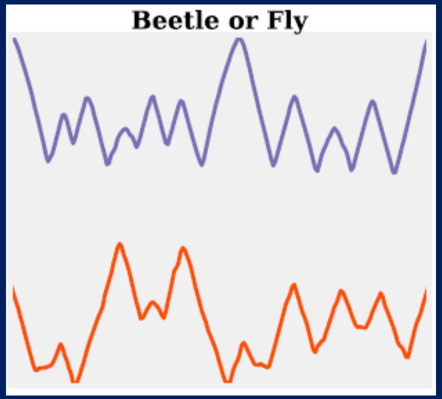


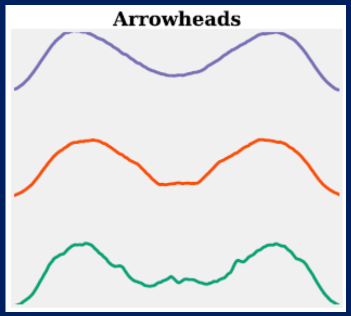
Image Outlines

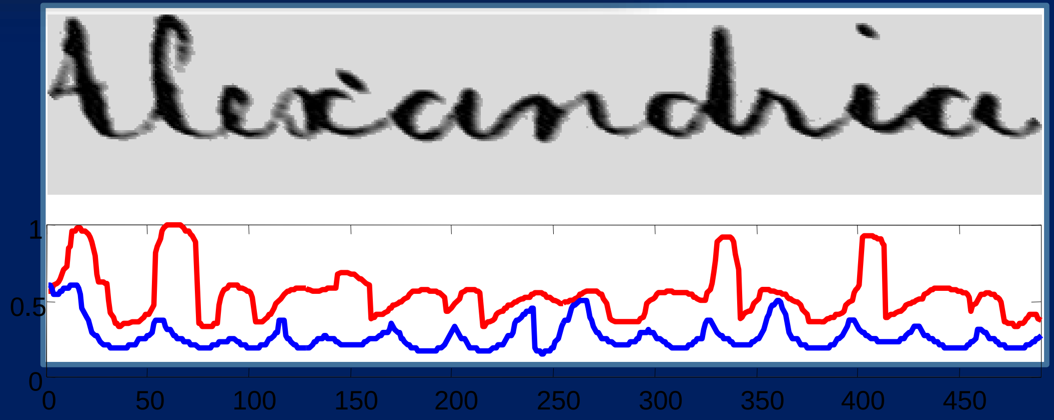
Hand and bone outlines, Herring Otoliths, Faces, Leaves, Arrow Heads, Yoga, Words/letter, Shapes (MPEG7)







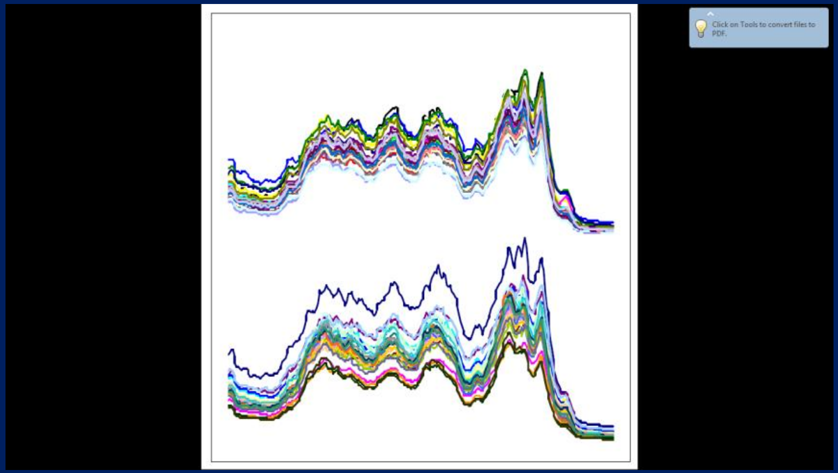


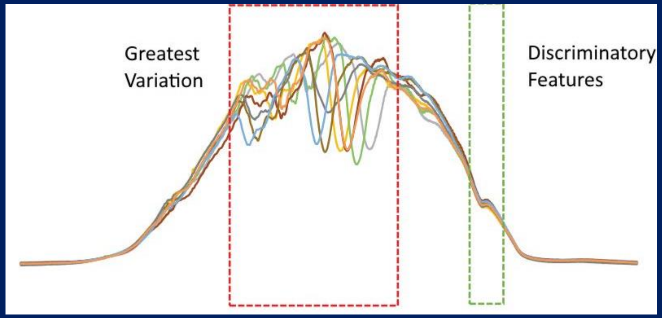


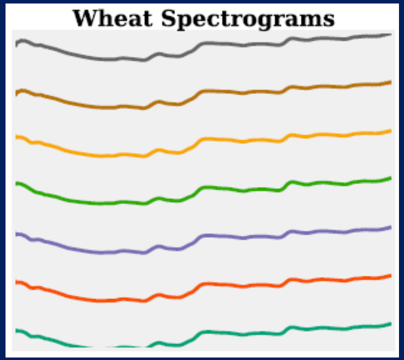
Food Spectrographs

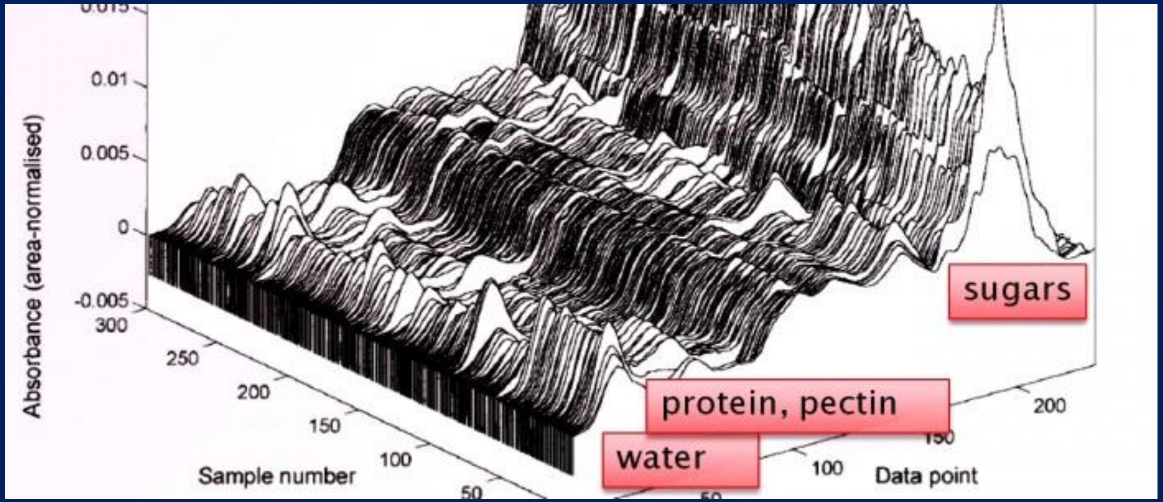
Beef, Coffee, Ham, Meat, Olive Oil, Strawberry, Wine, Whisky

Quadram Institute
Science Health Food Innovation

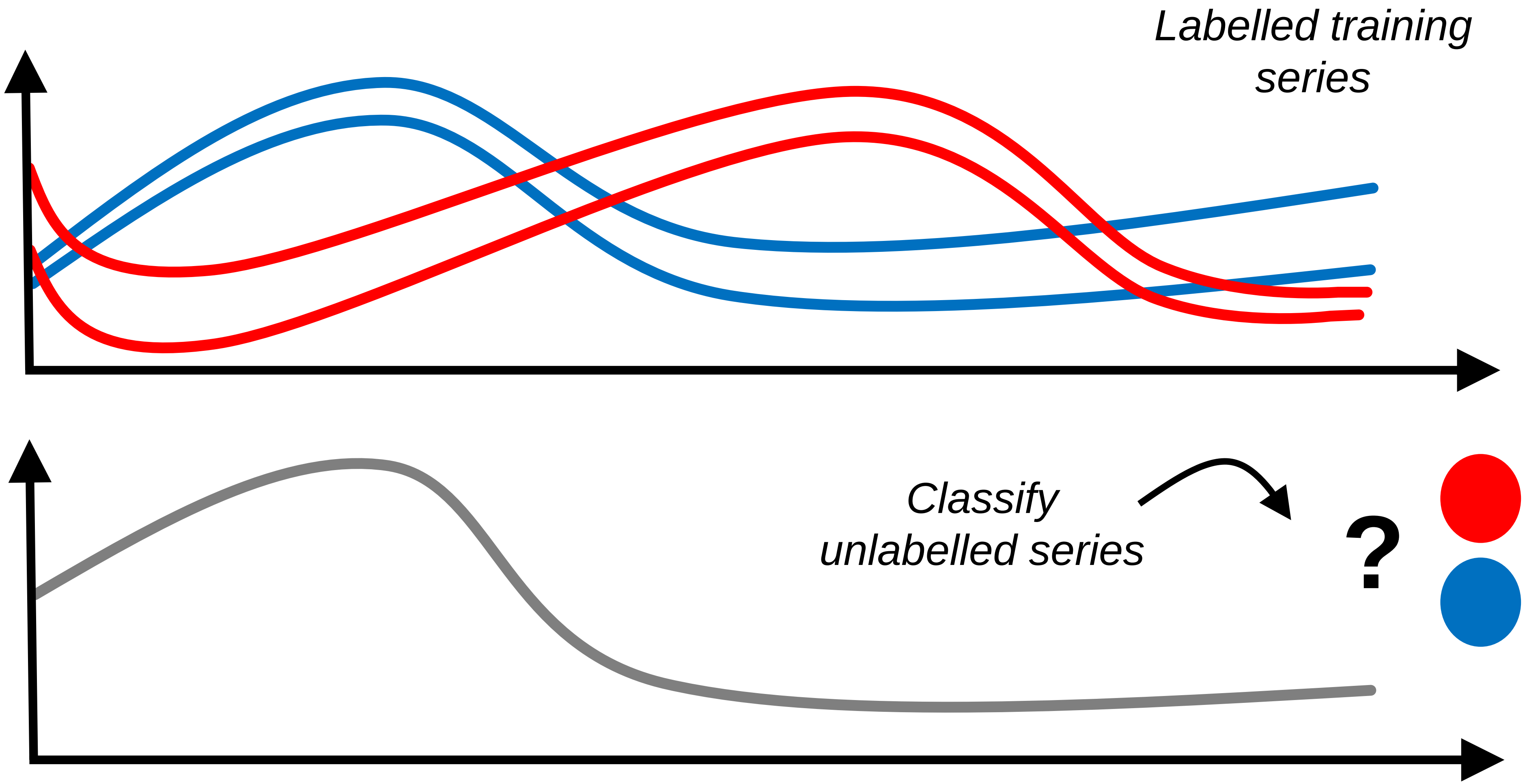






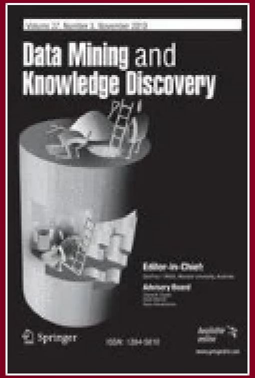


Time Series Classification (TSC)



Home > Data Mining and Knowledge Discovery > Article

The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances



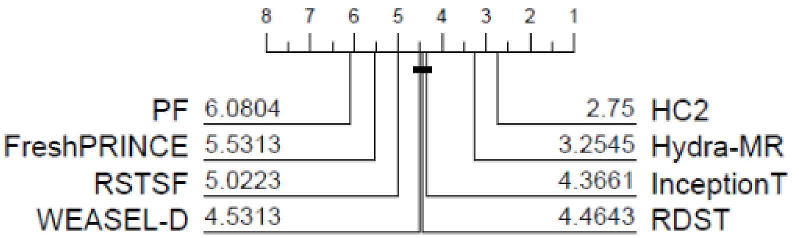
Open access | Published: 23 November 2016 | **31**, 606–660 (2017)

02-03-2019

Deep learning for time series classification: a review

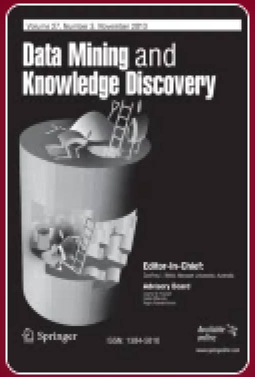
Authors: Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, Pierre-Alain Muller

Published in: Data Mining and Knowledge Discovery | Issue 4/2019



Home > Data Mining and Knowledge Discovery > Article

Bake off redux: a review and experimental evaluation of recent time series classification algorithms



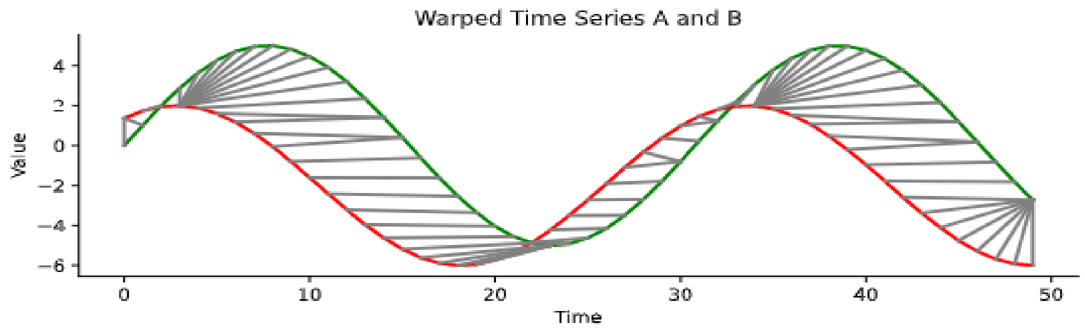
Open acce | Published: 19 April 2024

(2024) Cite this article

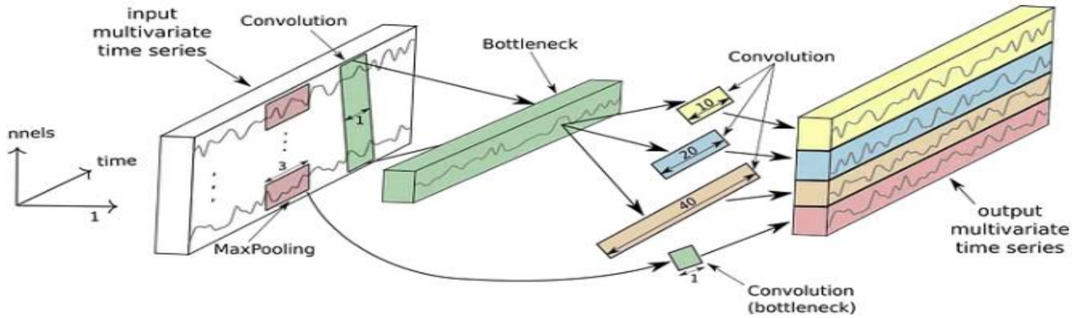
Data Mining a

Taxonomy of Time Series Classification Algorithms Part I

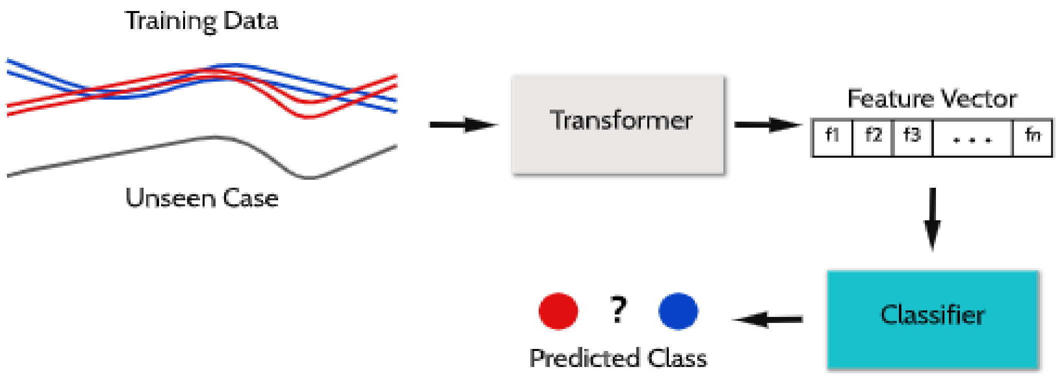
Distance based



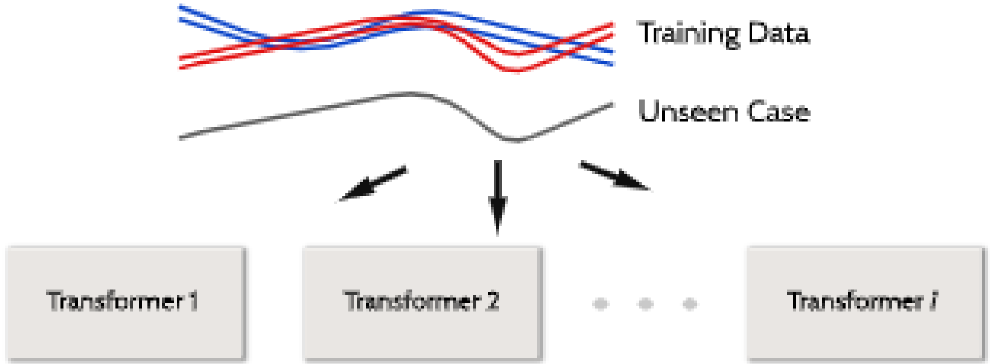
Deep learning



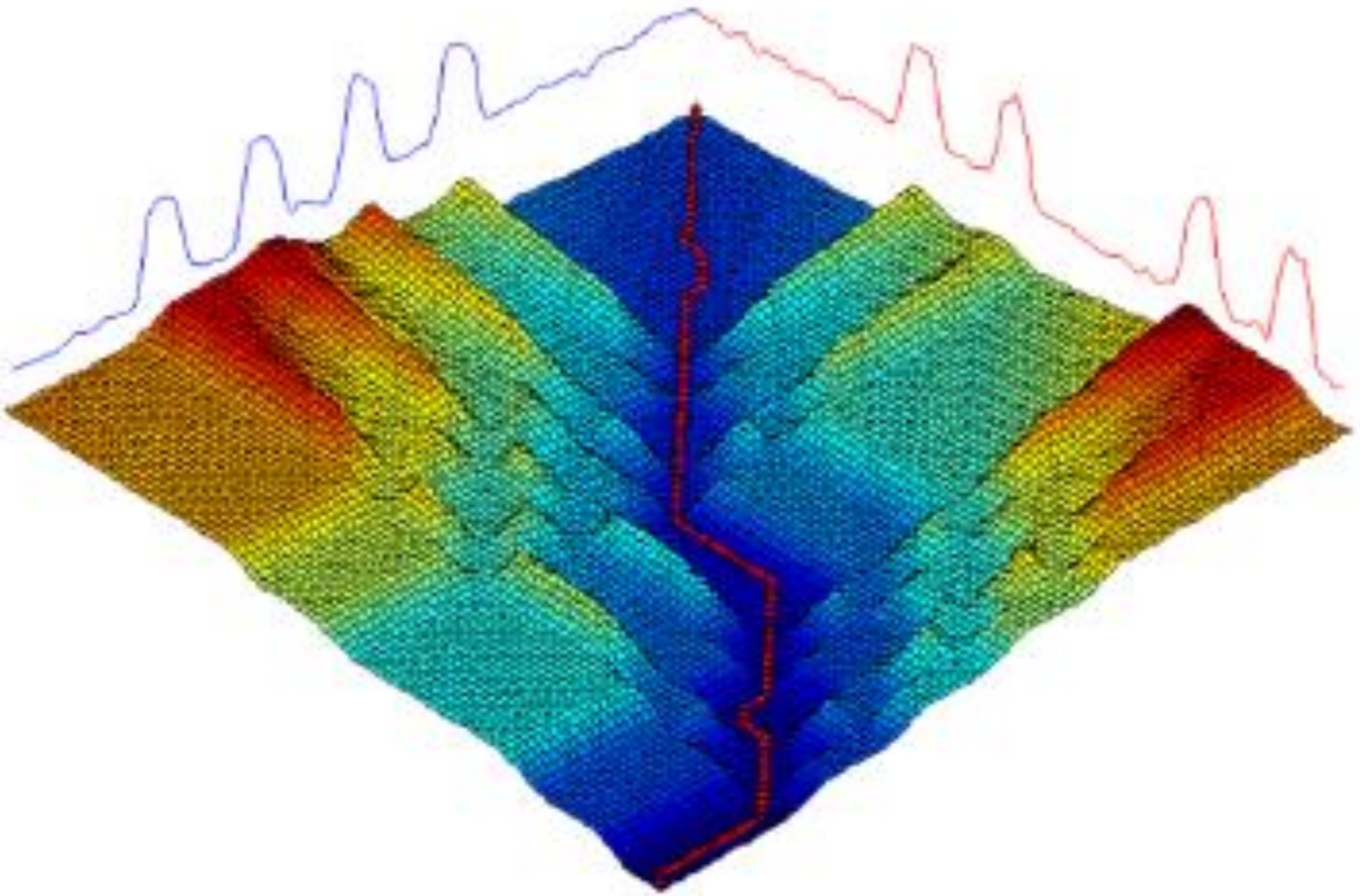
Feature based



Interval based



Distance Based Classifiers

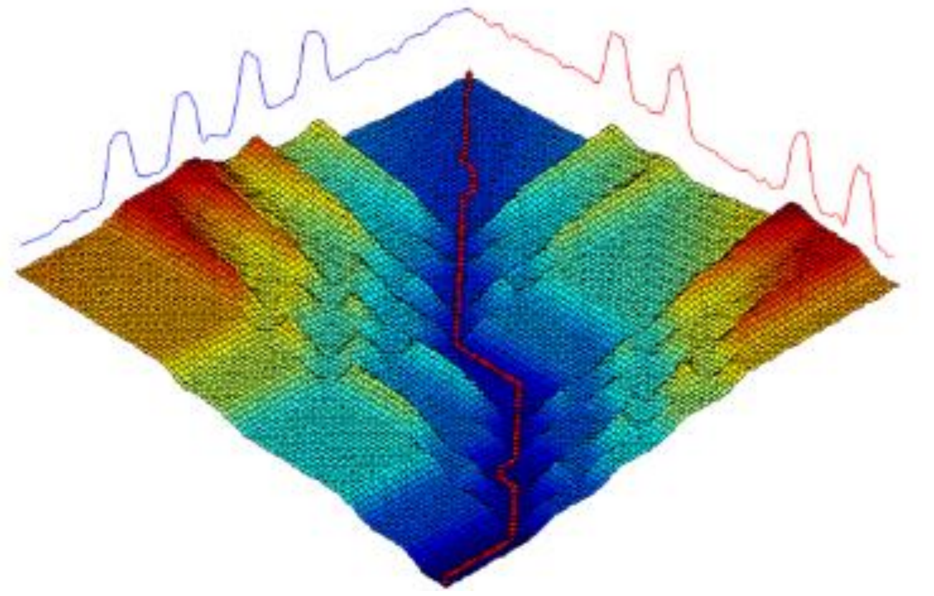


Distance Based Classifiers

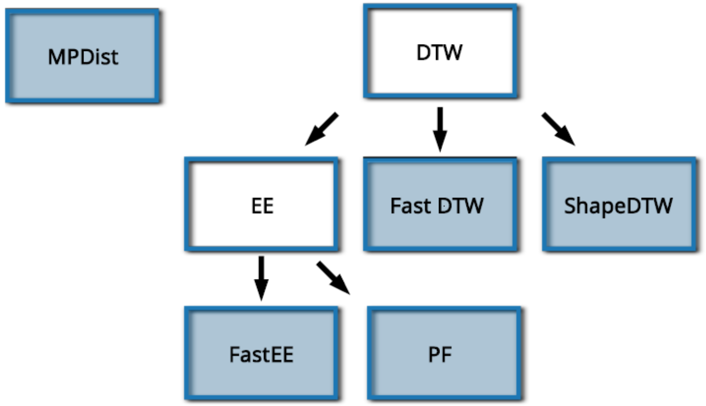
Algorithm 1 DTW ($\mathbf{a}, \mathbf{b}$, (*both series of length m*), w (*window proportion, default value* $w \leftarrow 1$), M (*pointwise distance matrix*))

Let C be an $(m+1) \times (m+1)$ matrix initialised to zero, indexed from zero.
for $i \leftarrow 1$ to m **do**
  for $j \leftarrow 1$ to m **do**
    if $|i - j| < w \cdot m$ **then**
      $C_{i,j} \leftarrow M_{i,j} + \min(C_{i-1,j-1}, C_{i-1,j}, C_{i,j-1})$
return $C_{m,m}$

DTW has no explicit penalty for moving off the diagonal.



More on distances tomorrow when we cover clustering



Distance Based

DTW	Dynamic Time Warping [Ratanamahatana and Keogh 2005]
Fast DTW	Fast Dynamic Time Warping [Tan et al. 2018]
EE	Elastic Ensemble [Lines and Bagnall 2015]
FastEE	Fast Elastic Ensemble [Oastler and Lines 2019]
PF	Proximity Forest [Lucas et al. 2019]
Shape DTW	Shape based Dynamic Time Warping [Zhao and Itti 2018]
MPDist	Matrix Profile Distance [Gharghabi et al. 2020]

Elastic Ensemble (EE)

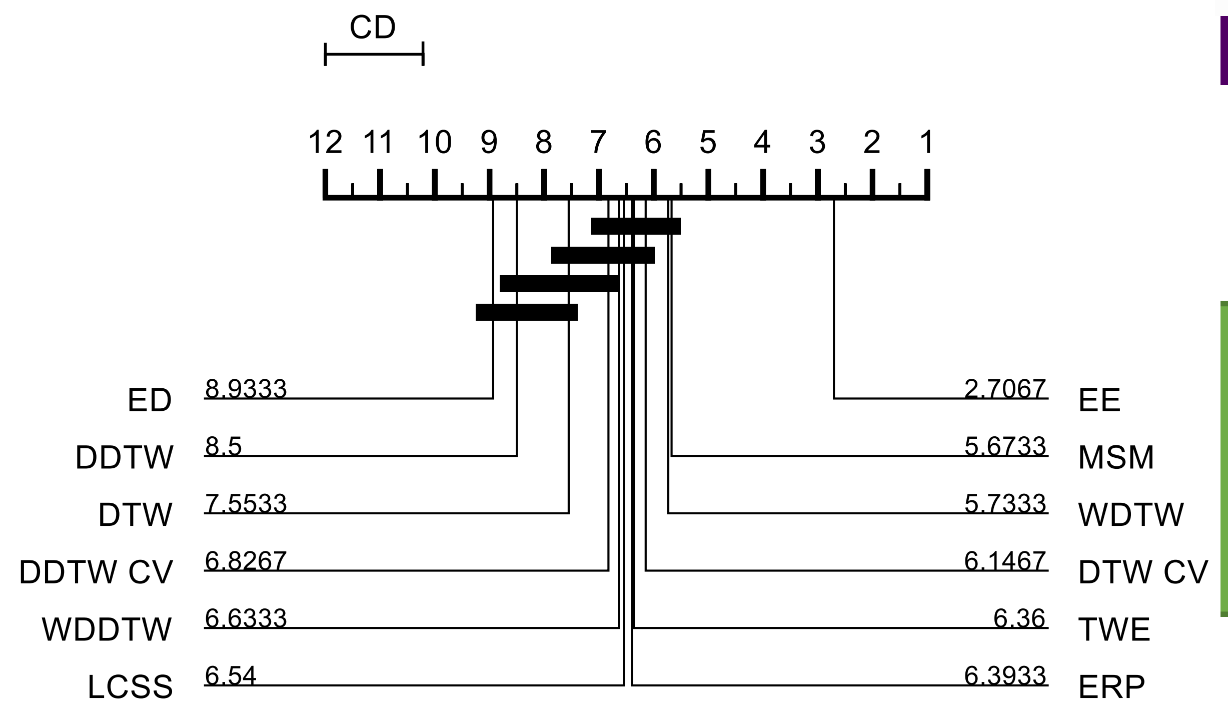
Published: 28 June 2014

Time series classification with ensembles of elastic distance measures

Jason Lines & Anthony Bagnall

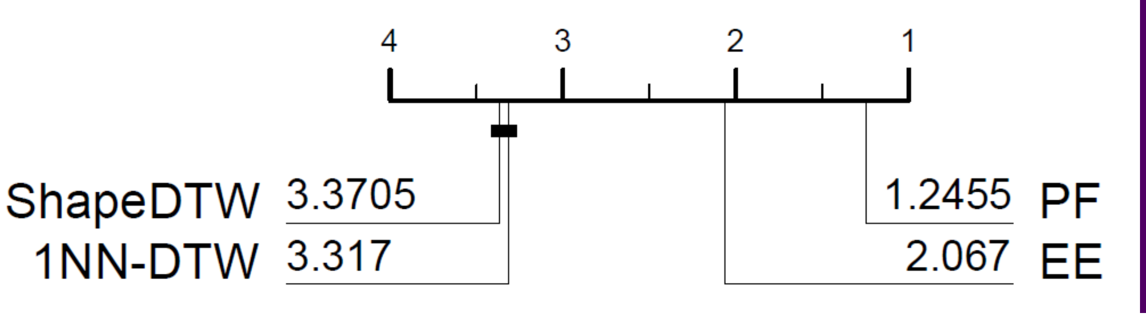
Data Mining and Knowledge Discovery **29**, 565–592 (2015) | Cite this article

Euclidean D
DTW (full)
DTW (cv wir
Derivati
Derivati
Weighte
Weighte
Longest
Edit Distanc
Move-Split-M
Time-warp Edit Distance



- Simple and transparent proportional voting scheme
- Constituents weighted by training cross-validation accuracy
- First reported TSC algorithm to **significantly outperform DTW** on the UCR datasets

Best in class: Proximity Forest (PF)



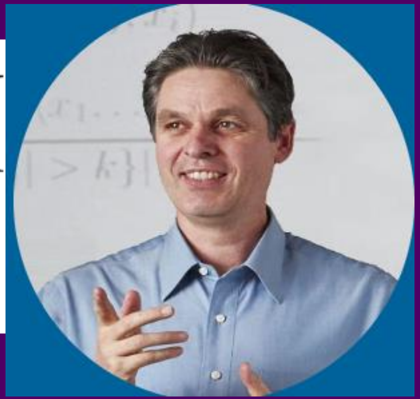
Published: 06 February 2019

Proximity Forest: an effective and scalable distance-based classifier for time series

Benjamin Lucas, Ahmed Shifaz, Charlotte Pelletier, Lachlan O'Neill, Nayyar Zaidi, Bart Goethals, François Petitjean & Geoffrey I. Webb

Data Mining and Knowledge Discovery **33**, 607–635 (2019) | Cite this article

MONASH University



BUT ....

Computer Science > Machine Learning

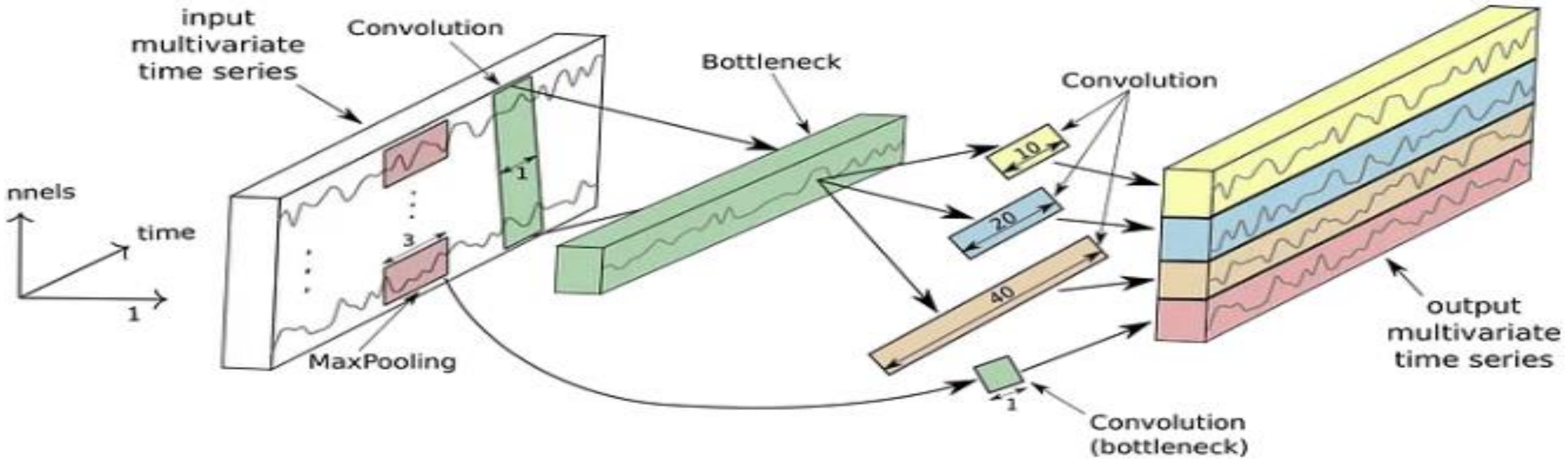
[Submitted on 12 Apr 2023 (v1), last revised 13 Apr 2023 (this version, v2)]

Proximity Forest 2.0: A new effective and scalable similarity-based classifier for time series

Matthieu Herrmann, Chang Wei Tan, Mahsa Salehi, Geoffrey I. Webb

We currently only have a Java implementation of PF, we would love to include it in aeon

Deep Learning Classifiers



Deep Learning Time Series Classifiers

Name	Year	Code
Disjoint-CNN	2021	y
Inception-FCN	2021	y
KDCTime	2022	n
Multi-Stage-Att	2020	n
CT_CAM	2020	n
CA-SFCN	2020	y
RTFN	2021	n
LAXCAT	2021	n
MACNN	2021	y
T2	2021	y
GTN	2021	y
TRANS	2021	n
FMLA	2022	n
AutoTransformer	2022	n
BENDER	2021	y
TST	2021	y
TARNET	2022	y

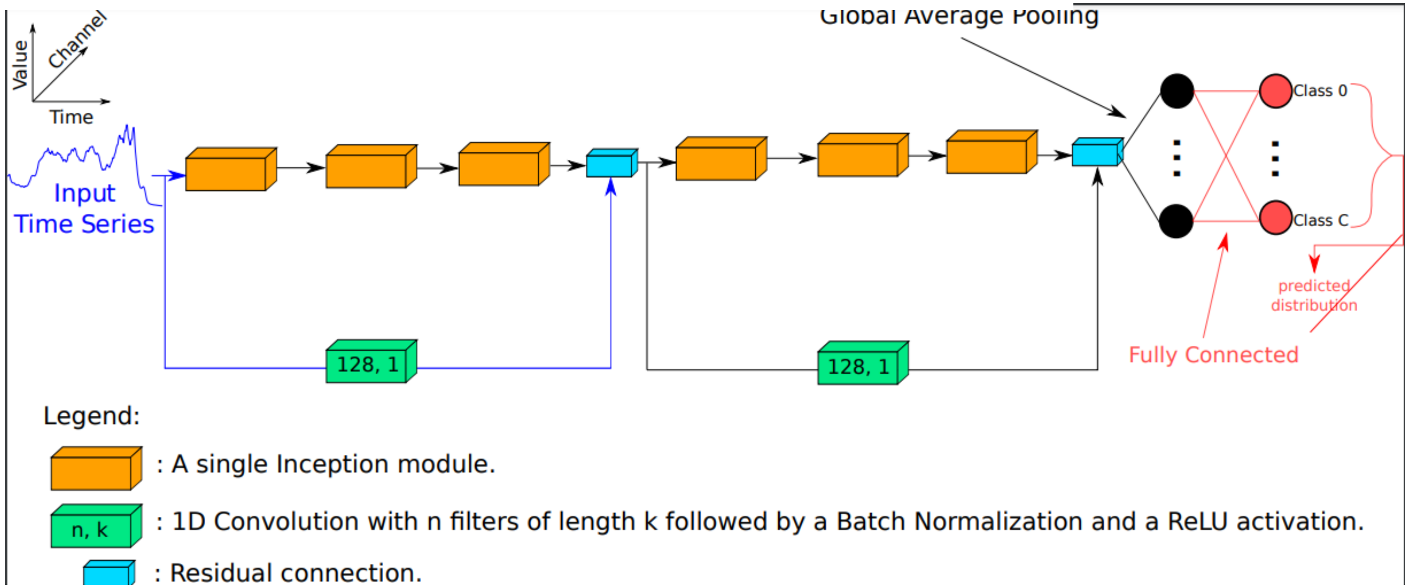
- There has been a huge international research effort to develop deep learners for TSC
- A recent survey references 246 papers, most of which have been published in the last three years.

[Submitted on 6 Feb 2023]

Deep Learning for Time Series Classification and Extrinsic Regression: A Current Survey

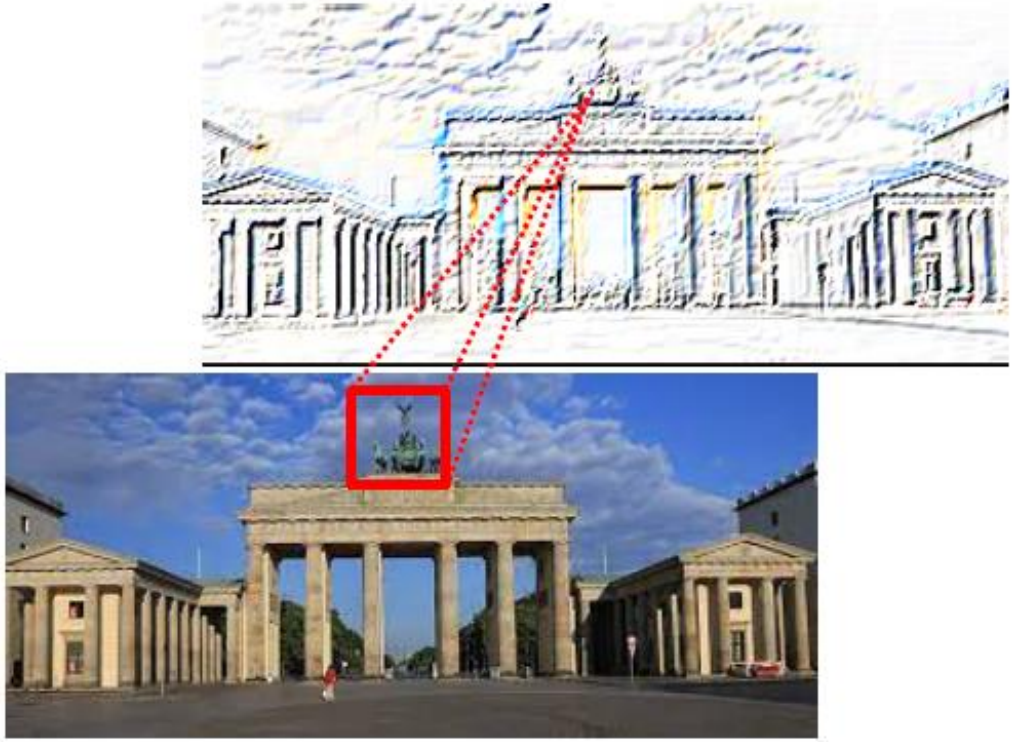
Navid Mohammadi Foumani, Lynn Miller, Chang Wei Tan, Geoffrey I. Webb, Germain Forestier, Mahsa Salehi

Generally poorly evaluated,not reproducible and self referential.

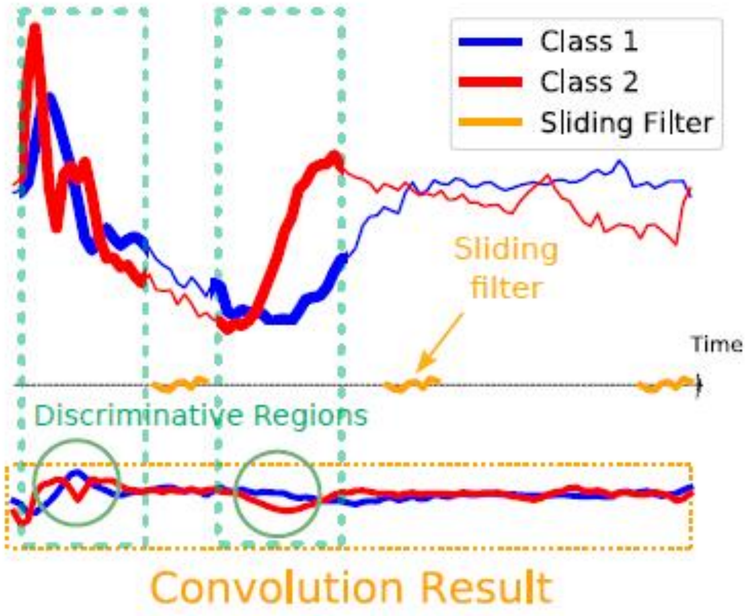


Convolutions

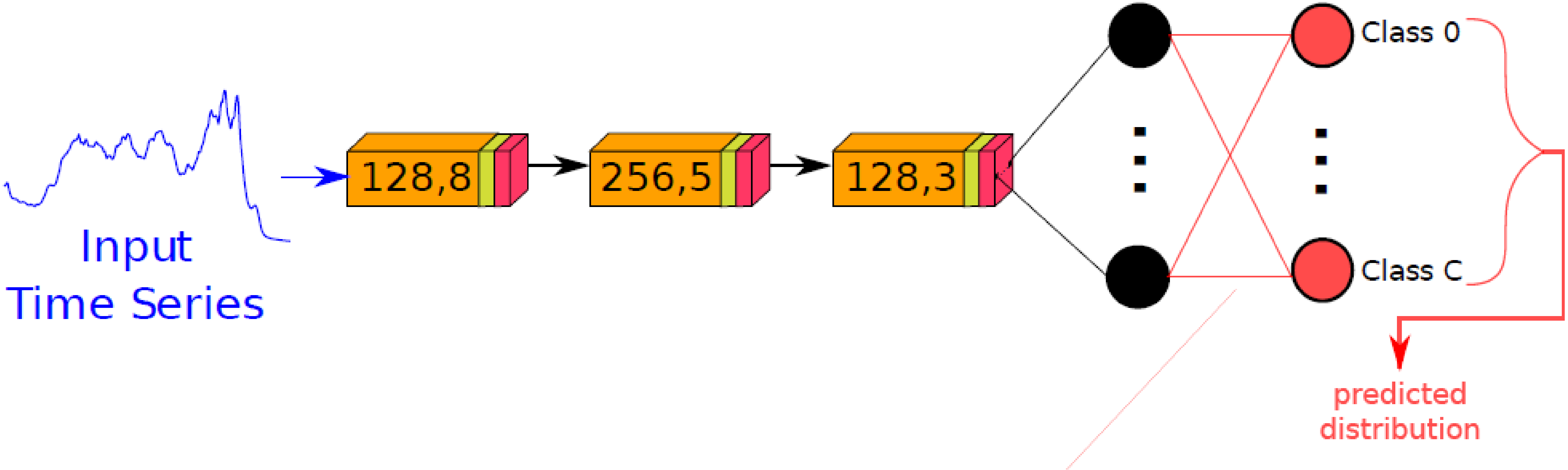
Convolutions on Images vs Time Series



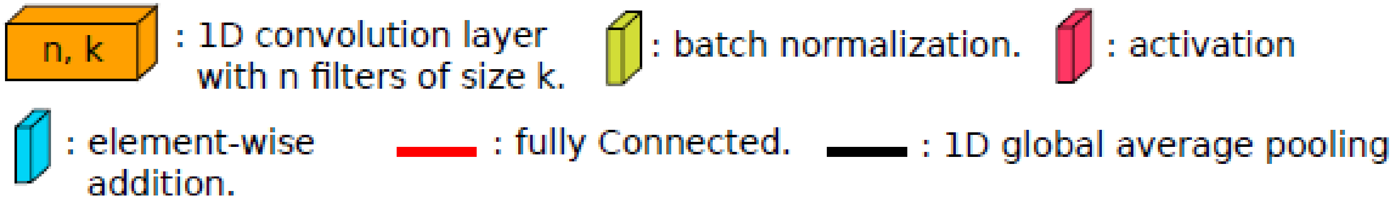
The result of a applying an edge detection convolution on an image



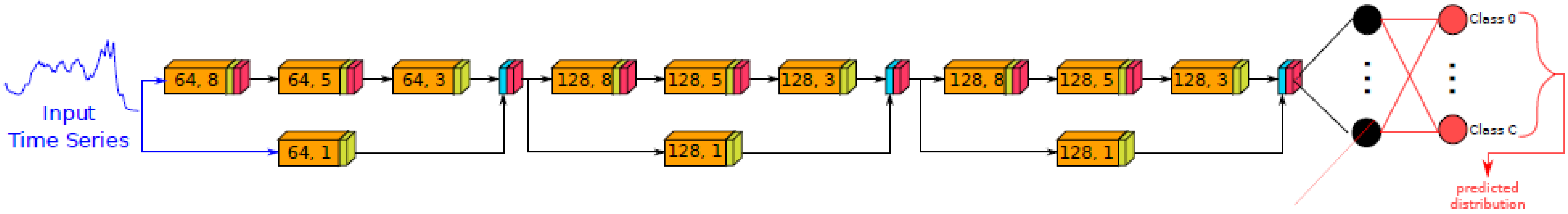
Fully Convolutional Networks (FCN)



Wang, Z., Yan, W., & Oates, T. (2017, May). Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International joint conference on neural networks (IJCNN) (pp. 1578-1585). IEEE.



Residual Network (ResNet)



Wang, Z., Yan, W., & Oates, T. (2017, May). Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International joint conference on neural networks (IJCNN) (pp. 1578-1585). IEEE.

Inception

- Originally proposed by Google for image recognition problems [1]
- Further developed to reach state-of-the-art results on ImageNet [2]

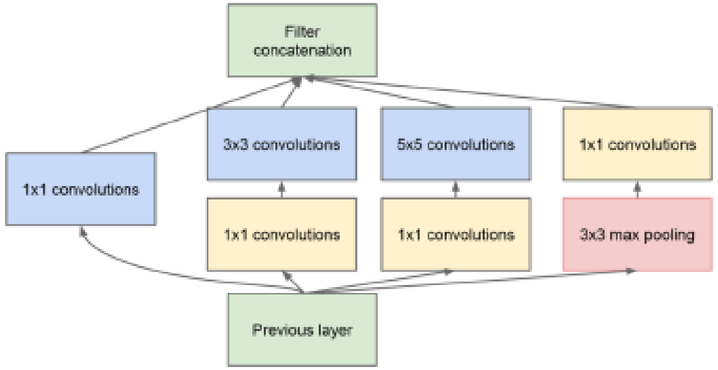
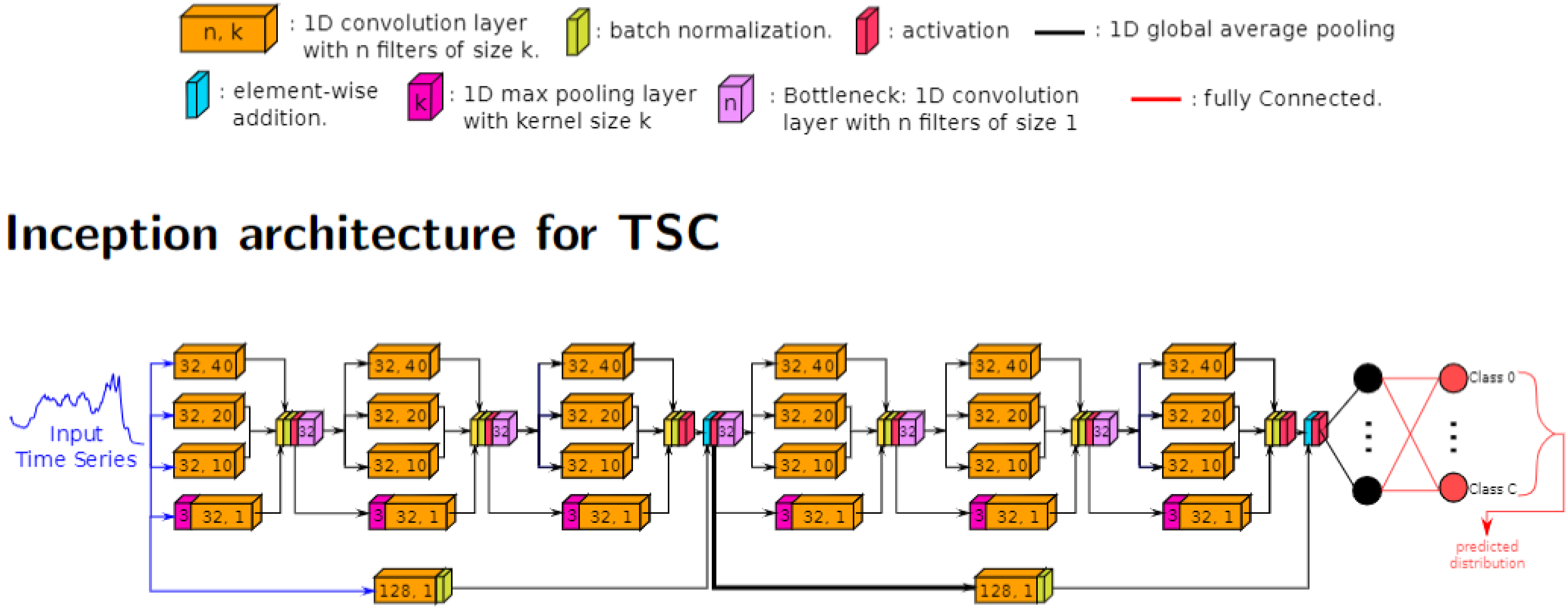


Figure: Inception module for image recognition [1]

[1] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-9).

[2] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2818-2826).

Inception Time

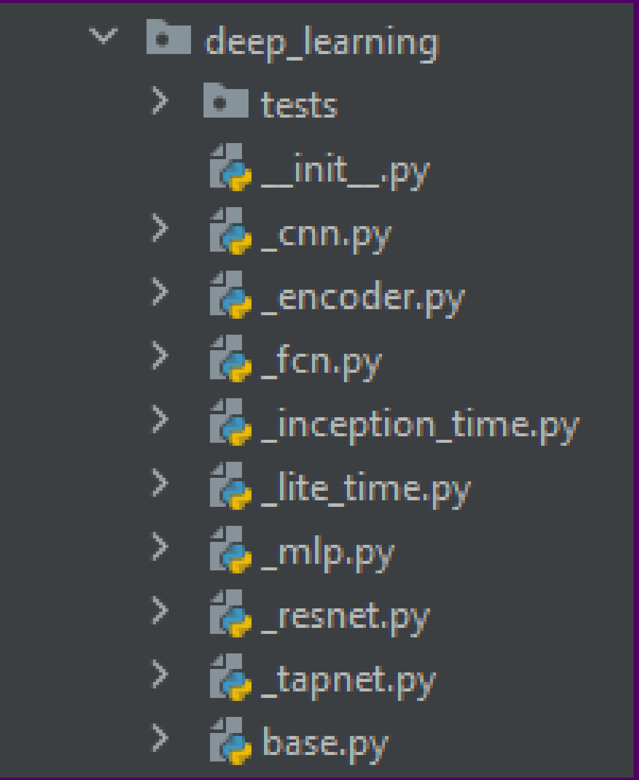


Ismail Fawaz, Hassan, et al. "Inceptiontime: Finding alexnet for time series classification." Data Mining and Knowledge Discovery 34.6 (2020): 1936-1962.

Inception Time

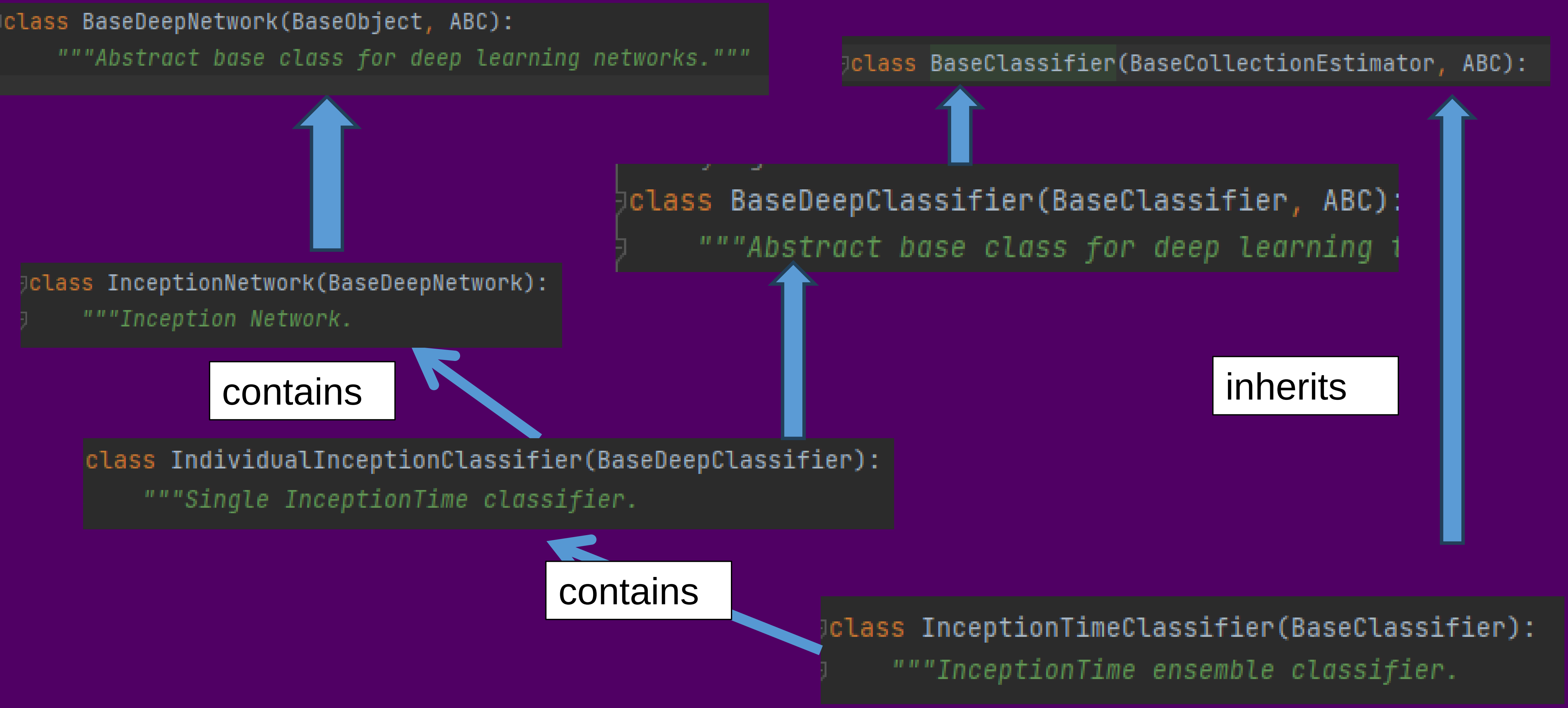
- Inception time is an ensemble of five inception modules
- An inception module is built up of convolutional layers
- It apply convolutions of different resolutions to capture various patterns
- Use a bottleneck layer in order to reduce the number of parameters

Inception Time Implementation




```python
from aeon.datasets import load_basic_motions, load_gunpoint
from aeon.classification.deep_learning import InceptionTimeClassifier
from sklearn.metrics import accuracy_score
X_train, y_train = load_basic_motions(split="train")
X_test, y_test = load_basic_motions(split="test")

clf = InceptionTimeClassifier()
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print("accuracy =",accuracy_score(y_test, y_pred))
```

Inception Time Implementation



Inception Time inception module

```python
for i in range(len(kernel_size_s)):
    conv_list.append(
        tf.keras.layers.Conv1D(
            filters=n_filters,
            kernel_size=kernel_size_s[i],
            strides=strides,
            dilation_rate=dilation_rate,
            padding=padding,
            activation="linear",
            use_bias=use_bias,
        )(input_inception)
    )

if use_max_pooling:
    max_pool_1 = tf.keras.layers.MaxPool1D(
        pool_size=max_pool_size, strides=strides, padding=padding
    )(input_tensor)

    conv_max_pool = tf.keras.layers.Conv1D(
        filters=n_filters,
        kernel_size=1,
        padding=padding,
        activation="linear",
        use_bias=use_bias,
    )(max_pool_1)

    conv_list.append(conv_max_pool)
```

```python
x = tf.keras.layers.Concatenate(axis=2)(conv_list)
x = tf.keras.layers.BatchNormalization()(x)
x = tf.keras.layers.Activation(activation=activation)(x)
```

```python
Ali El Hadi ISMAIL FAWAZ +1
def build_network(self, input_shape, **kwargs):
    """
    Construct a network and return its input and output layers.

    input_shape : tuple
        The shape of the data fed into the input layer

    Returns
    -------
    input_layer : a keras layer
    output_layer : a keras layer
    """
```

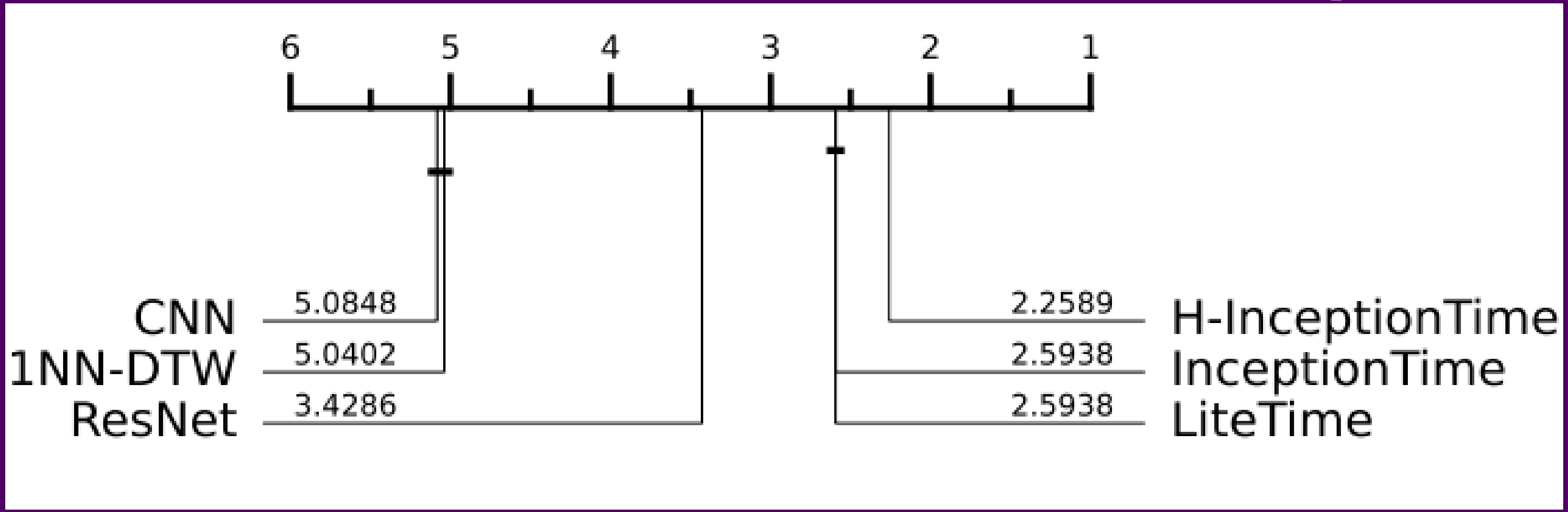
Inception Time Classifier

Ali El Hadi ISMAIL FAWAZ +3

```python
def _fit(self, X, y):
    """
    Fit the classifier on the training set (X, y).

    Parameters
    ----------
    X : np.ndarray
        The training input samples of,
        shape (n_cases, n_channels, n_timepoints).
        If a 2D array-like is passed, n_channels is assumed to be 1.
    y : np.ndarray
        The training data class labels of shape (n_cases,).


    Returns
    -------
    self : object
    """
```

```python
def build_model(self, input_shape, n_classes, **kwargs):
    """
    Construct a compiled, un-trained, keras model that is ready for training.

    Parameters
    ----------
    input_shape : tuple
        The shape of the data fed into the input layer
    n_classes: int
        The number of classes, which shall become the size of the output
         layer

    Returns
    -------
    output : a compiled Keras Model
    """
```

Best in class: H-InceptionTime



InceptionTime is an ensemble of inception based deep learners

H-InceptionTime adds some time series specific features

UNIVERSITÉ HAUTE-ALSACE

Published: 07 September 2020

InceptionTime: Finding AlexNet for time series classification

Hassan Ismail Fawaz, Benjamin Lucas, Germain Forestier, Charlotte Pelletier, Daniel F. Schmidt, Jonathan Weber, Geoffrey I. Webb, Lhassane Idoumghar, Pierre-Alain Muller & François Petitjean

Data Mining and Knowledge Discovery **34**, 1936–1962 (2020) | Cite this article

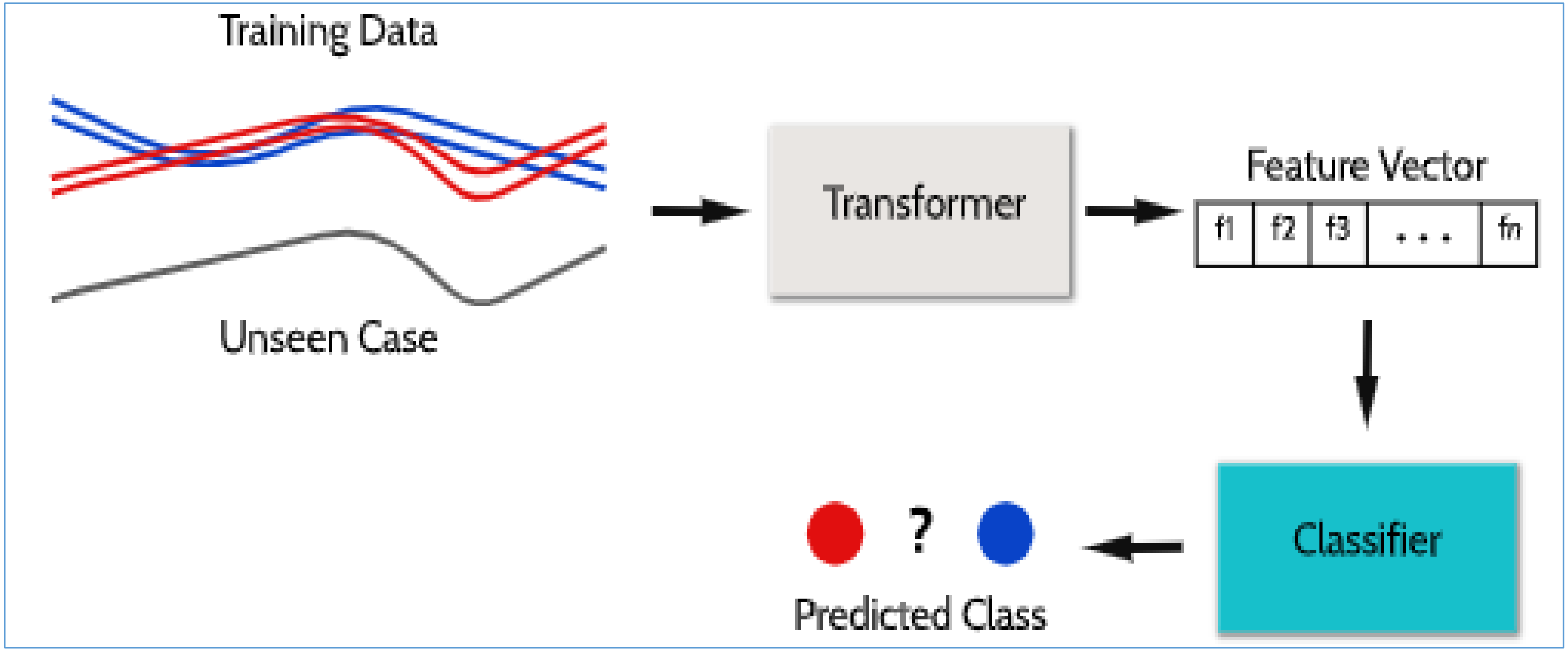
Deep Learning For Time Series Classification Using New Hand-Crafted Convolution Filters

Publisher: IEEE Cite This PDF

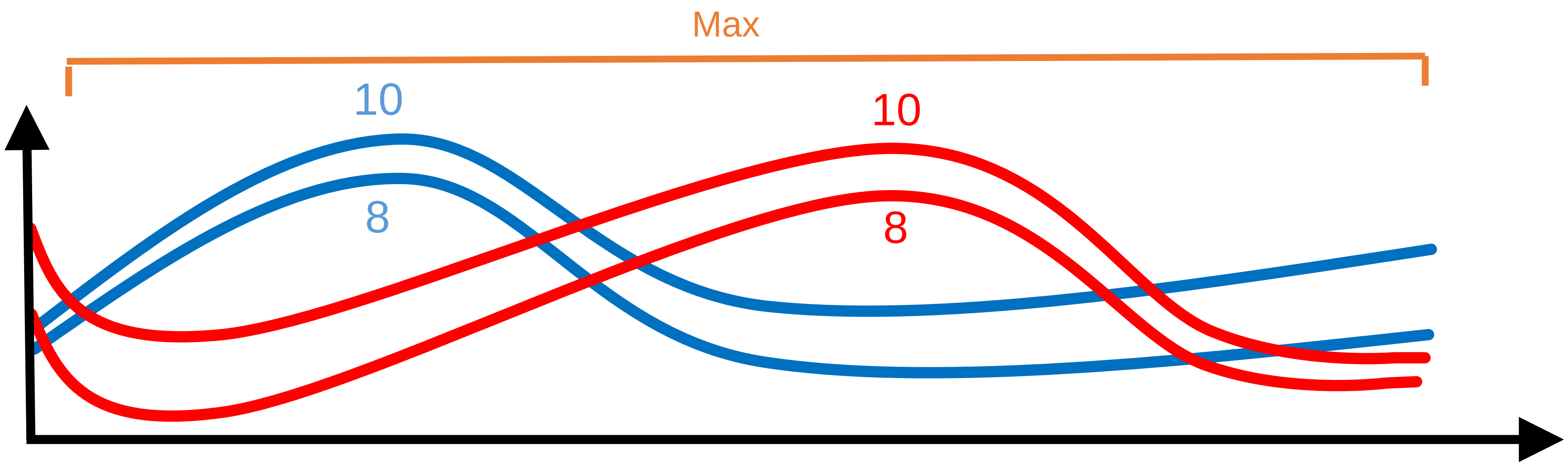
Published in Big Data 2022

Ali Ismail-Fawaz ; Maxime Devanne ; Jonathan Weber ; Germain Forestier **All Authors**

Feature Based Pipelines



Time Series Feature Extraction



Max	Min	…	Mean	*Class*
10	3	…	5	*Blue*
8	2	…	4	*Blue*
10	2	…	5	*Red*
8	1	…	4	*Red*

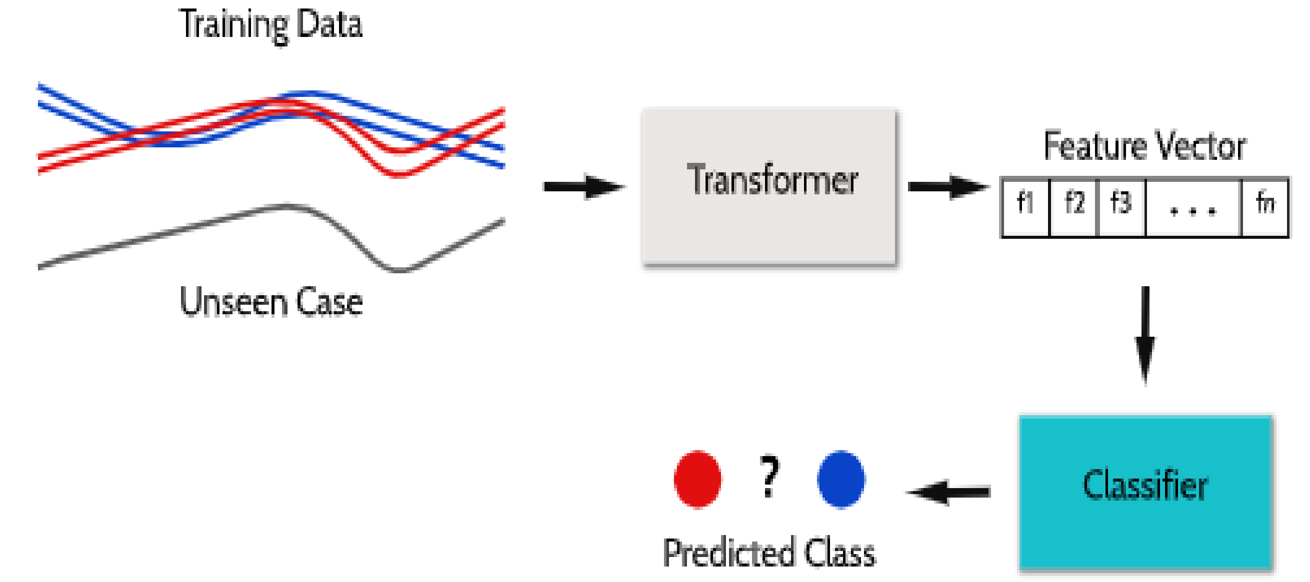
Feature Based Pipelines

The simplest transformation based approach is to create summary features then use a standard classifier

There are many toolkits that create summary features

tsfresh

MatrixProfile



hctsa

THE UNIVERSITY OF SYDNEY

catch22

Explanatory Transformations	
hctsa	Highly Comparative Time-series Analysis [Fulcher and Jones 2017]
catch22	Canonical Time Series Characteristics [Lubba et al. 2019]
tsfresh	TS Feature Extraction based on Scalable Hypothesis Tests [Christ et al. 2018]
FreshPRINCE	Fresh Pipeline with Rotation Forest Classifier [Middlehurst and Bagnall 2022]
Signatures	Generalised Rough Path Signatures [Morrill et al. 2020]
MP	Matrix Profile Transform [Yeh et al. 2018]

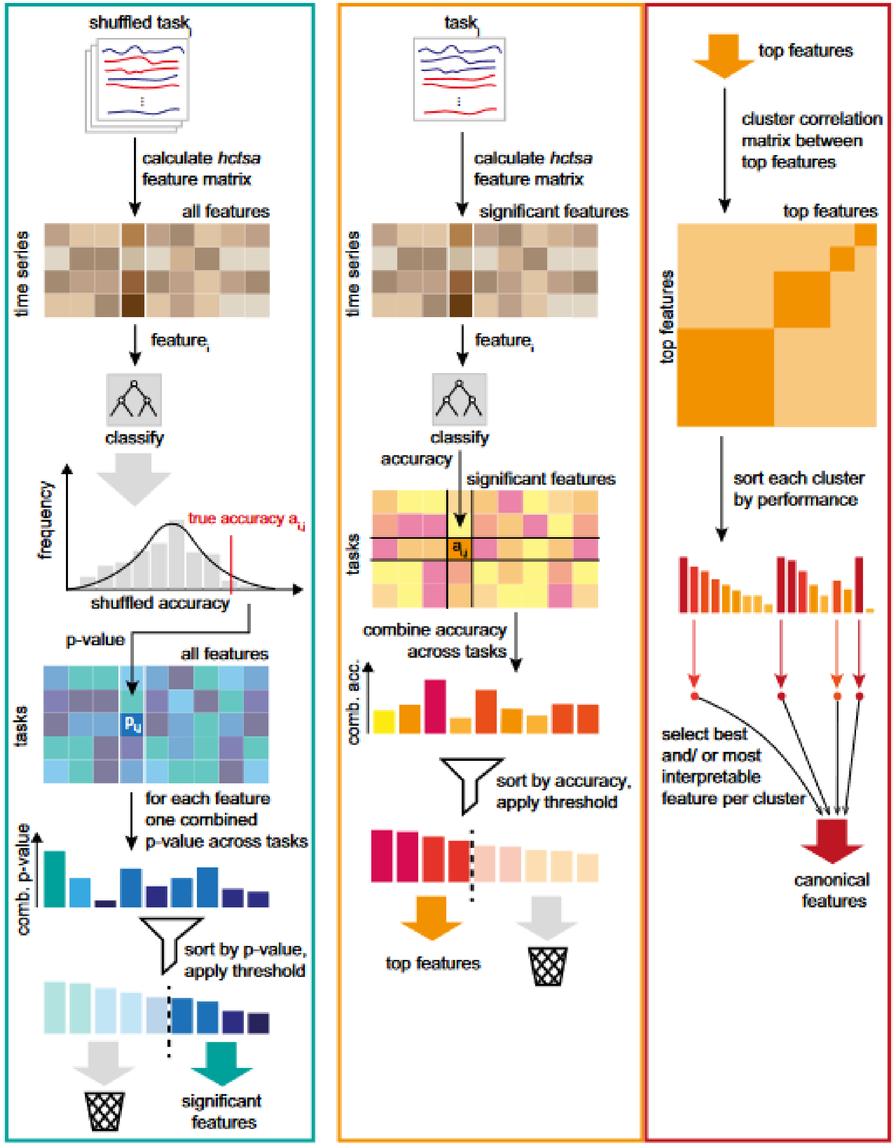
Summary Statistic Collections

Highly Comparative Time-Series Analysis (hctsa)
Over 7700 features.
Implemented in MATLAB.
https://github.com/benfulcher/hctsa

CAnonical Time-series Characteristics (Catch22)
22 features.
Implemented in C. Subset of hctsa.
https://github.com/chlubba/catch22

Time Series Feature extraction based on scalable hypothesis tests (TSFresh)
Just under 800 features.
Implemented in Python.
https://github.com/blue-yonder/tsfresh

Canonical Time-series Characteristics (catch22)



7658 features filtered down to 22 using a 3 step process:

1. Remove features that perform worse than random chance

2. Sort by balanced accuracy, remove using a threshold

3. Cluster remaining features, select a feature from each cluster for accuracy, speed and interpretability

Figure of the catch22 filtering process from: Lubba, Carl H., et al. "catch22: CAnonical Time-series CHaracteristics." *Data Mining and Knowledge Discovery* 33.6 (2019): 1821-1852.

Examples of Complex Summary Statistic

- **SB_BinaryStats_mean_longstretch1** - Longest period of consecutive values above the mean.

- **Linear_trend_timewise** - Calculate a linear least-squares regression for the values of the time series versus the sequence from 0 to length of the time series minus one.

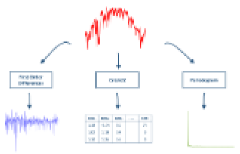
- **SP_Summaries_welch_rect_area_5_1** - Total power in lowest fifth of frequencies in the Fourier power spectrum.

- **Matrix_profile_mean** – Calculate the 1-D matrix profile and return the mean of it.

aeon feature based transformers

https://www.aeon-toolkit.org/en/stable/examples.html

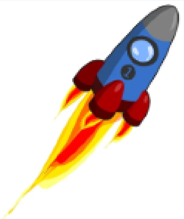
Transformation



Overview of Transformations

TSFresh transform

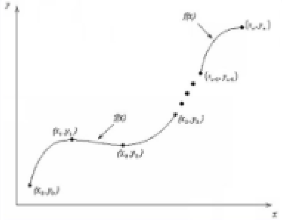
Catch22 transform



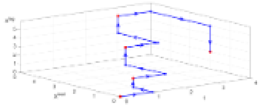
Rocket transform

MiniRocket transform

SAST transform



Interpolation



Signature method

The Canonical Time-series Characteristics (catch22) transform

catch22[1] is a collection of 22 time series features extracted from the 7000+ present in the *hctsa* [2][3] toolbox. A hierarchical clustering was performed on the correlation matrix of features that performed better than random chance to remove redundancy. These clusters were sorted by balanced accuracy using a decision tree classifier and a single feature was selected from the 22 clusters formed, taking into account balanced accuracy results, computational efficiency and interpretability.

In this notebook, we will demonstrate how to use the catch22 transformer on the ItalyPowerDemand univariate and BasicMotions multivariate datasets. We also show catch22 used for classification with a random forest classifier.

References:

[1] Lubba, C. H., Sethi, S. S., Knaute, P., Schultz, S. R., Fulcher, B. D., & Jones, N. S. (2019). catch22: CAnonical Time-series CHaracteristics. Data Mining and Knowledge Discovery, 33(6), 1821-1852.

[2] Fulcher, B. D., & Jones, N. S. (2017). hctsa: A computational framework for automated time-series phenotyping using massive feature extraction. Cell systems, 5(5), 527-531.

[3] Fulcher, B. D., Little, M. A., & Jones, N. S. (2013). Highly comparative time-series analysis: the empirical structure of time series and their methods. Journal of the Royal Society Interface, 10(83), 20130048.

1. Imports

[18]:
```
from sklearn import metrics

from aeon.classification.feature_based import Catch22Classifier
from aeon.datasets import load_basic_motions, load_italy_power_demand
from aeon.transformations.collection.feature_based import Catch22
```

2. Load data

[19]:
```
IPD_X_train, IPD_y_train = load_italy_power_demand(split="train")
IPD_X_test, IPD_y_test = load_italy_power_demand(split="test")
IPD_X_test = IPD_X_test[:50]
IPD_y_test = IPD_y_test[:50]

print(IPD_X_train.shape, IPD_y_train.shape, IPD_X_test.shape, IPD_y_test.shape)

BM_X_train, BM_y_train = load_basic_motions(split="train")
BM_X_test, BM_y_test = load_basic_motions(
    split="test",
)
```

Using Transformers

```python
import numpy as np
from aeon.transformations.collection.feature_based import TSFreshFeatureExtractor, \
    Catch22, TSFreshRelevantFeatureExtractor, SevenNumberSummaryTransformer
# 50 univariate time series with and 300 time points
X = np.random.random((50, 1, 300))
ts_fresh = TSFreshFeatureExtractor()
c22 = Catch22()
stats = SevenNumberSummaryTransformer()
catch22 = Catch22()
X2 = ts_fresh.fit_transform(X)
X3 = c22.fit_transform(X)
X4 = stats.fit_transform(X)
print(X2.shape)
print(X3.shape)
print(X4.shape)
```

We store collections of time series in numpy arrays shape **`(n_cases,n_channels,n_timepoints)`**

```
Python 3.9.9 (tags/v3.9.9:ccb0e6a,
(50, 777)
(50, 22)
(50, 7)
```

Classification pipelines

You can use these transformers with an sklearn pipeline and sklearn classifier

```
from sklearn.pipeline import make_pipeline
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

from aeon.datasets import load_italy_power_demand
pipe= make_pipeline(TSFreshFeatureExtractor(),
RandomForestClassifier())
trainX, trainy= load_italy_power_demand(split="train")
testX, testy= load_italy_power_demand(split="test")
pipe.fit(trainX, trainy)
pred = pipe.predict(testX)
print("accuracy =",accuracy_score(testy, pred))
```

```
accuracy = 0.9582118561710399
```

Home > Pattern Recognition and Artificial Intelligence > Conference paper

The FreshPRINCE: A Simple Transformation Based Pipeline Time Series Classifier

Conference paper | First Online: 29 May 2022

pp 150–161 | Cite this conference paper

Compared six different transformers with three different classifiers on the 112 UCR data

Transformers: Summary, PCA, Signature, RandomInterval, Catch22, TSFresh

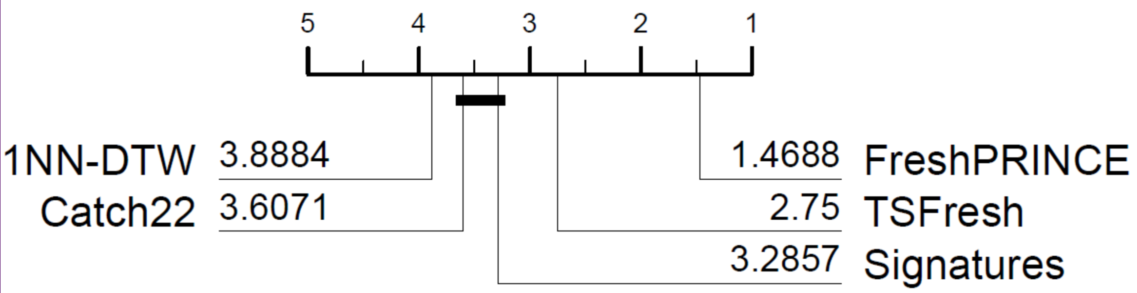
Classifiers:

- Rotation Forest (RotF) – Base classifier of STC, generally good with numeric data.
- Cross-validated Ridge Regression (RidgeCV) – commonly used base classifier.
- XGBoost – popular in certain domains

Best performing was

```
from aeon.transformations.collection.feature_based import TSFreshFeatureExtractor
from aeon.classification.sklearn import RotationForestClassifier
pipe = make_pipeline(TSFreshFeatureExtractor(), RotationForestClassifier())
```

Best in class: FreshPRINCE



The FreshPRINCE: A Simple Transformation Based Pipeline Time Series Classifier

Authors: Matthew Middlehurst, Anthony Bagnall Authors Info & Claims

Pattern Recognition and Artificial Intelligence: Third International Conference, ICPRAI 2022, Paris, France, June 1–3, 2022, Proceedings, Part II • Jun 2022 • Pages 150–161 • https://doi.org/10.1007/978-3-031-09282-4_13

The FreshPRINCE is a pipeline classifier combining TSFresh and the RotationForest classifier (FreshPRINCE).

BUT ....

orange™

arXiv > cs > arXiv:2308.01071

Computer Science > Machine Learning

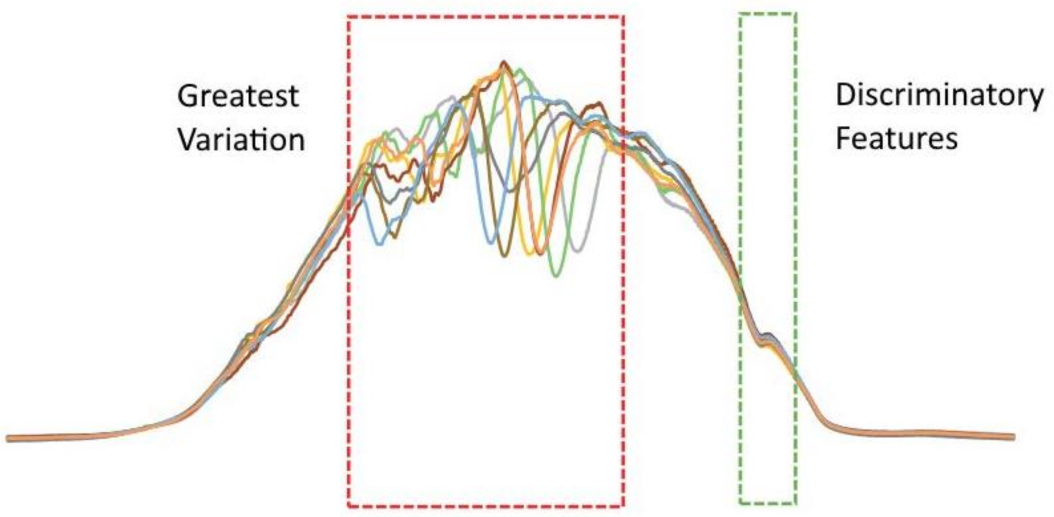
[Submitted on 2 Aug 2023]

Automatic Feature Engineering for Time Series Classification: Evaluation and Discussion

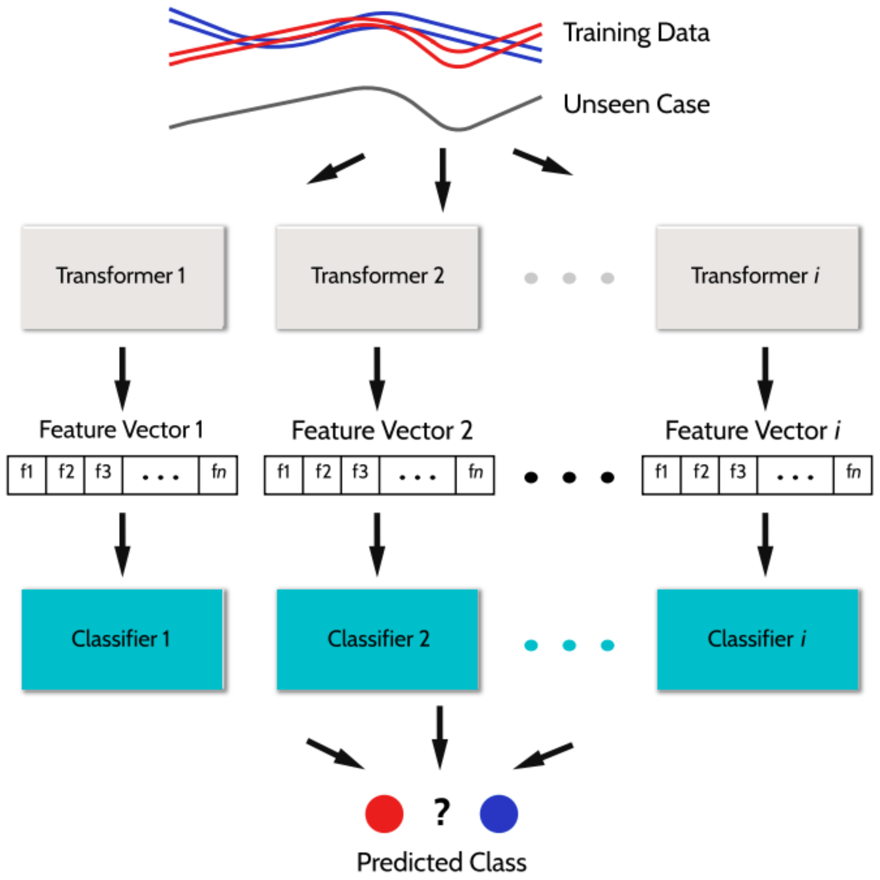
Aurélien Renault, Alexis Bondu, Vincent Lemaire, Dominique Gay

Published in IJCNN in 2023

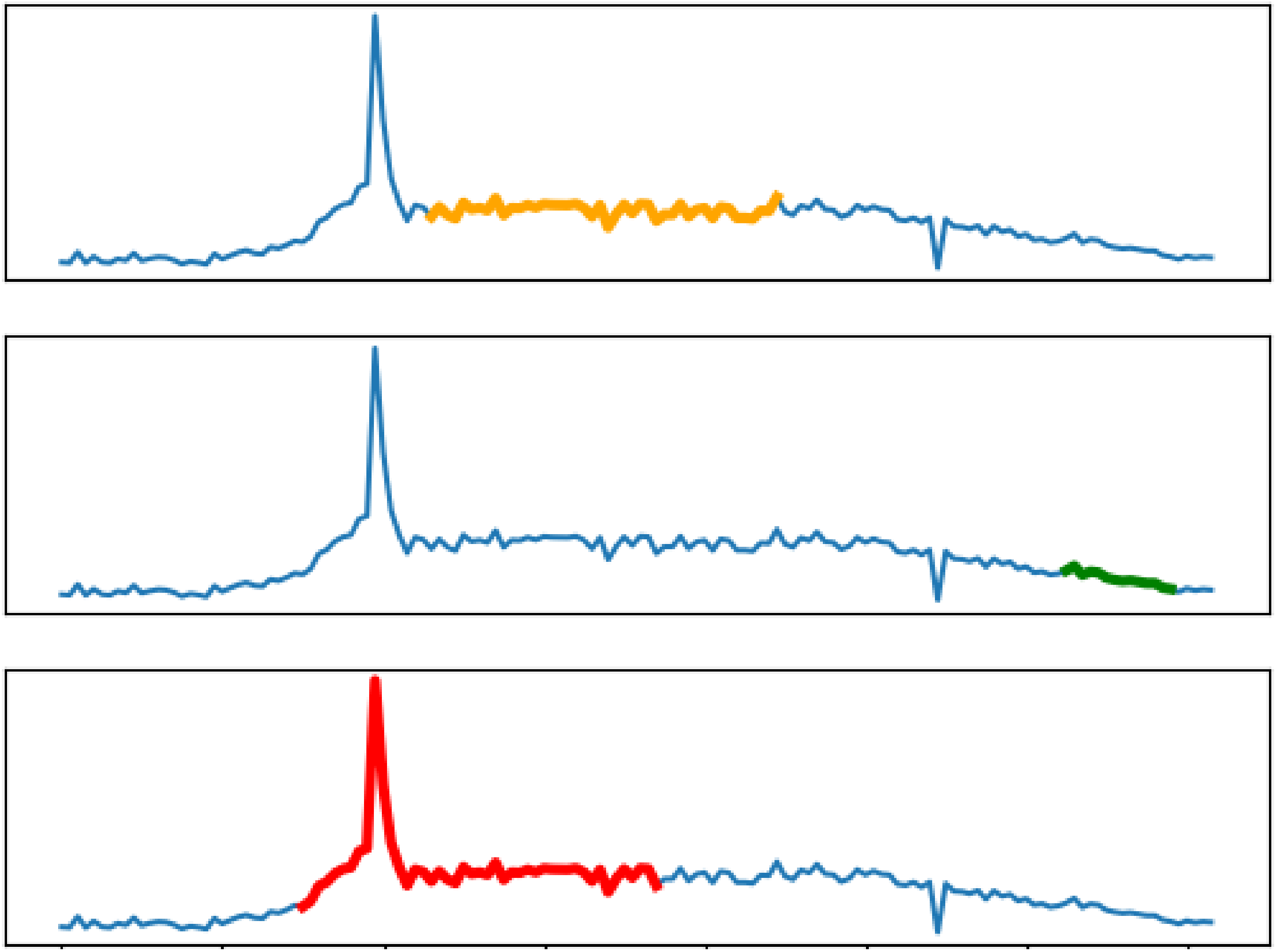
Interval Based Ensembles



Features measured over the whole series can get confounded by noise in irrelevant sections



Interval Based Approaches



Interval based classifiers attempt to overcome this problem by taking random intervals
The same intervals are used for all time series, meaning the algorithms are still **phase dependent**

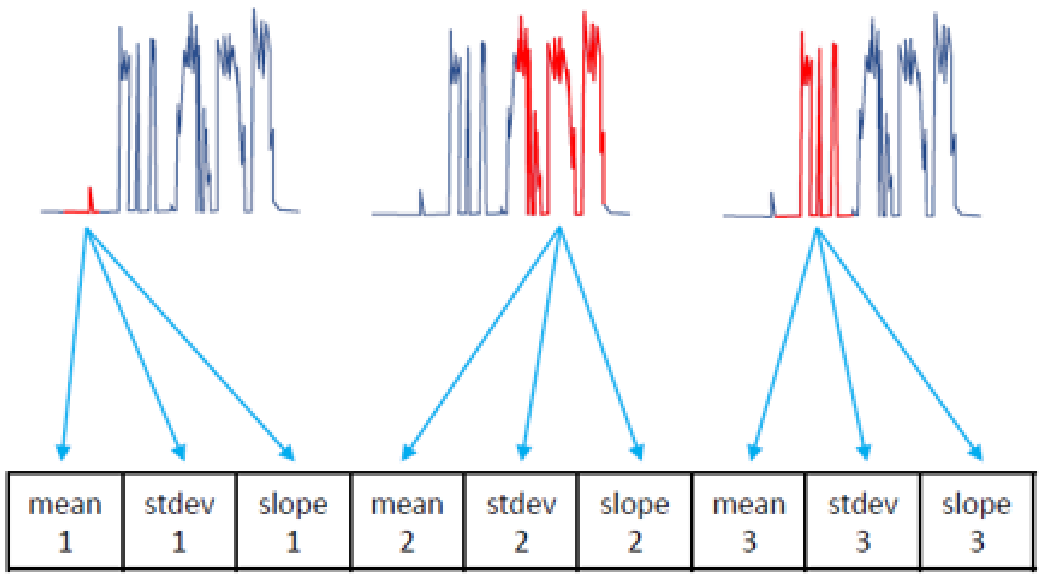
m(m-1)/2 possible intervals

Time Series Forest (TSF): Deng et al. 2013

The first interval based classifier based on random forest tree ensemble.

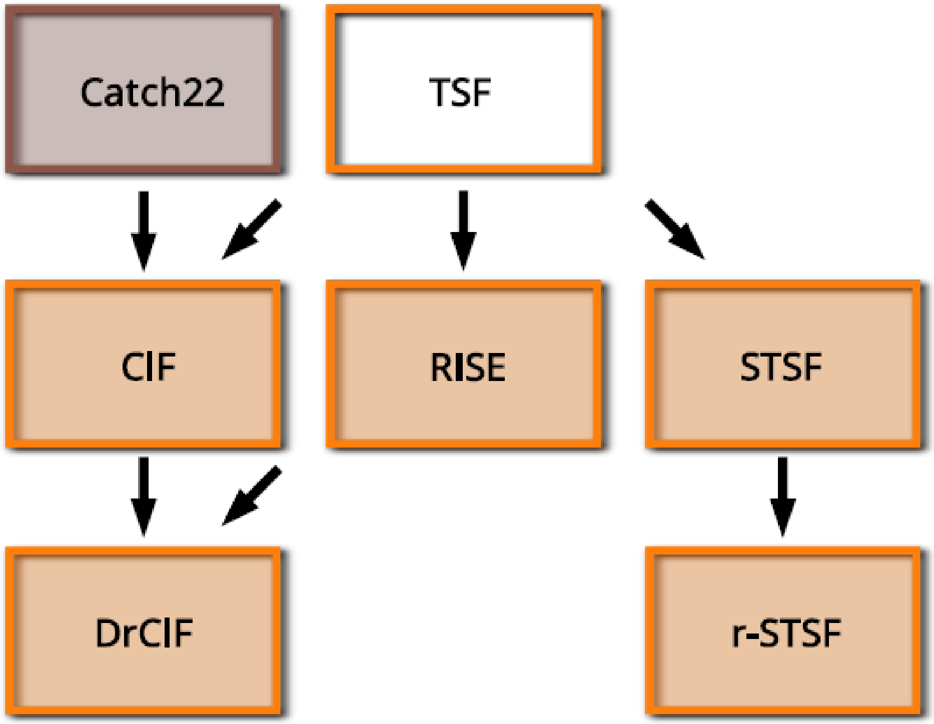
For each member of the ensemble:

1. Randomly select *k* intervals that apply to each time series
2. Find three summary statistics (mean, deviation and slope) for each interval and concatenate
3. Train a simple decision tree



Each tree has a different set of random intervals

Interval Based Ensembles



Conferences > 2020 IEEE International Confe...

The Canonical Interval Forest (CIF) Classifier for Time Series Classification

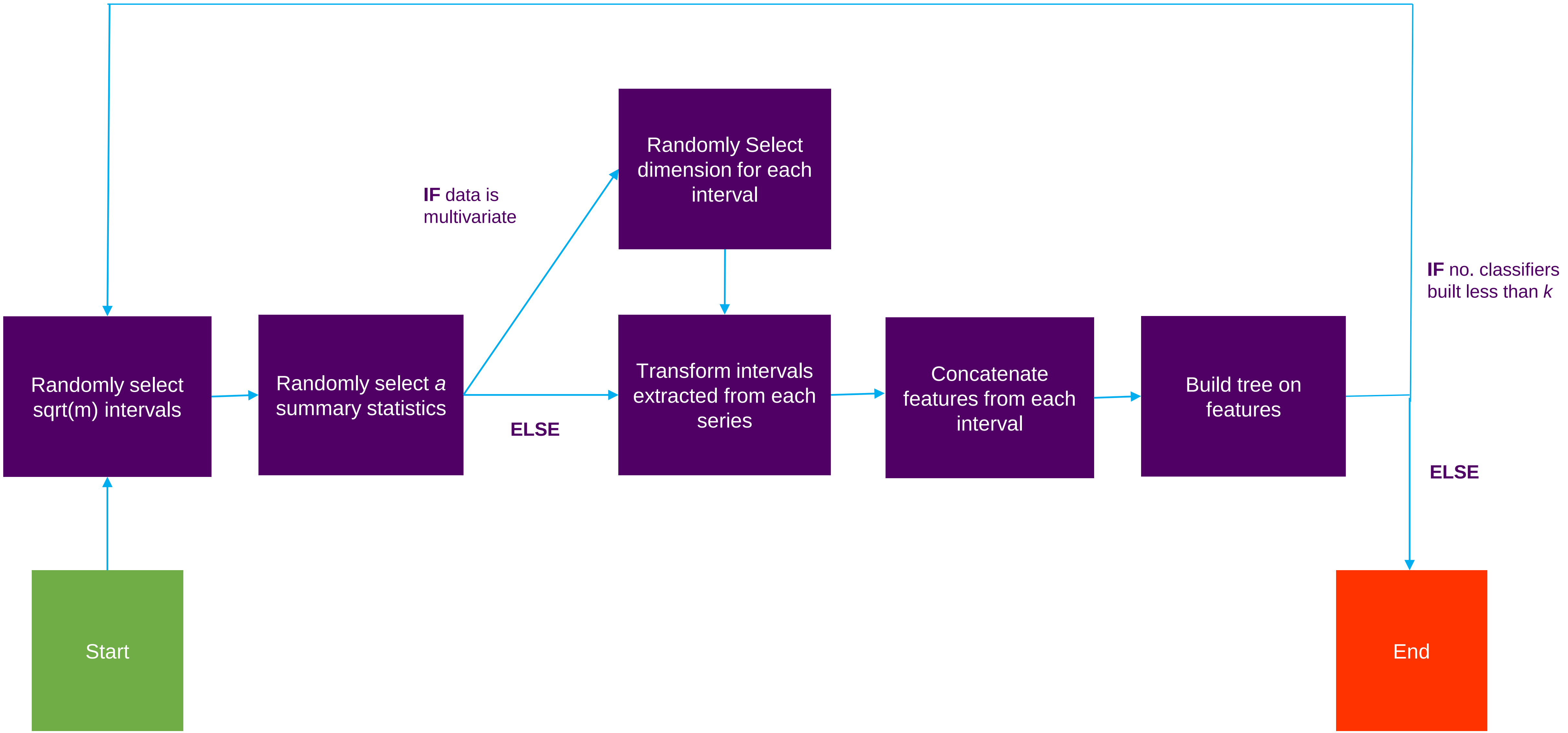
Publisher: IEEE | Cite This | PDF

Matthew Middlehurst ; James Large ; Anthony Bagnall **All Authors**

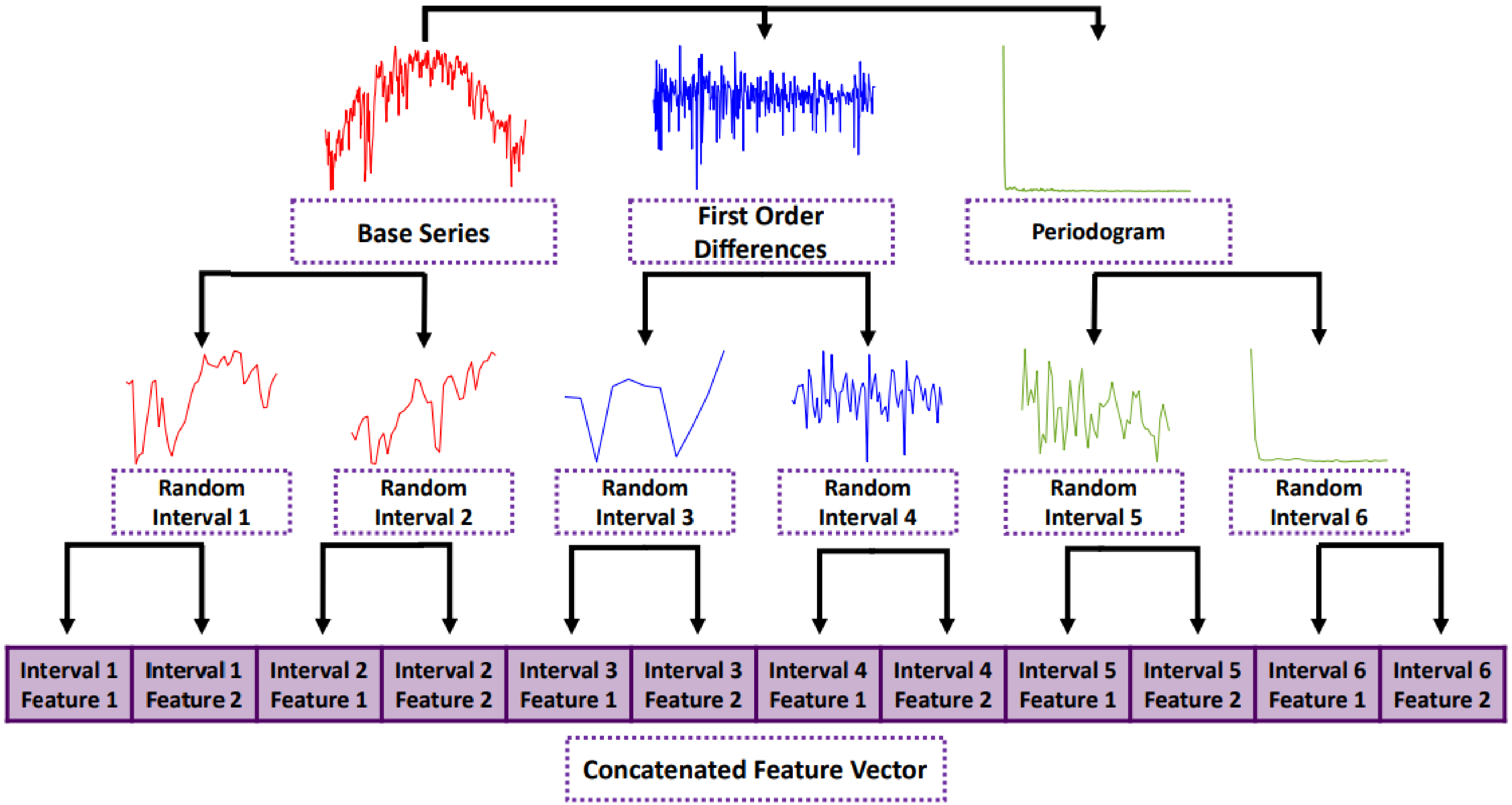
Diverse Representation: use raw data, the periodograms and first order differences

Canonical Interval Forest: derive random set of summary features (catch22) on each interval, concatenate into a new feature space for each tree

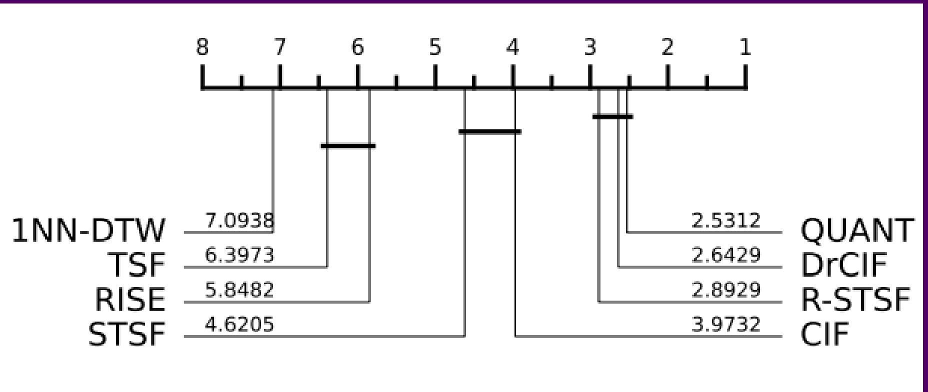
CIF Build Process



Diverse Representation Canonical Interval Forest (DrCIF)



Comparison



QUANT finds quantile intervals in a pipeline and is fast

Randomised Supervised Time Series Forest (RSTF) is is an interval based tree ensemble that includes a supervised method for extracting intervals

arXiv > cs > arXiv:2308.00928

Computer Science > Machine Learning

[Submitted on 2 Aug 2023]

QUANT: A Minimalist Interval Method for Time Series Classification

Angus Dempster, Daniel F. Schmidt, Geoffrey I. Webb

We show that it is possible to achieve the same accuracy, on average, as the most accurate existing interval methods for time series classification on a standard set of benchmark datasets using a single type of feature (quantiles), fixed intervals, and an 'off the shelf' classifier. This distillation of interval-based approaches represents a fast and accurate method for time series classification, achieving state-of-the-art accuracy on the expanded set of 142 datasets in the UCR archive with a total compute time (training and inference) of less than 15 minutes using a single CPU core.

Comments: 26 pages, 20 figures

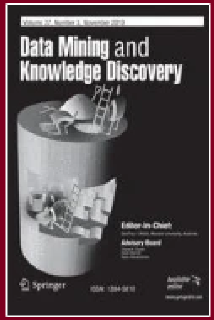
MONASH University

THE UNIVERSITY OF MELBOURNE

Home > Data Mining and Knowledge Discovery > Article

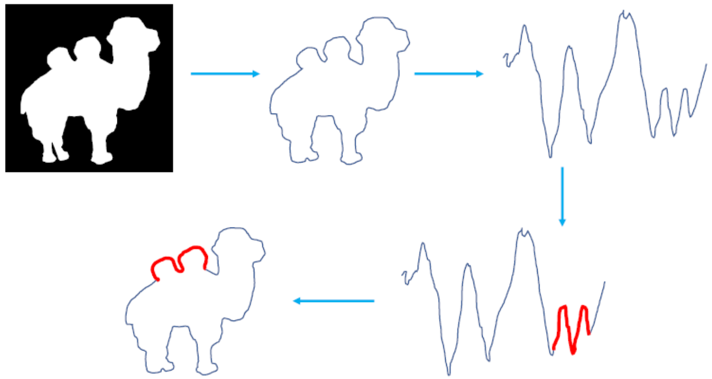
Fast, accurate and explainable time series classification through randomization

Open access | Published: 16 October 2023 | (2023)

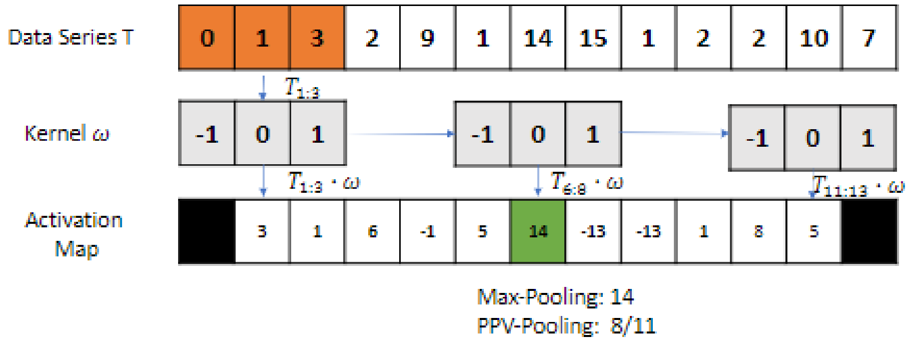


Part II: Time Series Classification

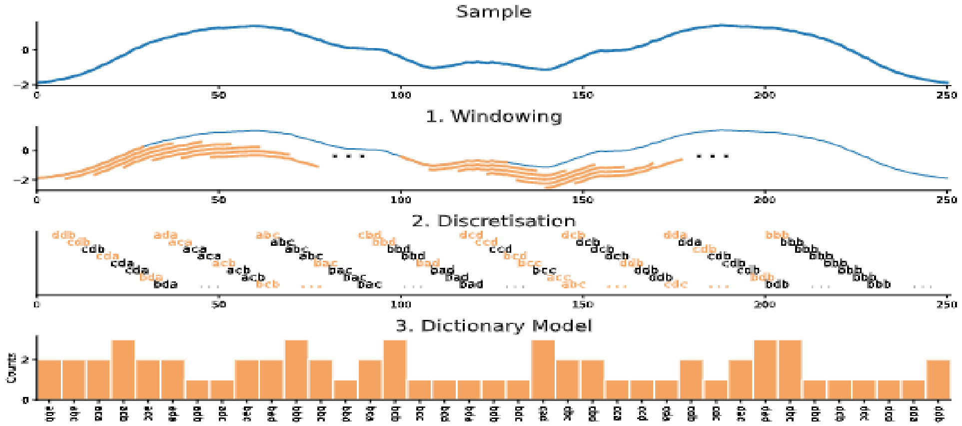
Shapelet based



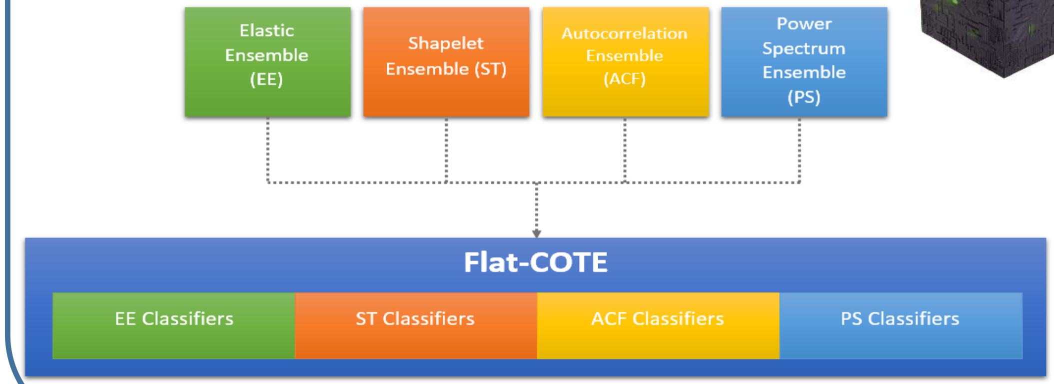
Convolution based



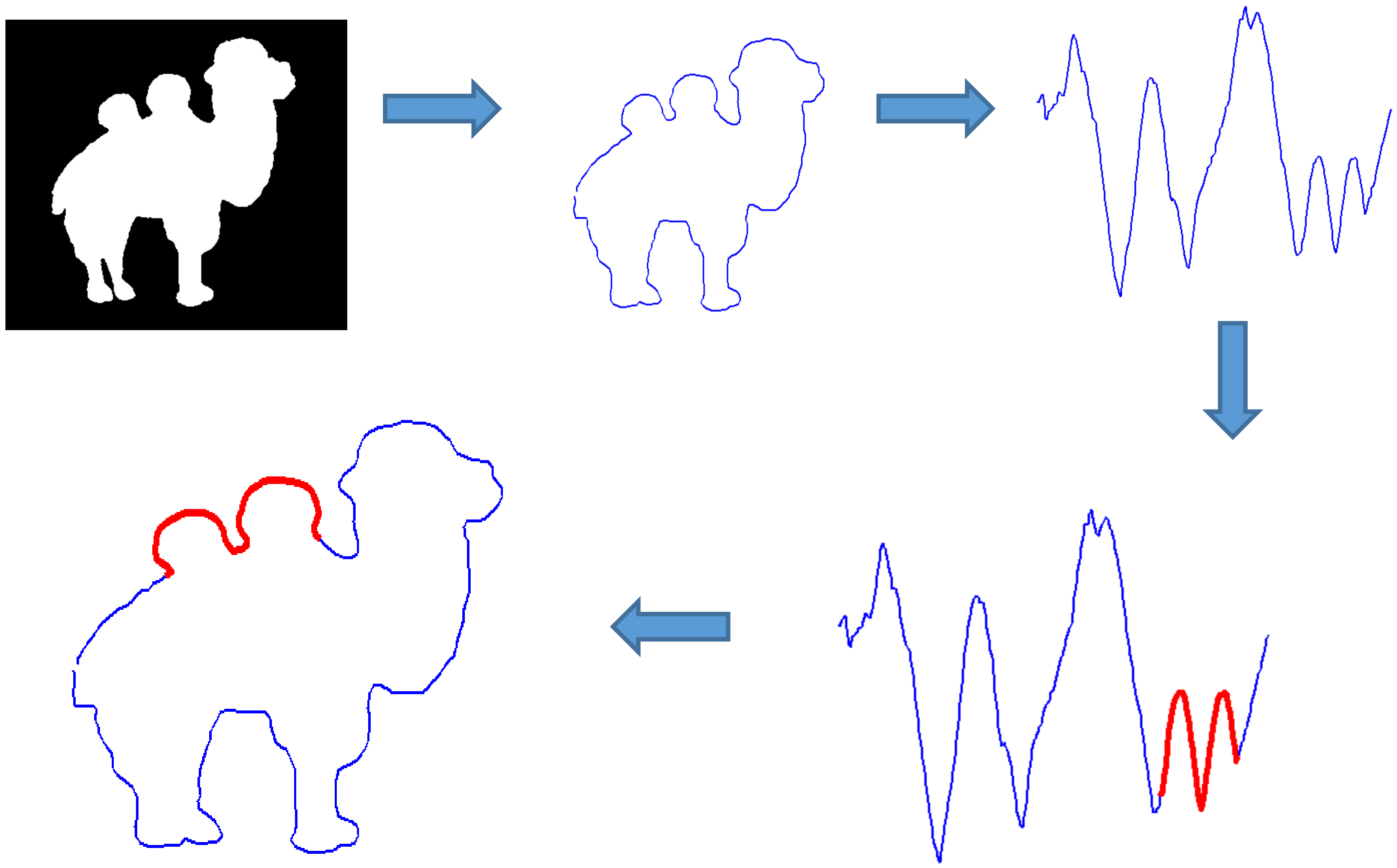
Dictionary based



Hybrids



Shapelet Based Classifiers



Shapelets fundamentals

- Time series shapelets are a data mining primitive that were first introduced by [1].
- They provide a mechanism for measuring local similarity between series by observing common subsequences.
- The original research embedded shapelets in a decision tree for TSC.
- The **shapelet transform** [2] separated the finding of shapelets from the classifier.
- There have been many variants proposed subsequently

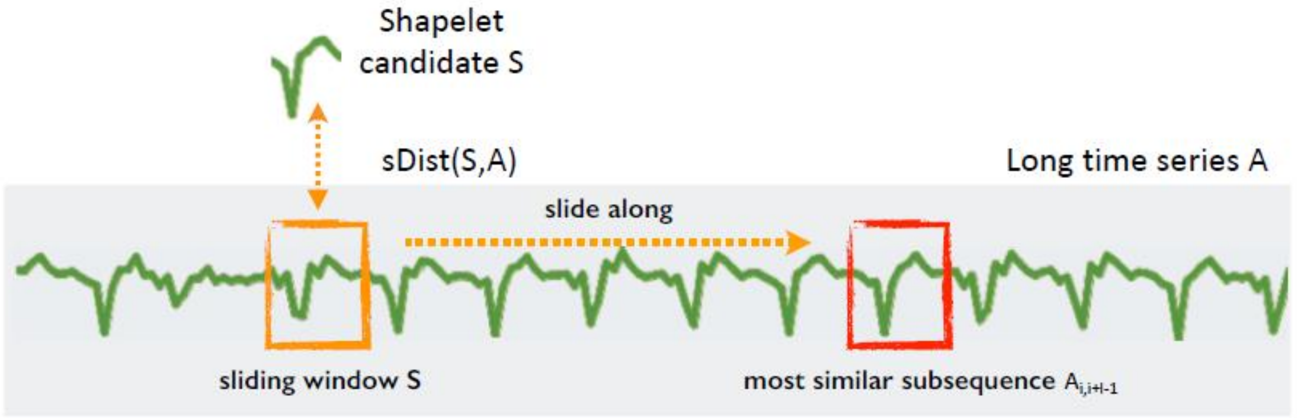
[1] *L. Ye and E. Keogh. Time series shapelets: A new primitive for data mining. In Proc. 15th ACM SIGKDD, 2009.*

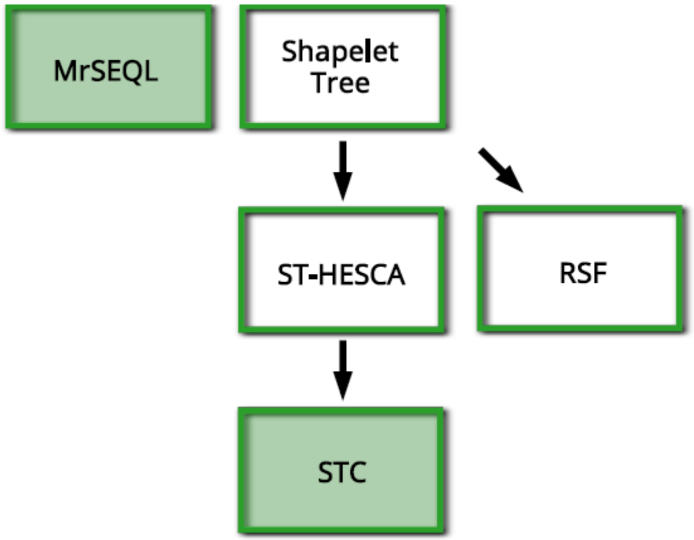
[2] *J. Lines et al. A shapelet transform for time series classification. In Proc. 18th ACM SIGKDD, 2012.*

Shapelet Based Classifiers

Shapelets are discriminatory phase independent subseries taken from the train data

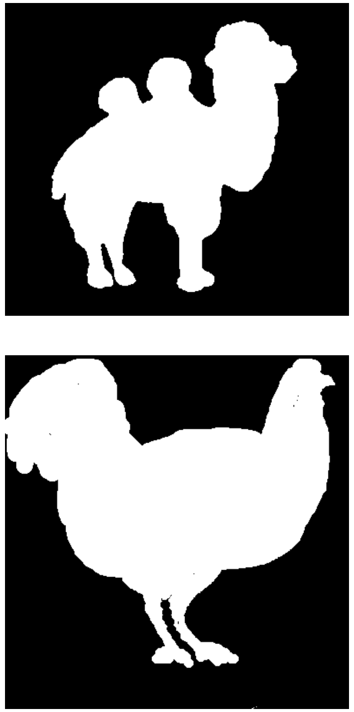
STC is a pipeline: select a set of shapelets, transform into distance to shapelet then build standard classifier



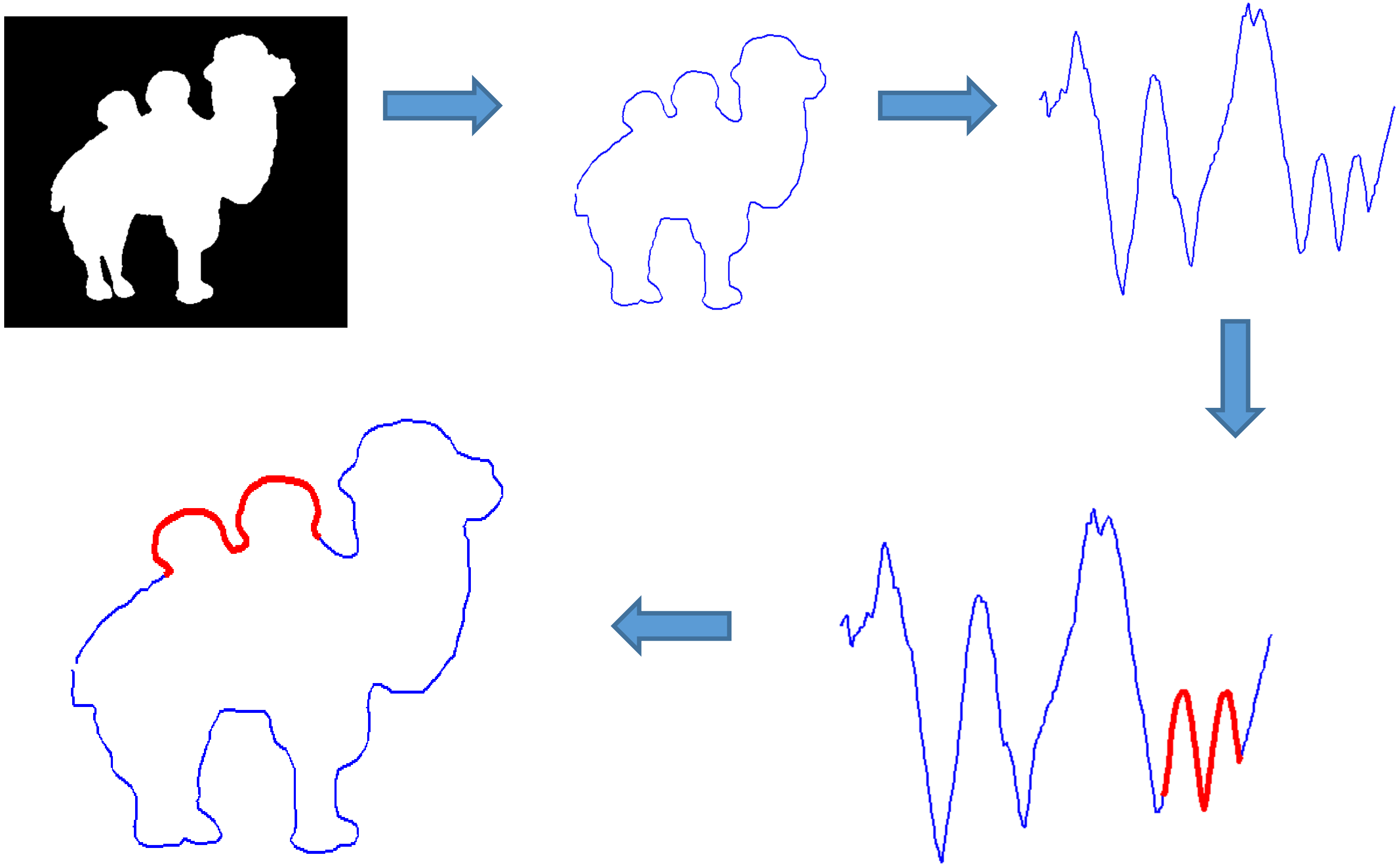


Shapelet Based	
Shapelet Tree	Shapelet Tree [Ye and Keogh 2011]
RSF	Generalised Random Shapelet Forest [Karlsson et al. 2016]
ST-HESCA	Shapelet Transform with HESCA Base Classifier [Bostrom and Bagnall 2017]
STC	Shapelet Transform Classifier [Bagnall et al. 2020]
MrSEQL	Multiple Representation Sequence Learner [Nguyen et al. 2017]

Camel or Chicken?



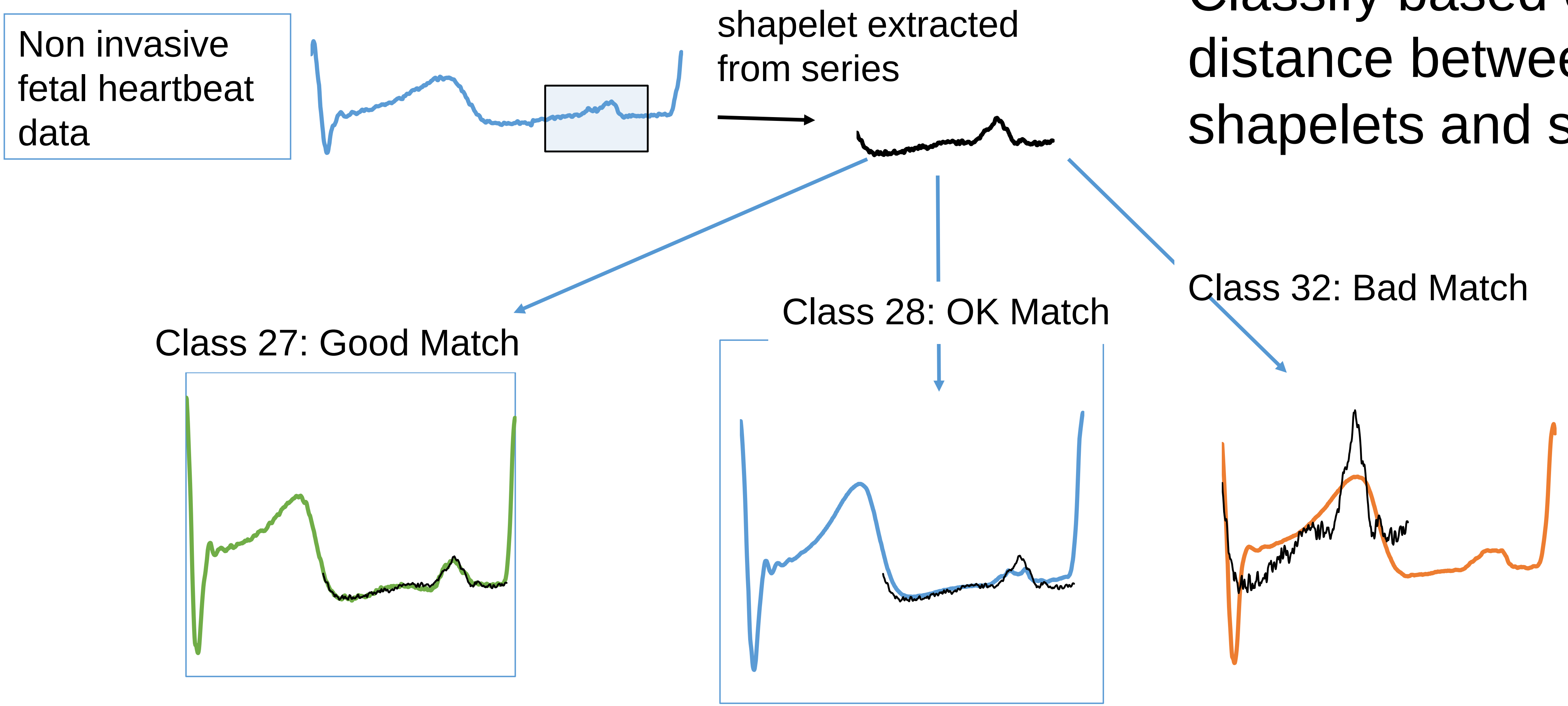
Shapelets are discriminatory phase independent



Shapelets are taken from the train data

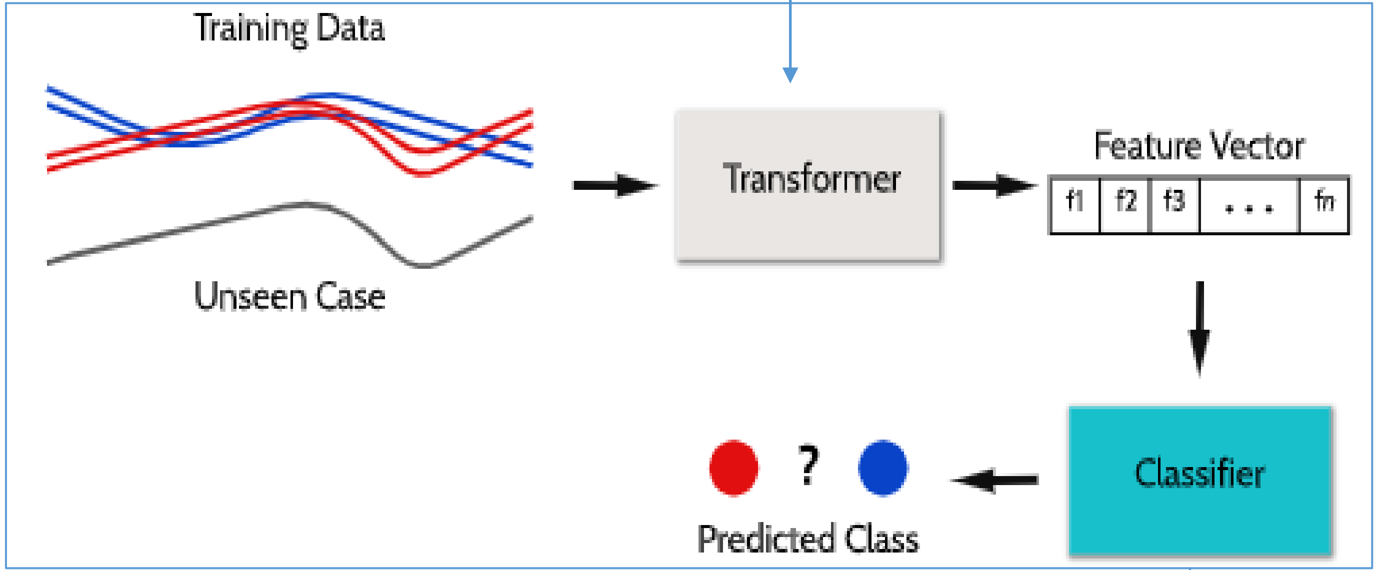
Using Shapelets

Classify based on distance between shapelets and series



Shapelet Transform Classifier

Transform using *k* shapelets into a vector of distances to shapelets



Rotation Forest Classifier

Key Issues

1. How to select shapelets?
2. How to measure the distance between a shapelet and a single series?
3. How to measure shapelet quality?
4. What classifier to use?

Selecting shapelets

1. The original shapelet transform followed the original paper and enumerated **all** possible shapelets
2. This is **very inefficient** and is in impractical for all but the smallest problems.
3. It is also **less effective** as it can lead to over fitting
4. The current version of STC randomly samples a large number (default 10k) and keeps the best k (default 1000)

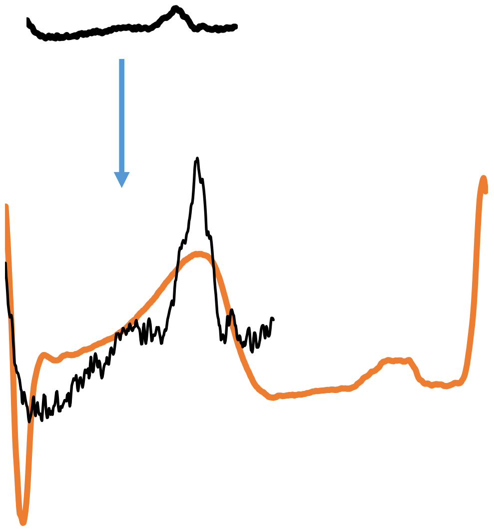
Home > Advanced Analytics and Learning on Temporal Data > Conference paper

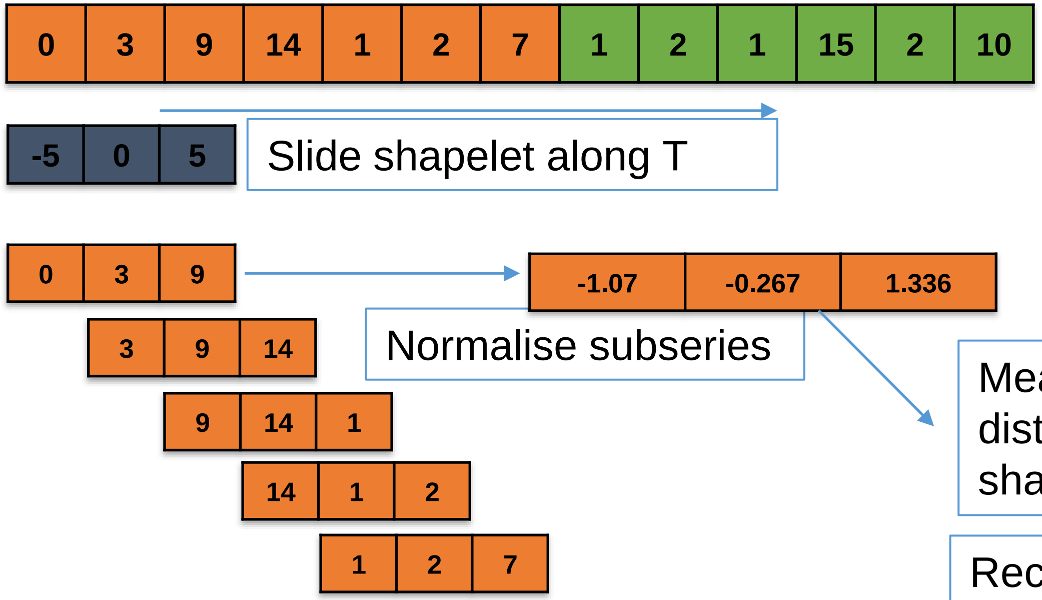
On the Usage and Performance of the Hierarchical Vote Collective of Transformation-Based Ensembles Version 1.0 (HIVE-COTE v1.0)

Conference paper | First Online: 16 December 2020

Shapelet Distance

We need to measure how likely it is that a shapelet is in any time series





$$dist(S,R)=\sum_{i=1}^{l}(s_i-r_i)^2$$

Measure Euclidean distance to normalised shapelet

Record lowest distance so far slide along

The distance between a subsequence S of length l and a time series T_i is the minimum distance between S and all **normalised** subsequences of T_i of length l

Algorithm 3 sDist(T, S)

```
Where T is a time series and S is a shapelet candidate.
l = |S|
min_dist = ∞
for p = 0 to |T| − l + 1 do
   dist = dist(S, T_p^l)
   if dist < min_dist then
      min_dist = dist
return min_dist
```

Shapelet Quality

For each shapelet we calculate a list of distances from the shapelet to each of the training set time series

The quality of the shapelet is based on how well we can use these distance to predict the class values

We use binary shapelets that predict a single class. Suppose we have eight training cases and get these distances

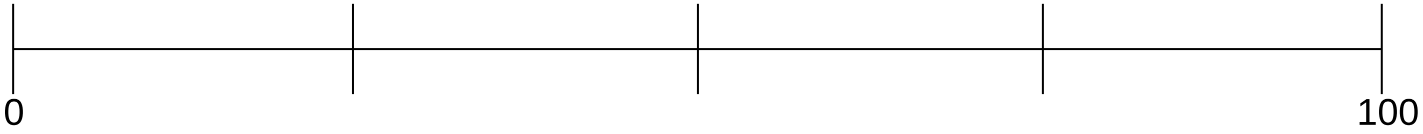
Class 1:
- 10.5
- 42.3
- 23.8
- 72.1

Class 2:
- 54.2
- 67.4
- 28.2
- 98.9

We measure the quality using a measure like information gain

The Order line

$D_S =$



Class 1:

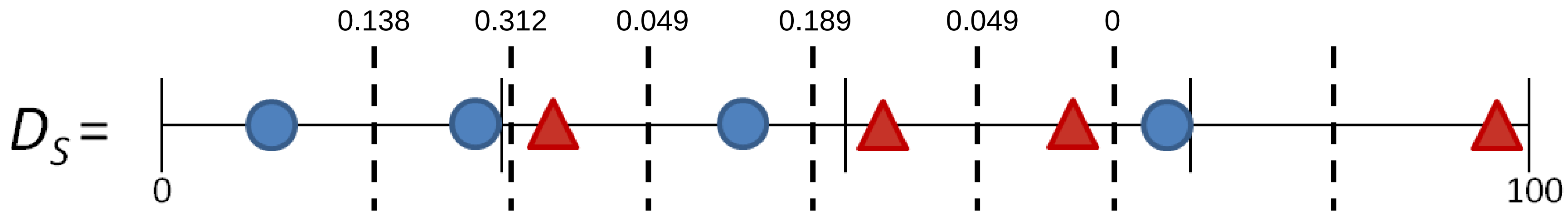
- 10.5
- 42.3
- 23.8
- 72.1

Class 2:

- 54.2
- 67.4
- 28.2
- 98.9

Information Gain

- Now that D_S is sorted, assess all possible split points, where a split point, *sp*, is the average between two consecutive distances.



Note: the position of the best split point is used as the decision criteria for the decision tree

Shapelet Transform

Find Shapelets

1. Randomly sample shapelets a fixed number of times for each class
2. Find distances to all series in the training set
3. Measure quality with IG
4. Keep the best scoring shapelets

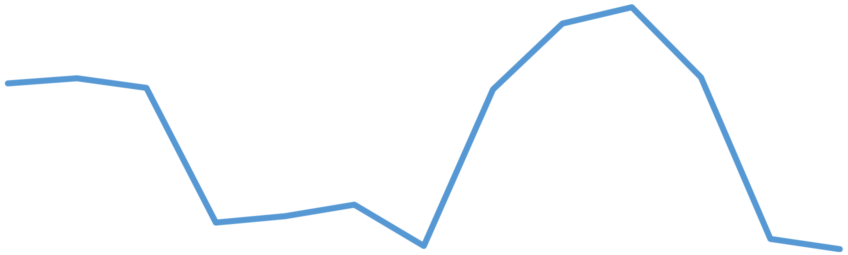
Find Transform

Train Classifier

The Transform

- 4 extracted shapelets:

- Given a series T_i



- Calculate: dist(T_i,$S1$), dist(T_i,S_2), dist(T_i,S_3), dist(T_i,S_4)

0.024 5.124 7.682 0.715

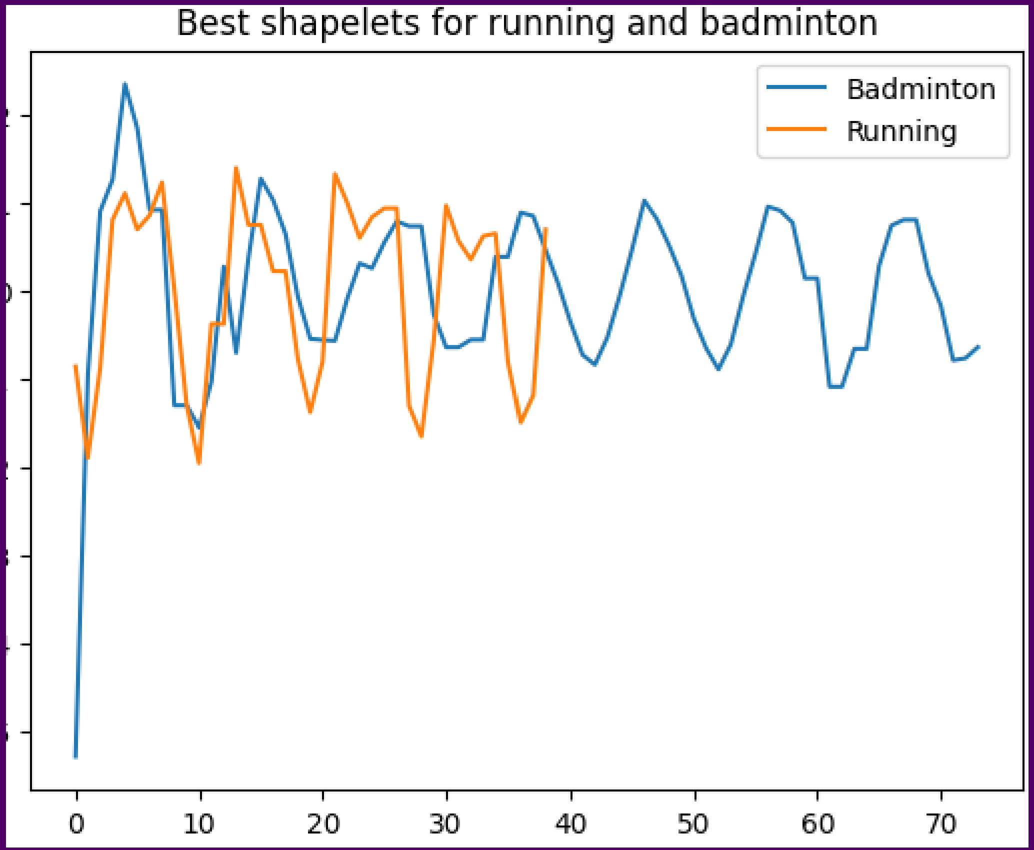
Create a tabular train dataset

	S1	S2	S3	S4
T1	0.024	5.124	7.682	0.715
T2	0.567	1.234	4.667	0.001
T3	4.567	6.231	12.435	4.456
T4	2.33	0.184	3.111	5.232

Visualising Shapelets

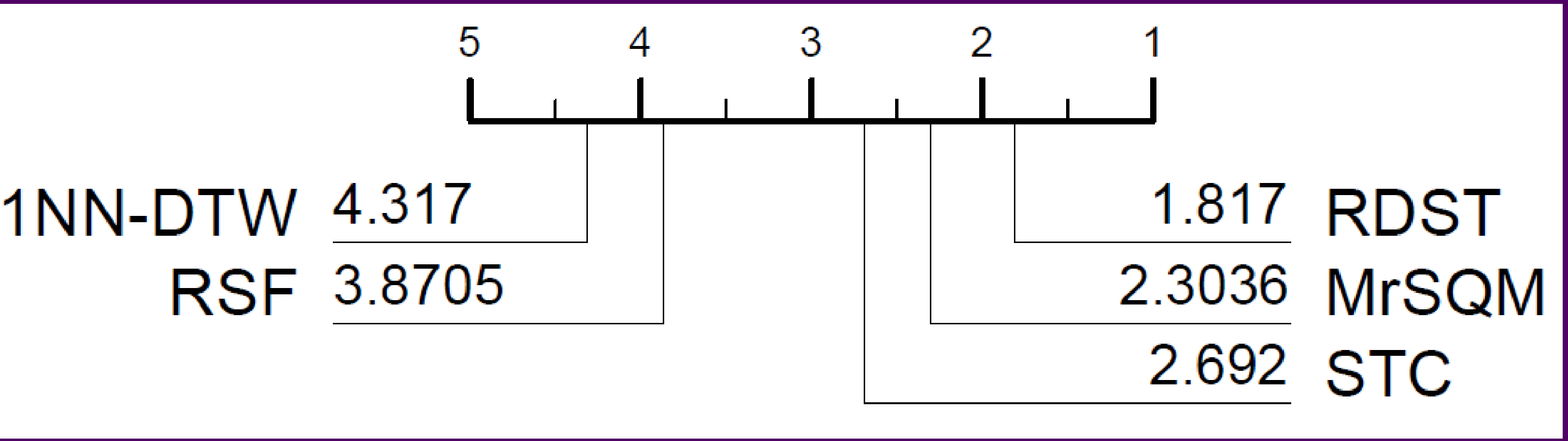
https://www.aeon-toolkit.org/en/stable/examples/classification/shapelet_based.html

```python
X, y = load_basic_motions(split="train")
rst = RandomShapeletTransform(n_shapelet_samples=100, max_shapelets=10, random_state=42)
st = rst.fit_transform(X, y)
print(" Shape of transformed data = ", st.shape)
print(" Distance of second series to third shapelet = ", st[1][2])
testX, testy = load_basic_motions(split="test")
tr_test = rst.transform(testX)
rf = RandomForestClassifier(random_state=10)
rf.fit(st, y)
preds = rf.predict(tr_test)
print(" Shapelets + random forest acc = ", accuracy_score(preds, testy))
```




```python
"""Shapelet based time series classifiers."""

__all__ = [
    "MrSQMClassifier",
    "ShapeletTransformClassifier",
    "RDSTClassifier",
    "SASTClassifier",
    "RSASTClassifier",
    "LearningShapeletClassifier",
]
```

Best in class: RDST



Inspired by ROCKET, RDST employs dilation to improve performance.

Université d'ORLÉANS

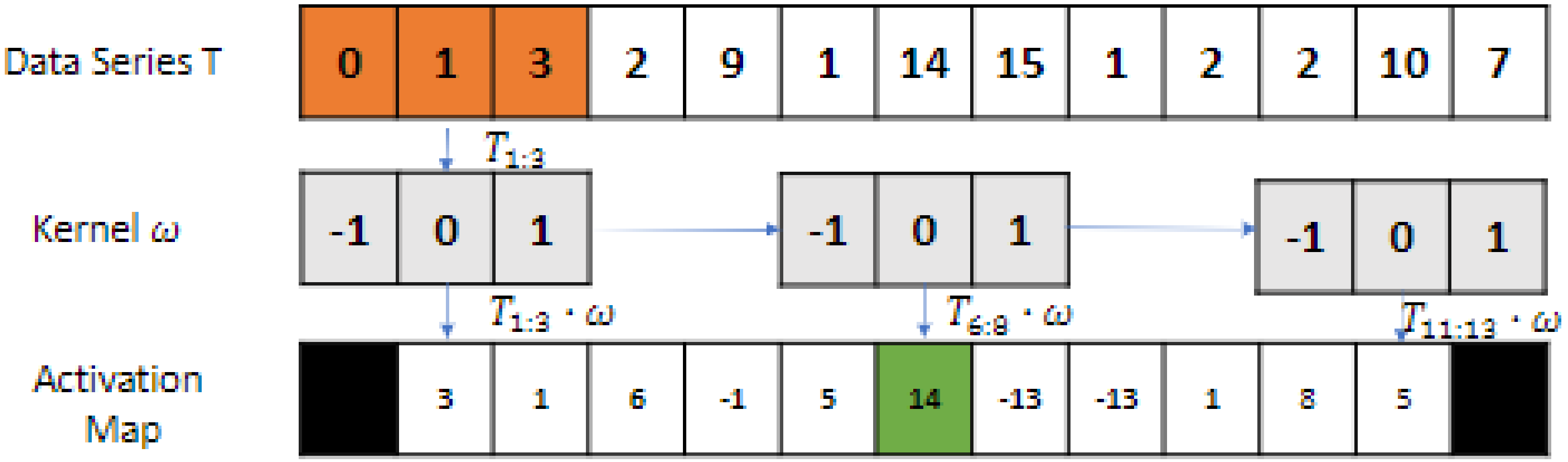
Random Dilated Shapelet Transform: A New Approach for Time Series Shapelets

Authors: Antoine Guillaume, Christel Vrain, Wael Elloumi Authors Info & Claims

Pattern Recognition and Artificial Intelligence: Third International Conference, ICPRAI 2022, Paris, France, June 1–3, 2022, Proceedings, Part I • Jun 2022 • Pages 653–664 • https://doi.org/10.1007/978-3-031-09037-0_53

Published: 01 June 2022 Publication History

Convolution Based Pipelines

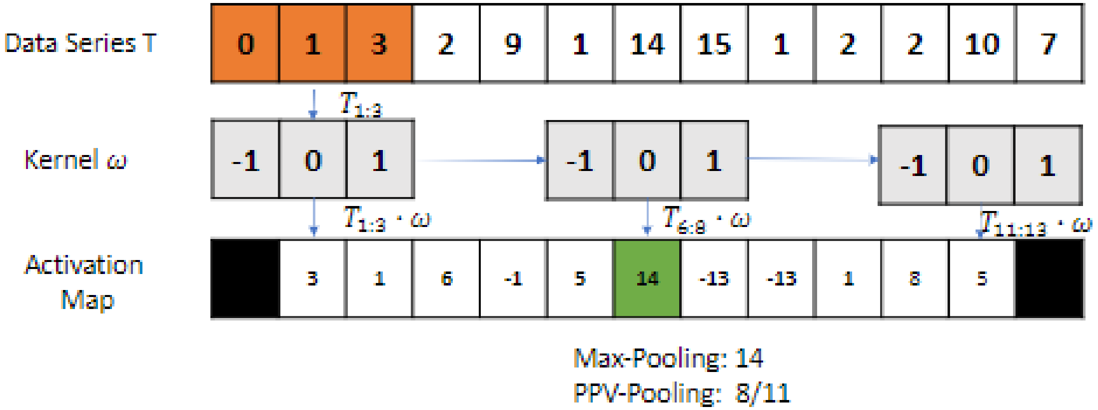


Max-Pooling: 14
PPV-Pooling: 8/11

Convolution/kernel based:ROCKET

Geoff Webb’s group in Monash proposed a simple approach to TSC that does surprisingly well

1. Create a large number of random convolutions
2. Create feature vectors by pooling operations
3. Fit a linear classifier



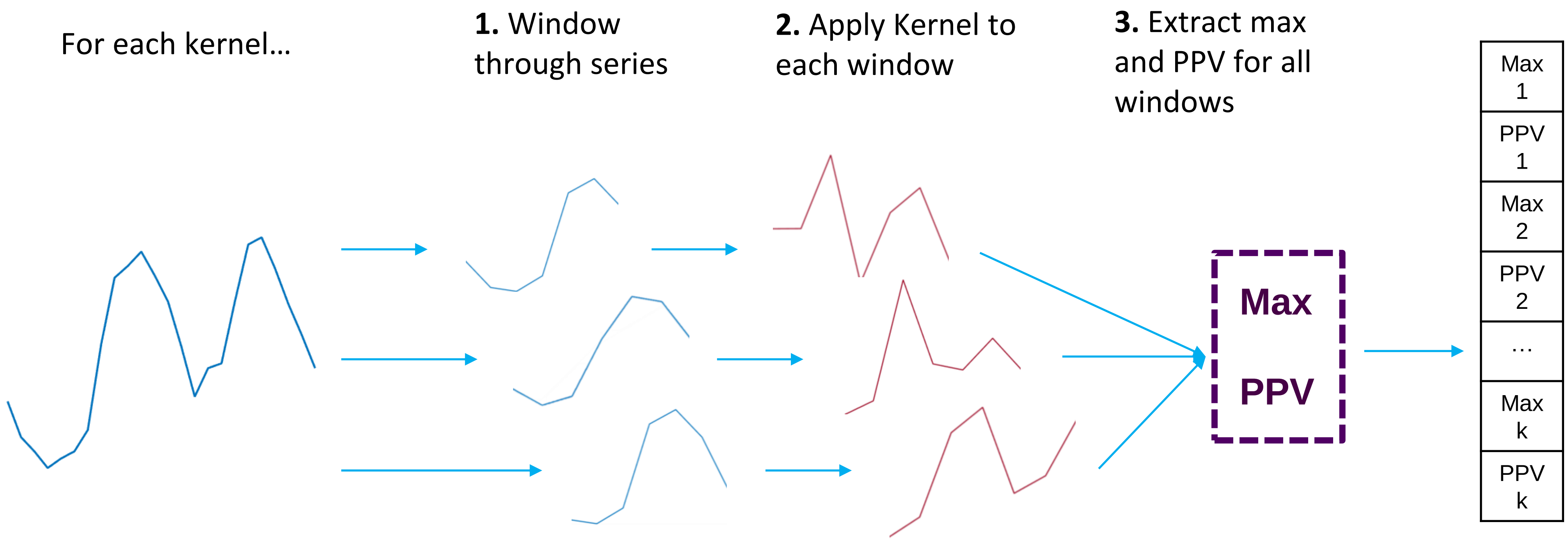
Published: 13 July 2020

ROCKET: exceptionally fast and accurate time series classification using random convolutional kernels

Angus Dempster, François Petitjean & Geoffrey I. Webb

MONASH University

ROCKET transform



The terms kernel and convolution are used interchangably

It is used in the same way as a deep learning convolution, **but it is not learnt. It is random**

It is also similar to a shapelet, but it is not from the training data.

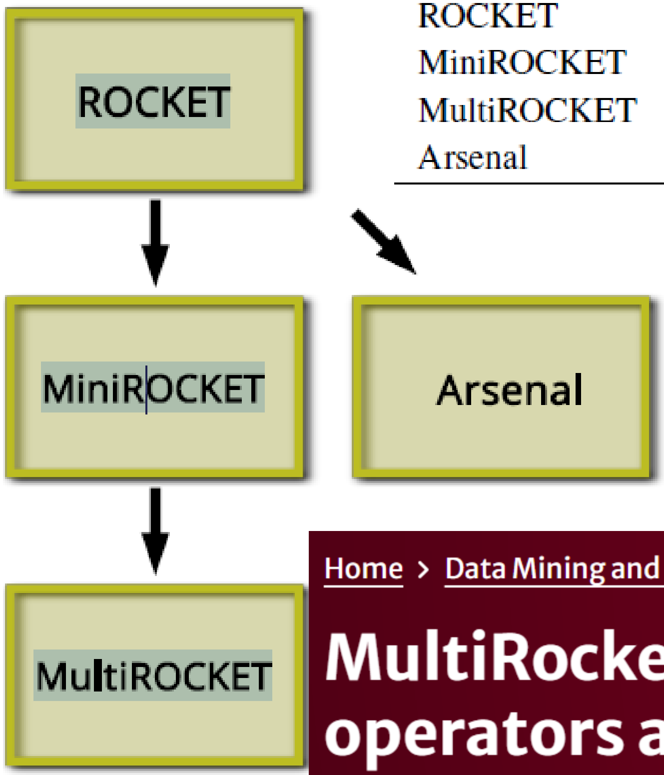
This is called the **activation map** in deep learning

This is called the **pooling operation** in deep learning

PPV is **percentage of positive values**

Shapelet operation is like max pooling, but shapelets use Euclidean distance instead of dot product, and minimise

ROCKET Family: Convolution-Based



Kernel/Convolution Based

Hybrid STC	Hybrid Shapelet Transform Classifier [Guijo-Rubio et al. 2019]
ROCKET	Random Convolutional Kernel Transform [Dempster et al. 2020]
MiniROCKET	MINImally RandOm Convolutional KErnel Transform [Dempster et al. 2021]
MultiROCKET	MiniRocket with multiple pooling operators and transformations [Tan et al. 2022
Arsenal	Arsenal [Middlehurst et al. 2021]

Home > Data Mining and Knowledge Discovery > Article

MultiRocket: multiple pooling operators and transformations for fast and effective time series classification

Open access | Published: 29 June 2022 | **36**, 1623–1646 (2022)

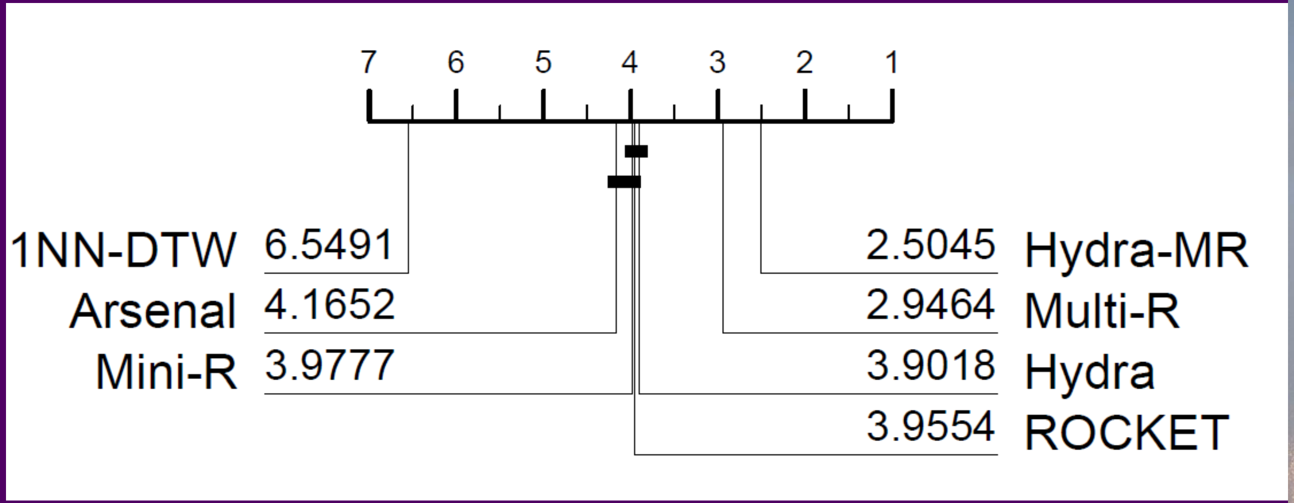
KDD > Proceedings > KDD '21 > MiniRocket: A Very Fast (Almost) Deterministic Transform

RESEARCH-ARTICLE

MiniRocket: A Very Fast (Almost) Deterministic Transform for Time Series Classification

Authors: Angus Dempster, Daniel F. Schmidt, Geoffrey I. Webb Authors Inf

Best in class: Multi-Rocket-Hydra

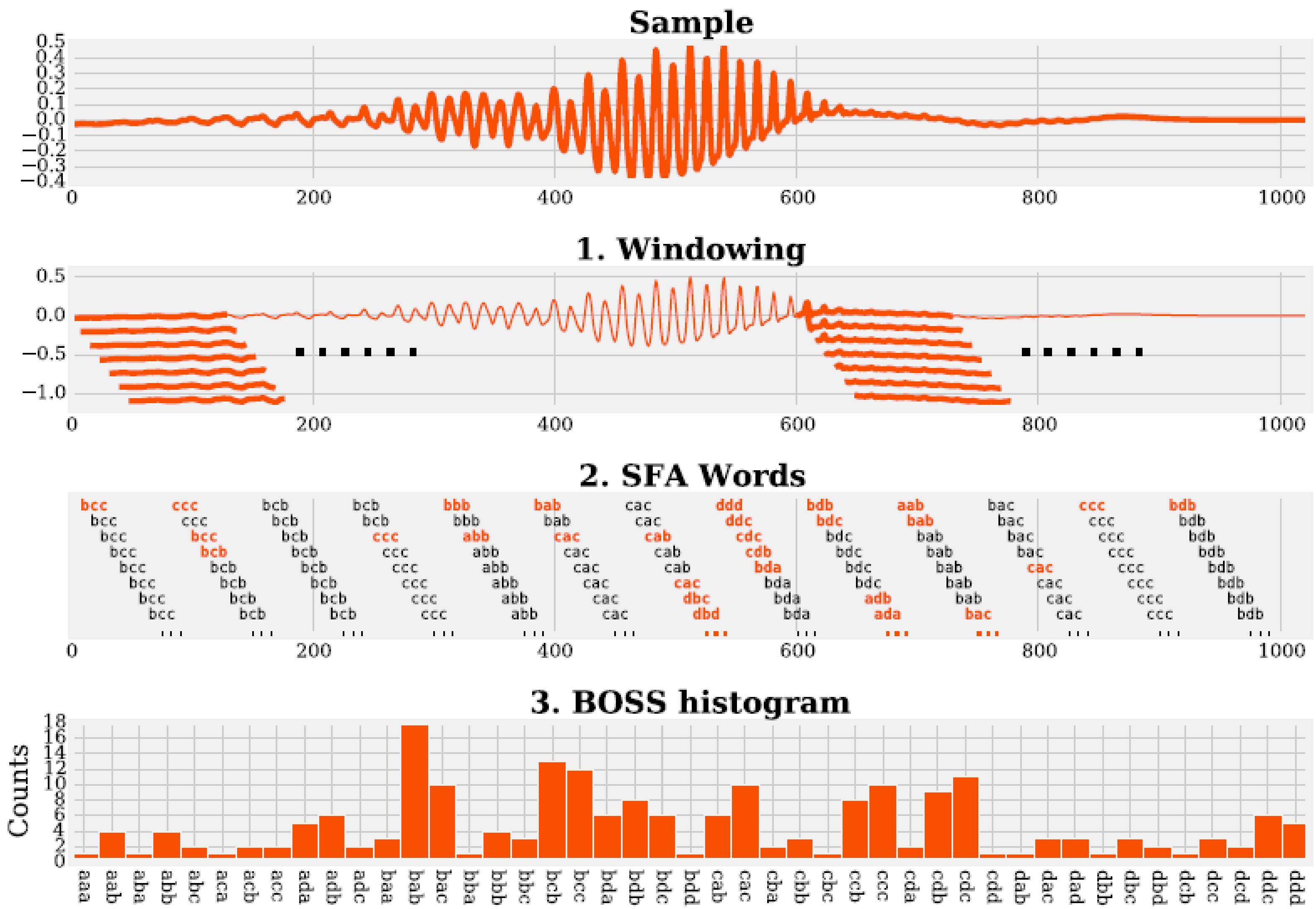


Home > Data Mining and Knowledge Discovery > Article

Hydra: competing convolutional kernels for fast and accurate time series classification

Open access | Published: 16 May 2023 | 37, 1779–1805 (2023)

Dictionary Based Ensembles

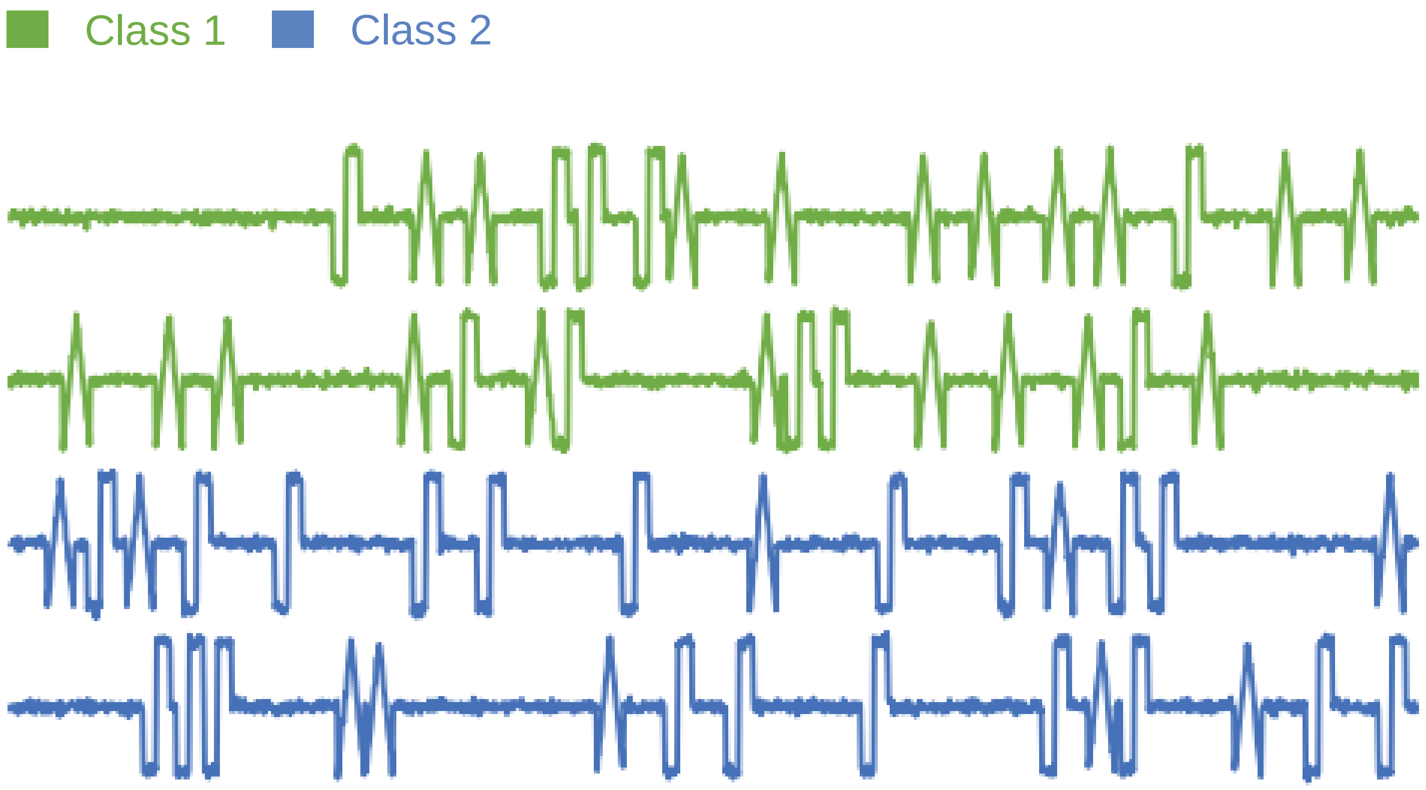


Dictionary based classifiers look for repeating patterns rather than the presence/absence of a pattern. This involves

1. Window series
2. Discretise windows
3. Count frequency of occurrence to form histogram

Dictionary Based Time Series Classification

Simulated time series suitable for dictionary based classification:



Bag of SFA Symbols (BOSS)

A comparison of time series classification techniques found BOSS to be the most accurate dictionary based classifier.

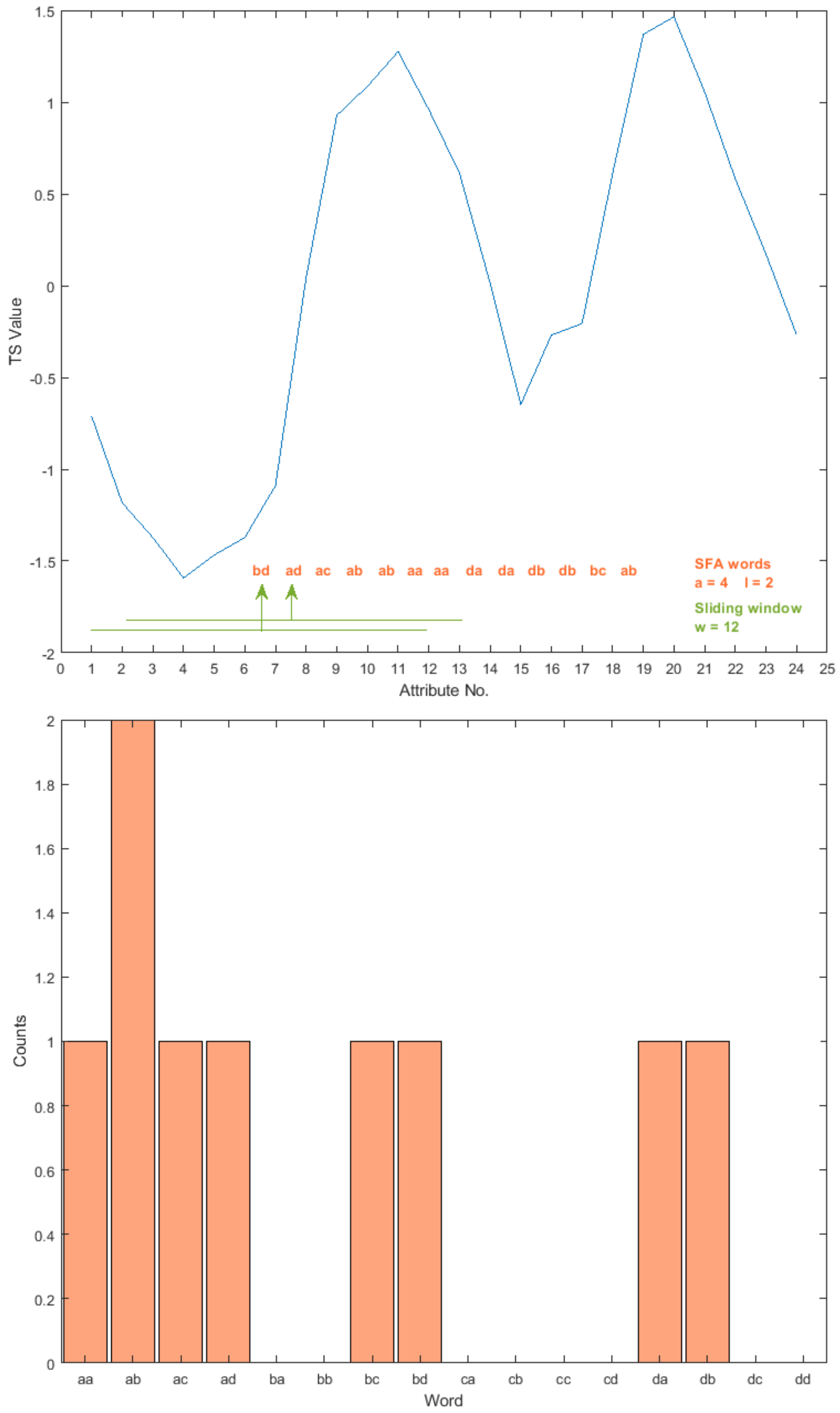
Classifier	Absolute Acc Diff to Best	Overall Rank
Flat-COTE	0%	1
ST	-2.00%	2
BOSS	-2.45%	3
EE	-4.61%	4
DTW 1-NN	-8.14%	16
BOP	-11.17%	18
SAXVSM	-11.41%	19

■ Dictionary Approaches

Bag of Words Model

Dictionary based classifiers make use of bags of words.

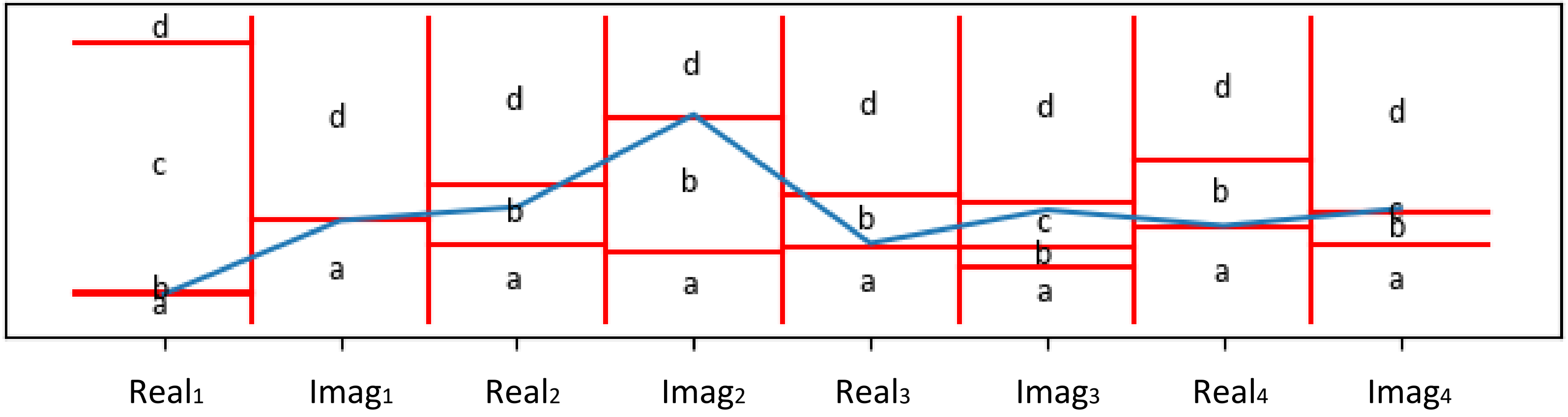
A sliding window is run over each series, with each being discretised into a word to form a histogram.



Symbolic Fourier Approximation (SFA)

A Discrete Fourier Transform (DFT) is performed on each window.
The first *l*/2 coefficients are discretised into a word using breakpoints generated using Multiple Coefficient Binning (MCB).

Word: A A B D B C B D



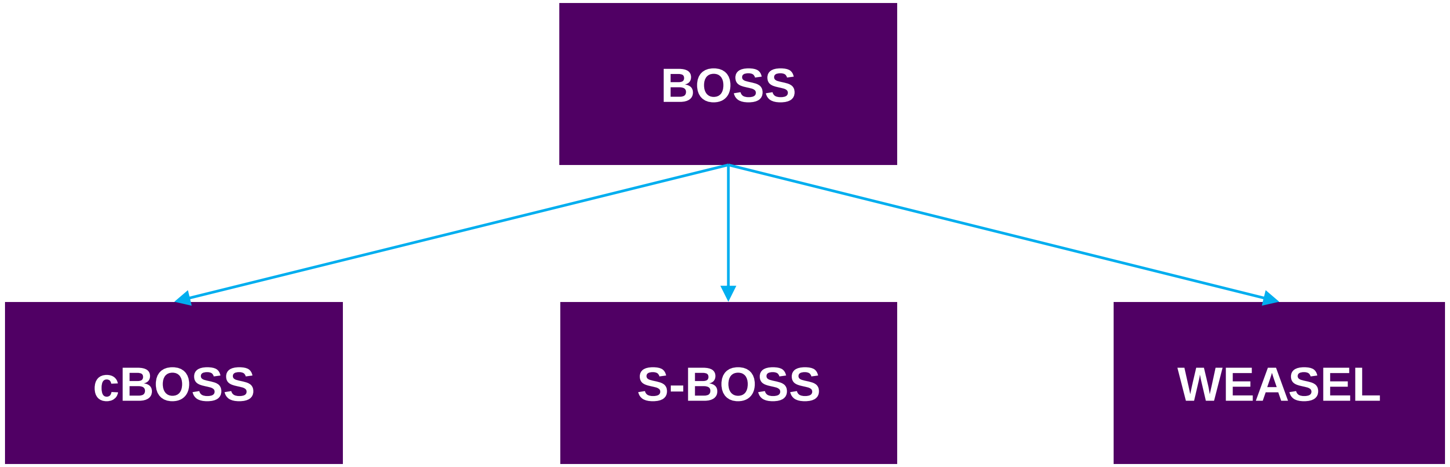
BOSS Parameters

Dictionary methods have a large range of parameters.

Most form an ensemble of classifiers using the top performing parameter sets.

Parameter		No. Values	Range
Word length	l	5	{16,14,12,10,8}
Window length	w	$m/4$	{10…m}
Normalise	p	2	{true,false}
Alphabet size	α	1	{4}

BOSS Extensions



cBOSS

Better scalability and usability.

Revised ensemble structure.

S-BOSS

Higher accuracy.

Addition of spatial pyramids, new distance measure.

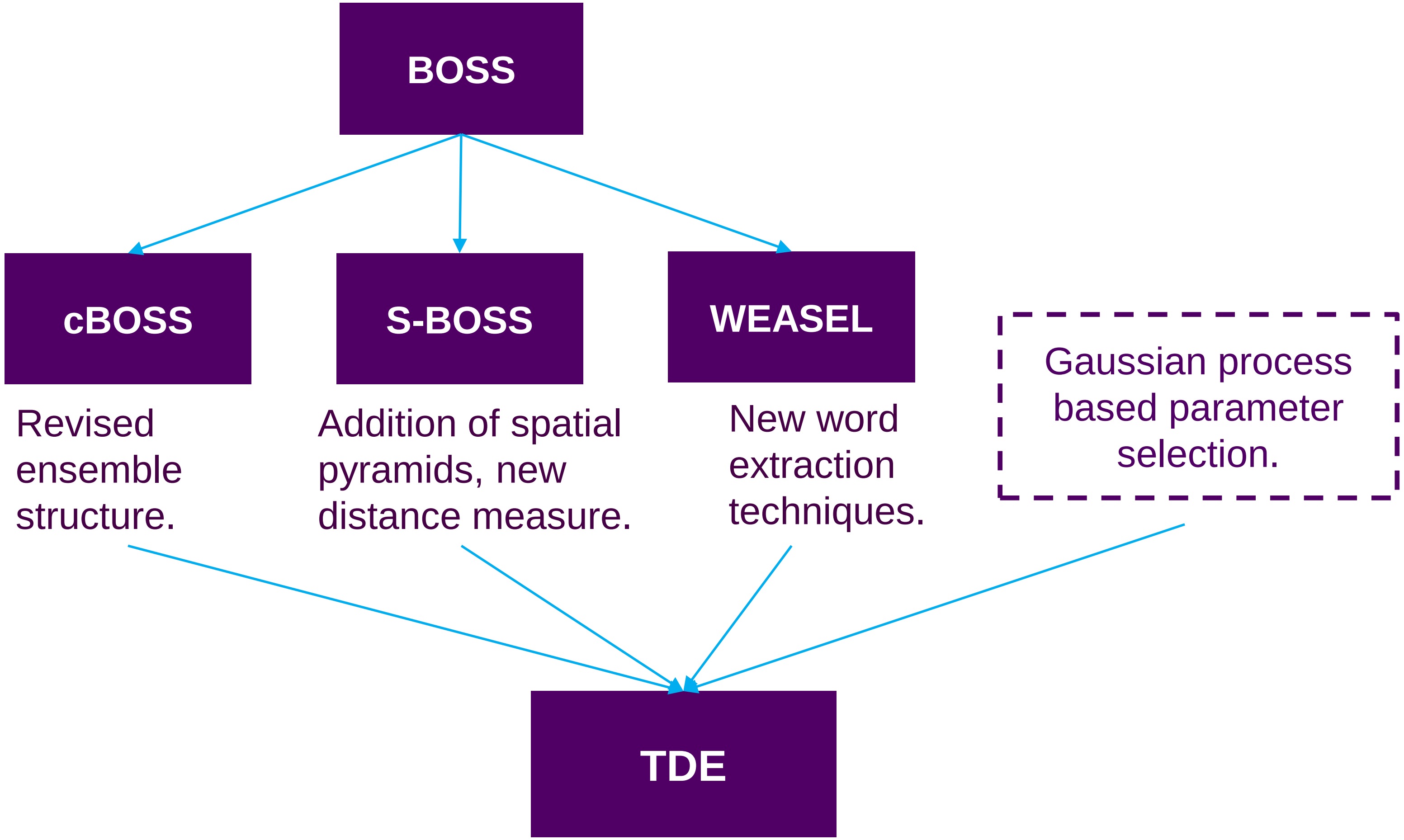
WEASEL

Higher accuracy, better prediction time.

No ensemble, new word extraction techniques.

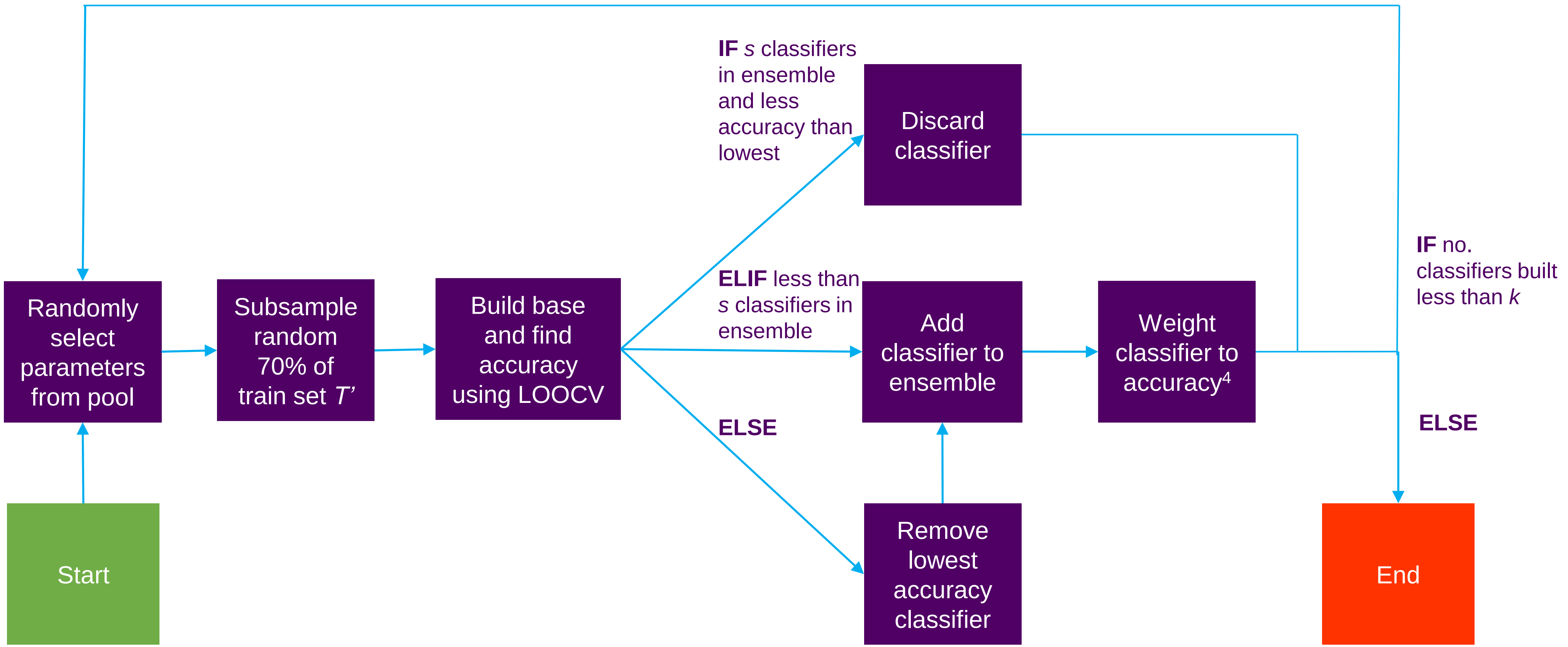
The Temporal Dictionary Ensemble (TDE)

While most classifier features can seamlessly merge, a smarter method for selecting ensemble member parameters is required.



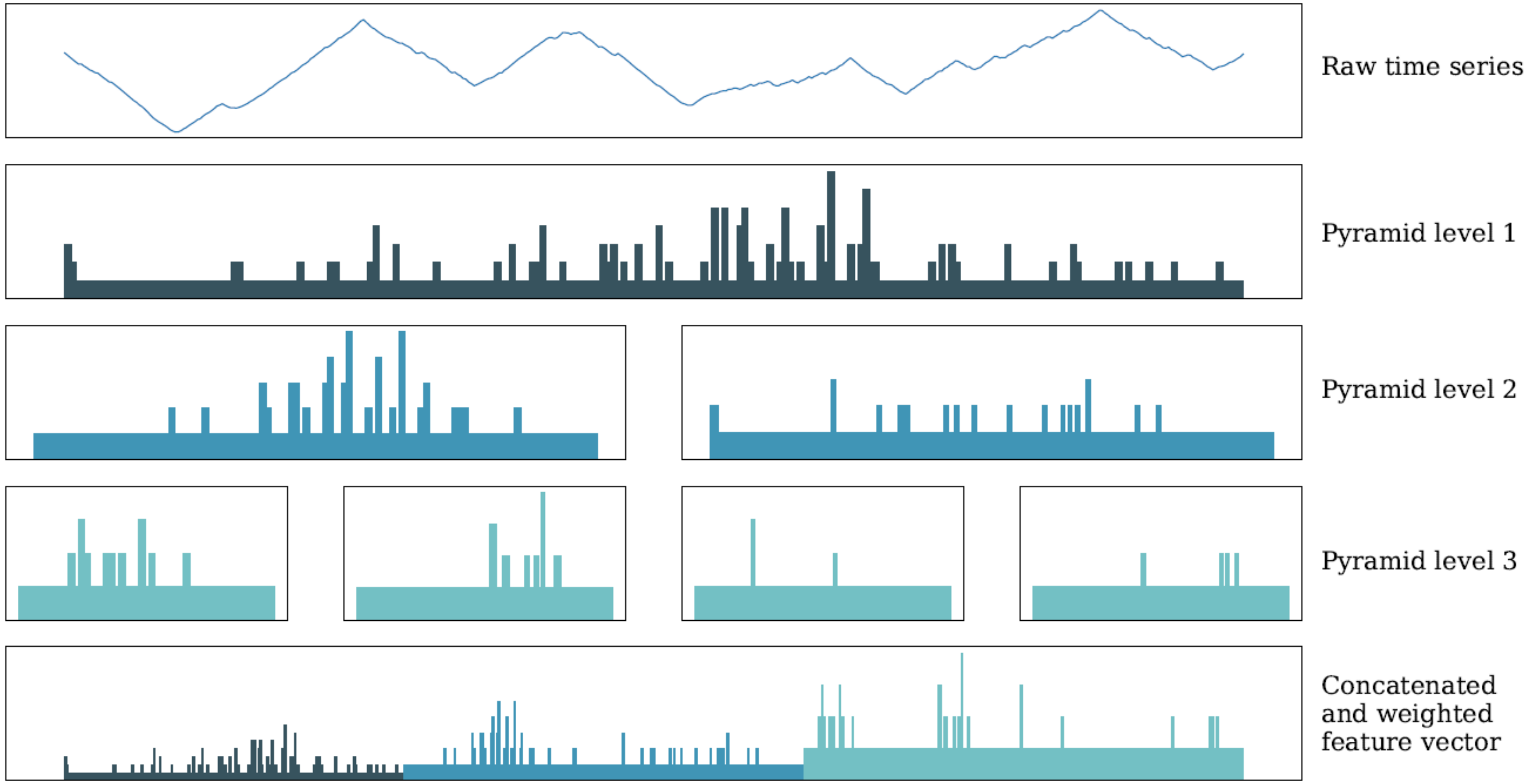
cBOSS Improvements

Improved ensemble building structure.



S-BOSS Improvements

Weighted spatial pyramid features.



WEASEL Improvements

- Introduction of bigrams for each window using the previous non-overlapping window.

AA BB AB BA → AA BB AB BA AA + AA AA + BB AA + AB . . .

- Information gain binning (IGB) as an option replacing MCB.

New TDE Parameter Space

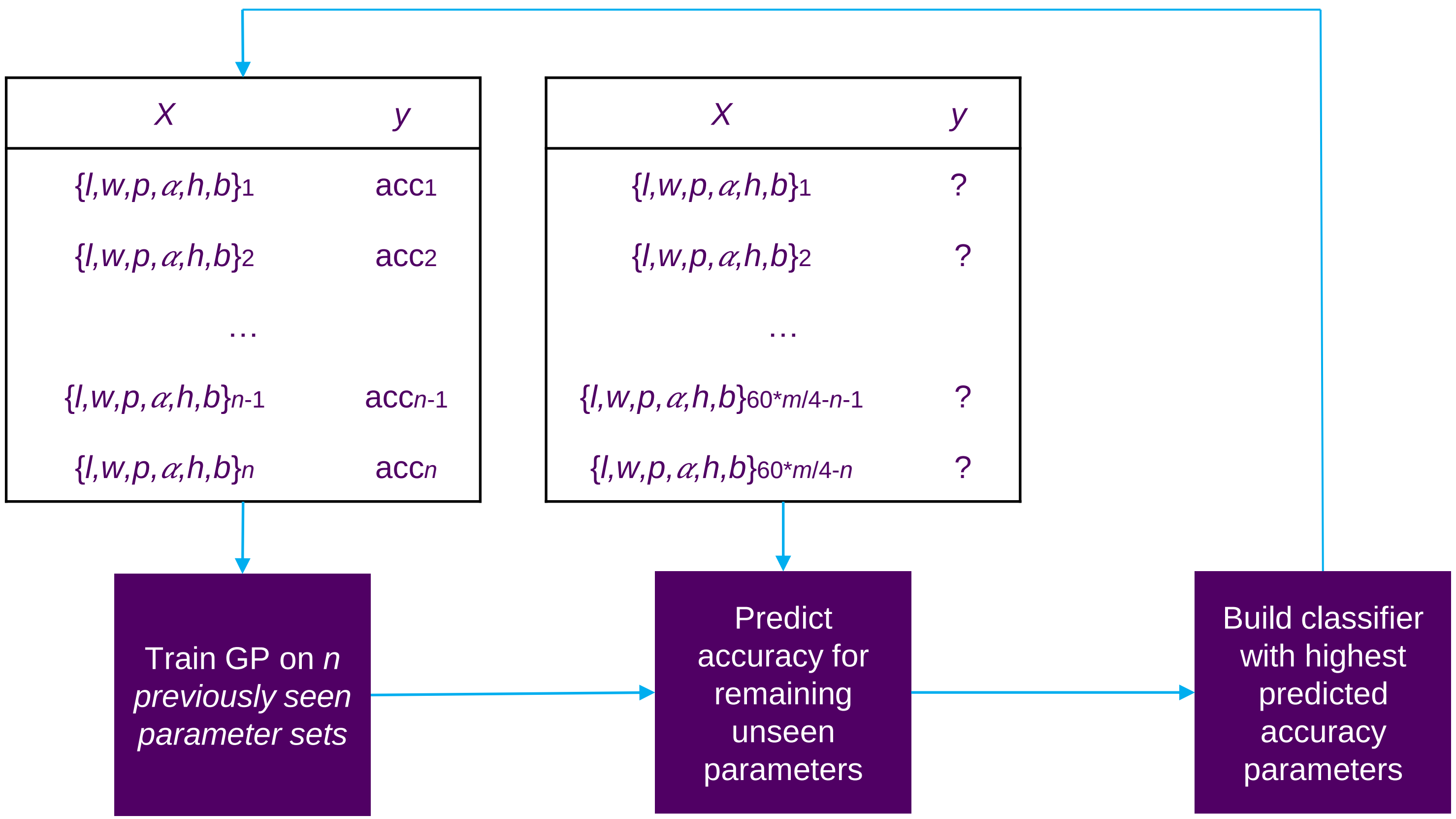
Problem: Cant effectively search expanded space with cBOSS random selection

Parameter		No. Values	Range
Word length	l	5	{16,14,12,10,8}
Window length	w	$m/4$	{10…m}
Normalise	p	2	{true,false}
Alphabet size	α	1	{4}
Pyramid levels	h	3	{1,2,3}
Discretisation	b	2	{MCB,IGB}

Gaussian Process (GP) Parameter Selection

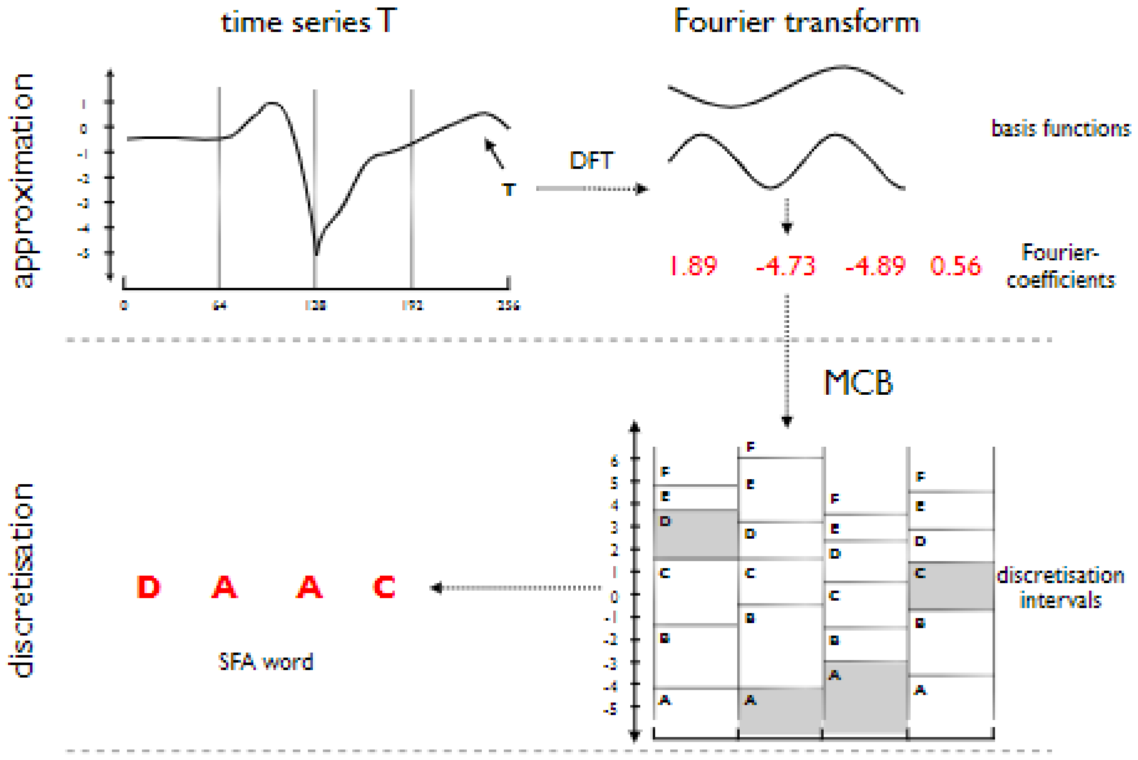
Using parameter sets from previously built classifiers, predict the accuracy of unseen ones.

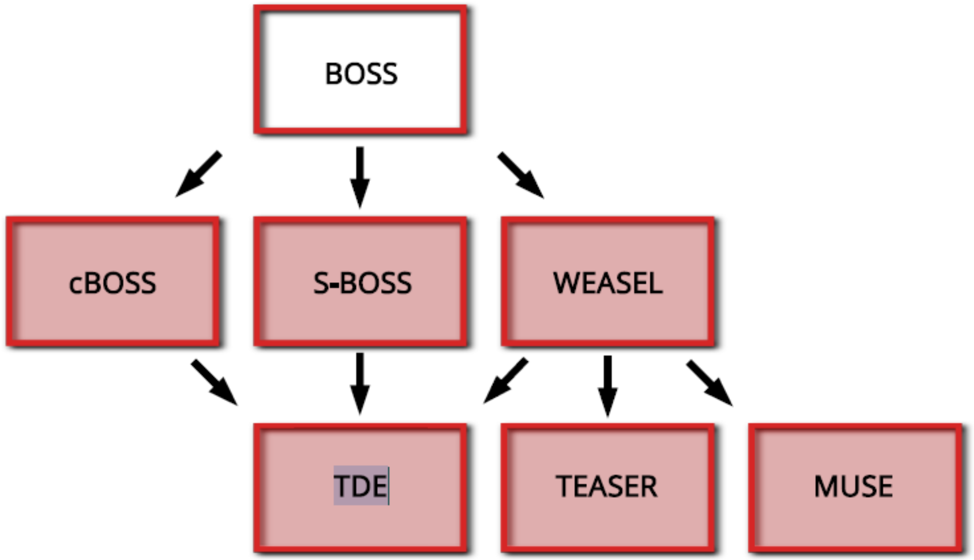
To include a diverse selection, randomly select the first 50 parameter sets.



Dictionary Based Classifiers

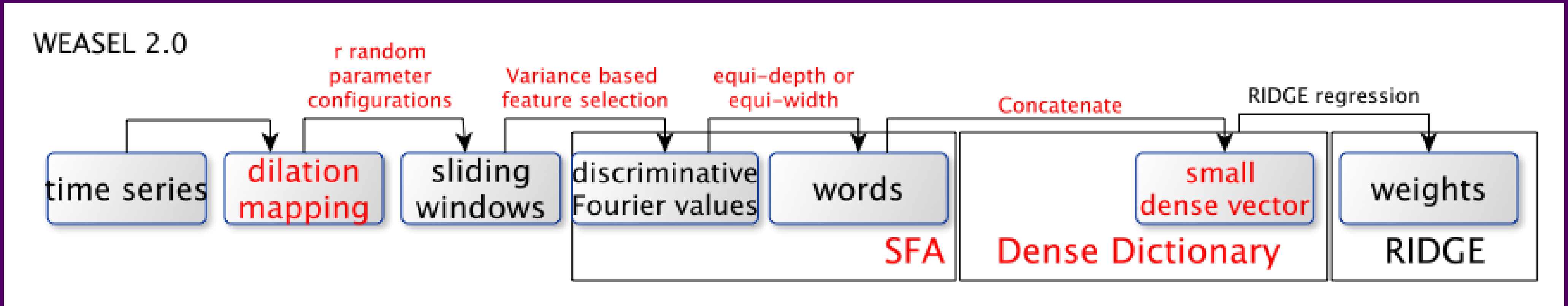
Classifiers built on histograms have historically been either nearest neighbour classifiers or linear classifiers

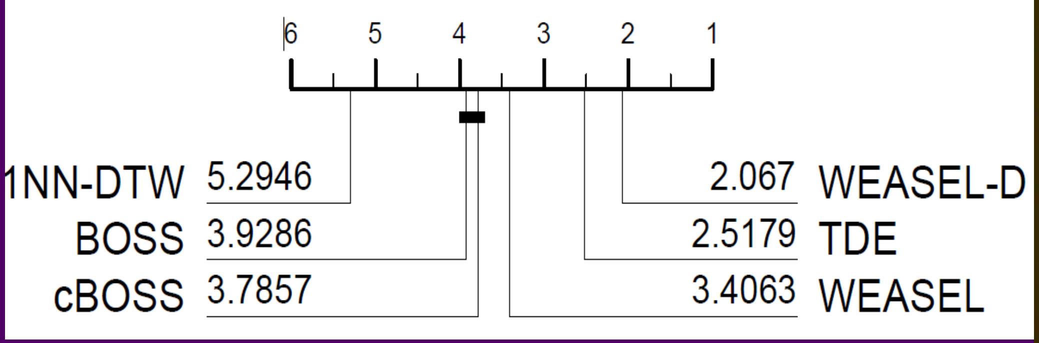




	Dictionary Based
BOSS	Bag of Symbolic Fourier Approximation Symbols [Schäfer 2015]
WEASEL	Word Extraction for Time Series Classification [Schäfer and Leser 2017a]
MUSE	Multivariate Symbolic Extension [Schäfer and Leser 2017b]
cBOSS	Contractable BOSS [Middlehurst et al. 2019]
S-BOSS	Spatial BOSS [Large et al. 2019a]
TDE	Temporal Dictionary Ensemble [Middlehurst et al. 2020b]
TEASER	Two-tier Early and Accurate Series Classifier [Schäfer and Leser 2020]

Best in class: WEASEL 2





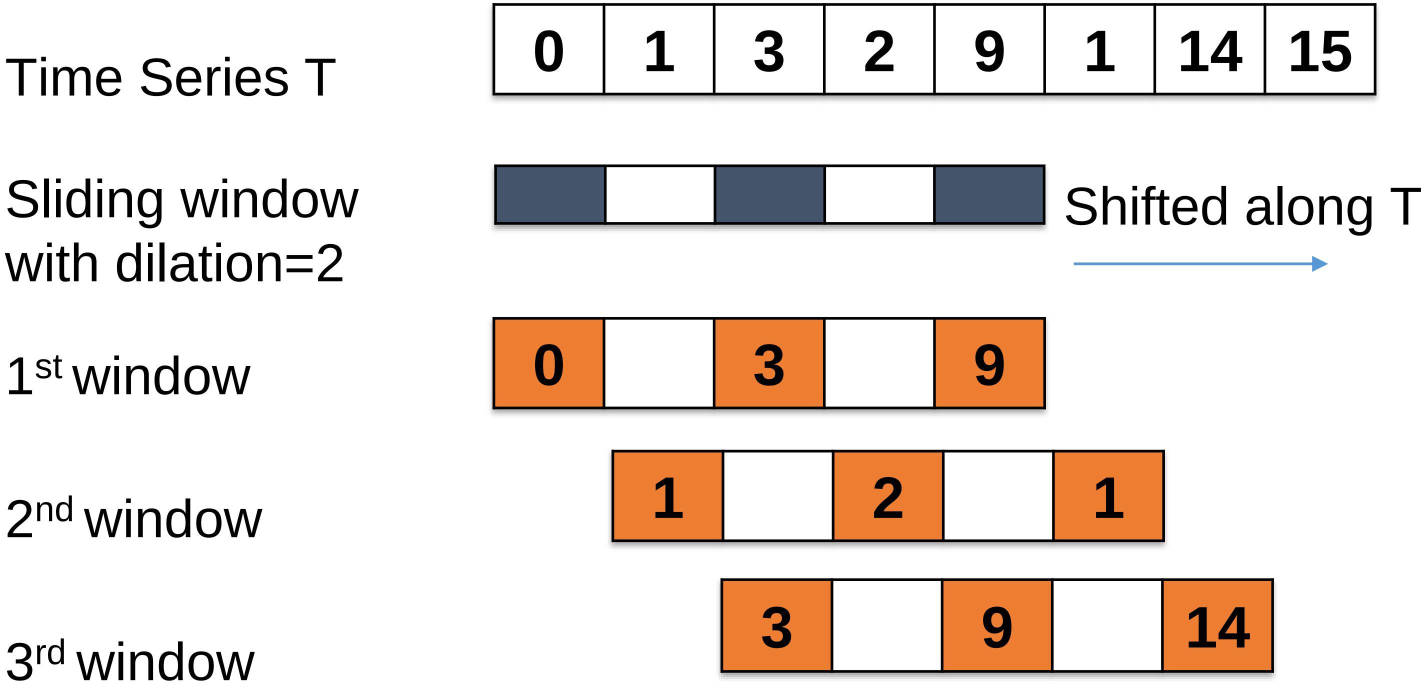

Home > Machine Learning > Article

WEASEL 2.0: a random dilated dictionary transform for fast, accurate and memory constrained time series classification

Open access | Published: 19 September 2023 | (2023)

Machine Learning

Dilated Sliding Window



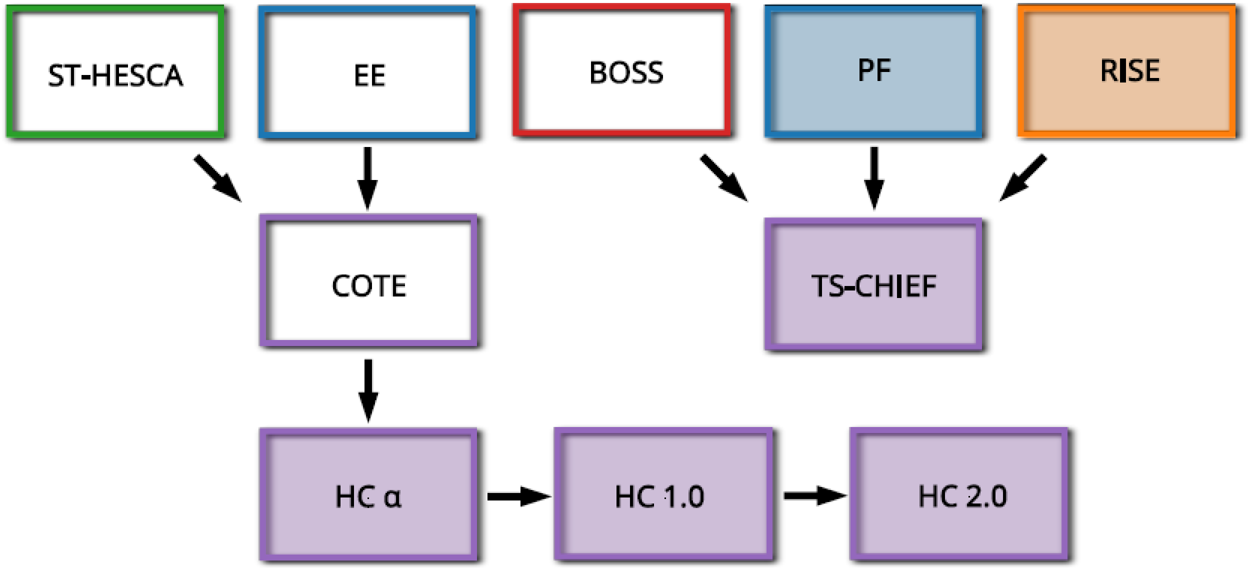
Example: d = 2 inserts gap of 1, down-samples by 2, doubles size of the receptive field

Improves:

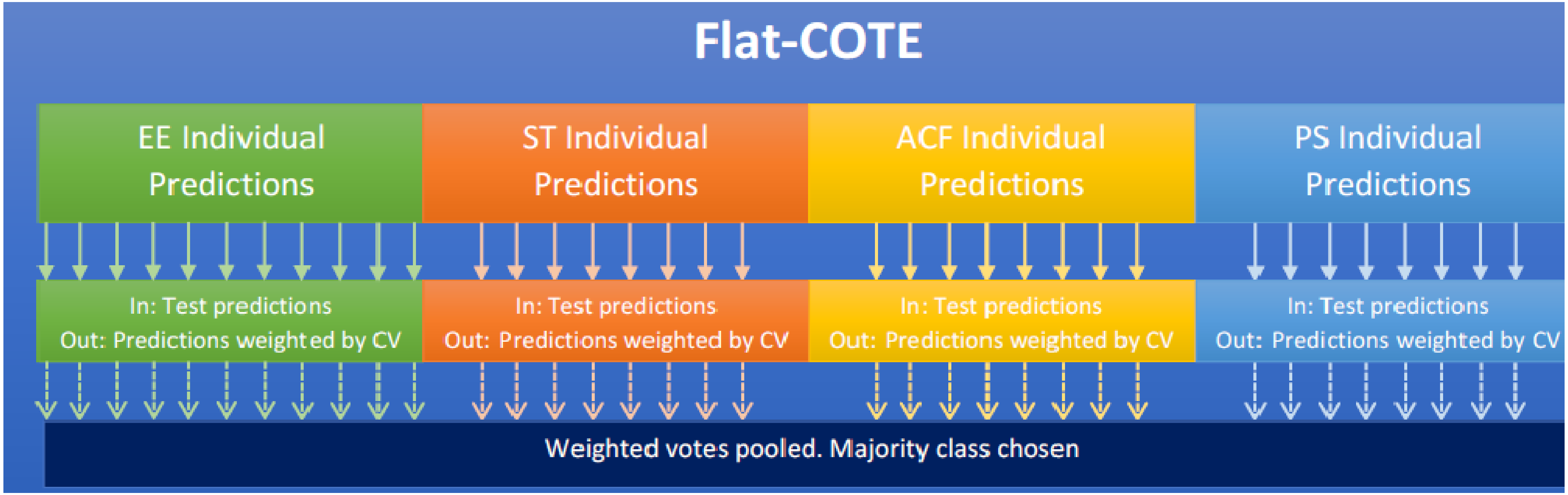
- Rocket (convolution based)
- RDST (shapelet based)
- WEASEL (dictionary based)

Hybrids: Combine Approaches

Meta ensembles explicitly combine classifiers built on different representations



Decision trees embed different representations internally at nodes



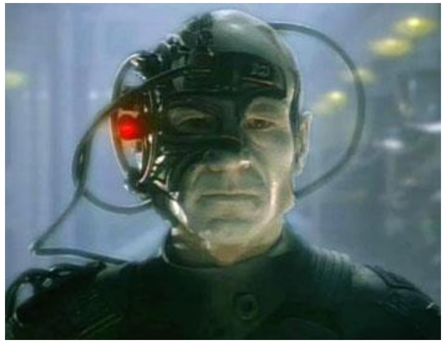
Journals & Magazines > IEEE Transactions on Knowledg... > Volume: 27 Issue: 9

Time-Series Classification with COTE: The Collective of Transformation-Based Ensembles

Publisher: IEEE Cite This PDF

Anthony Bagnall ; Jason Lines ; Jon Hills ; Aaron Bostrom **All Authors**

HIVE-COTE 1.0 (HC1)



Prediction: Class 2

CAWPE

Alpha (α) = 4	
Prob Class 1	$0.82^4 \times 0.1 + 0.59^4 \times 0.8 + 0.58^4 \times 0.7 + 0.6^4 \times 0.5 = 0.29/(0.29+0.44+0.09) = 0.35$
Prob Class 2	$0.82^4 \times 0.8 + 0.59^4 \times 0.1 + 0.58^4 \times 0.1 + 0.6^4 \times 0.4 = 0.44/(0.29+0.44+0.09) = \underline{0.54}$
Prob Class 3	$0.82^4 \times 0.1 + 0.59^4 \times 0.1 + 0.58^4 \times 0.2 + 0.6^4 \times 0.1 = 0.09/(0.29+0.44+0.09) = 0.11$
Prediction	**arg max(0.35, 0.54, 0.11)**

Shapelet Transform Classifier

Performance Estimate	82%
Prob Class 1	0.1
Prob Class 2	0.8
Prob Class 3	0.1
Prediction	**2**

Contractable Bag of SFA Symbols

Performance Estimate	59%
Prob Class 1	0.8
Prob Class 2	0.1
Prob Class 3	0.1
Prediction	**1**

Time Series Forest

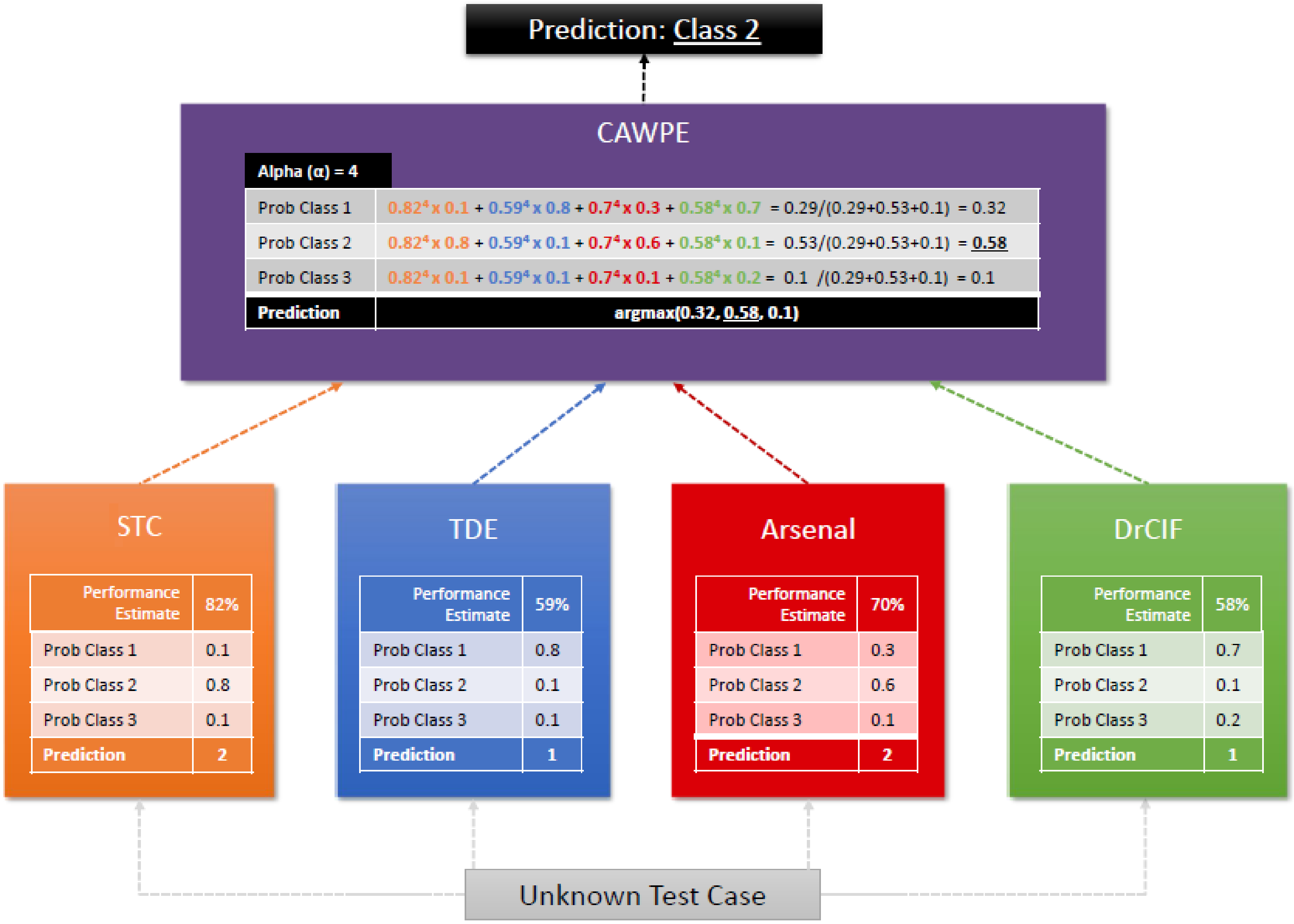
Performance Estimate	58%
Prob Class 1	0.7
Prob Class 2	0.1
Prob Class 3	0.2
Prediction	**1**

Random Interval Spectral Ensemble

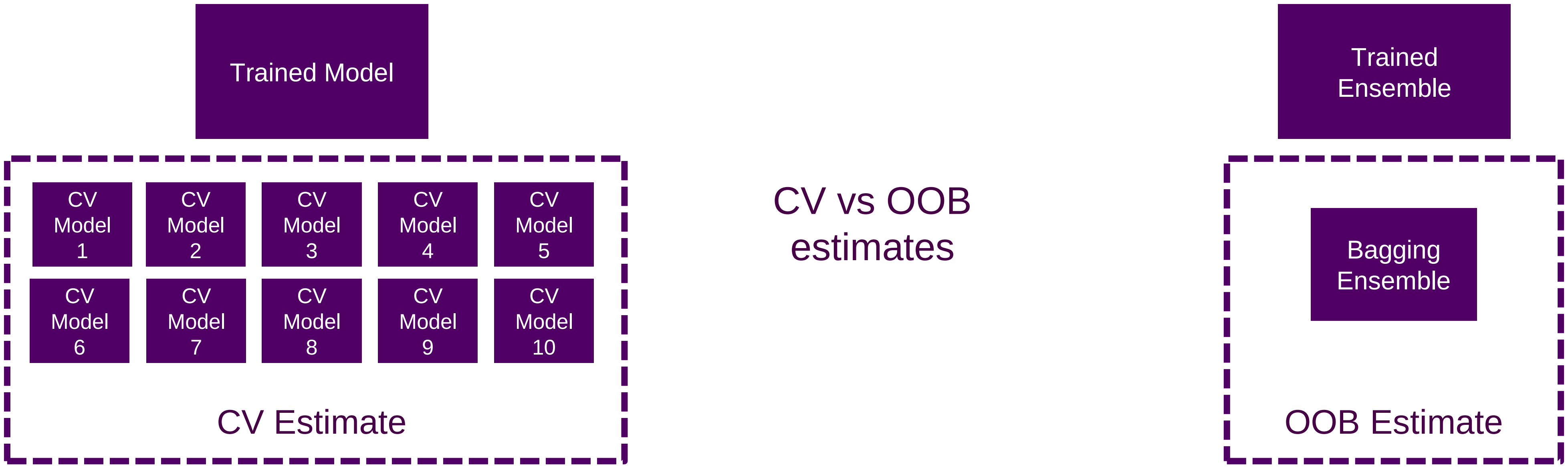
Performance Estimate	60%
Prob Class 1	0.5
Prob Class 2	0.4
Prob Class 3	0.1
Prediction	**1**

Unknown Test Case

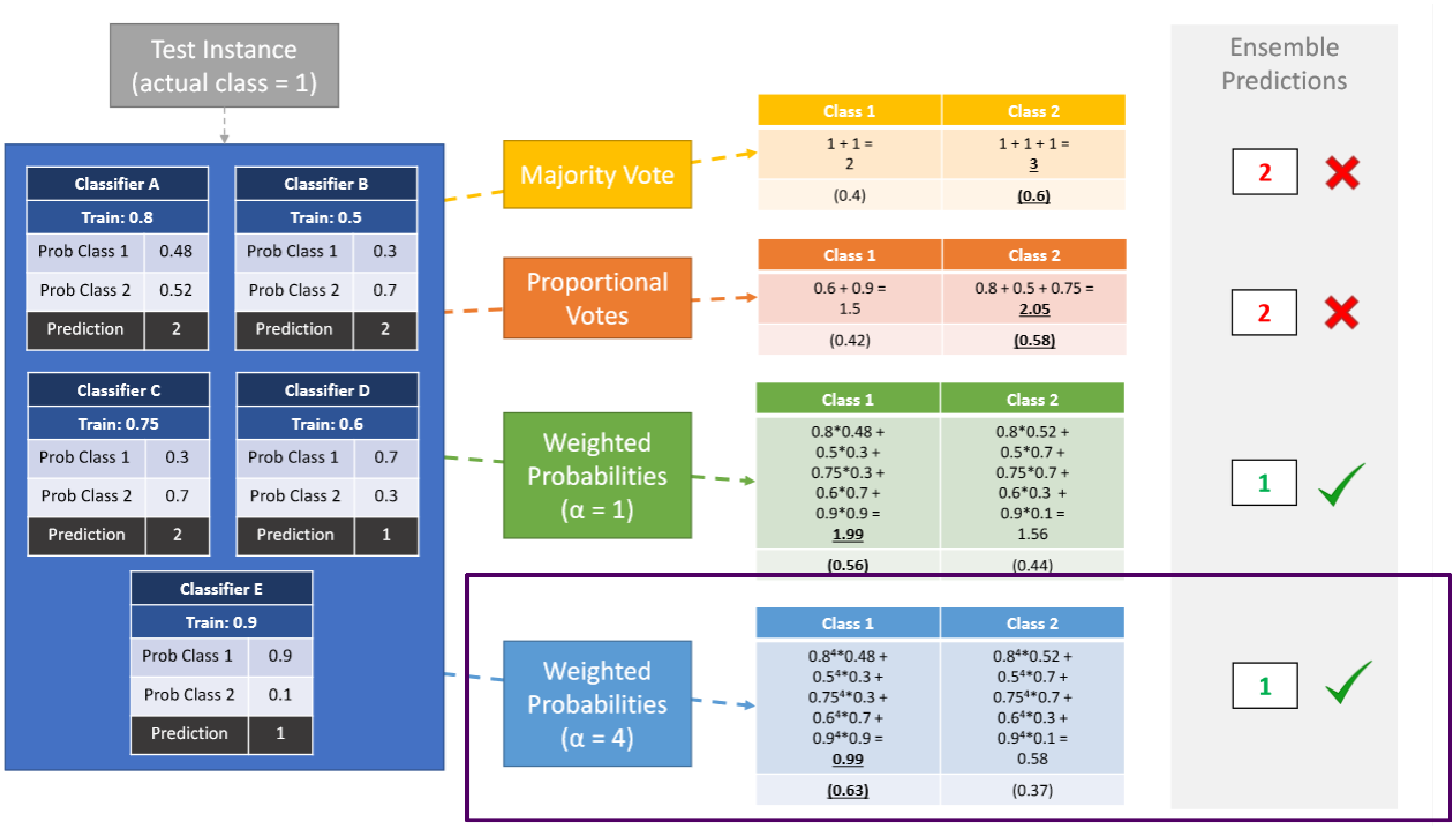
HIVE-COTE 2.0 (HC2)



Train Accuracy Estimate



HC2 Controller (CAWPE)

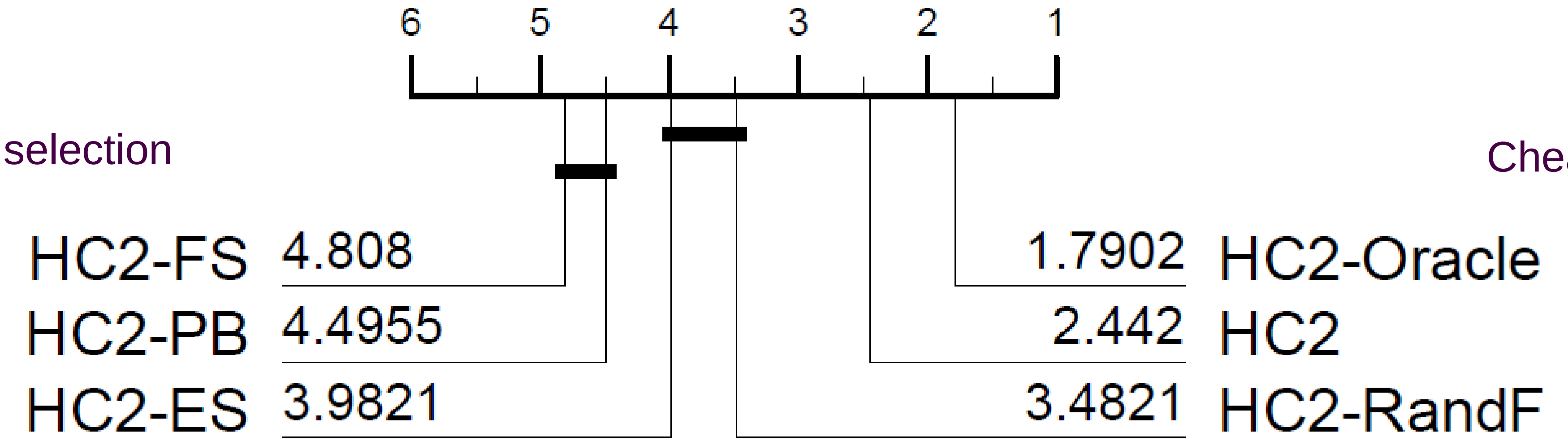


CAWPE

Alternate Ensemble Methods

Forward selection

Cheating

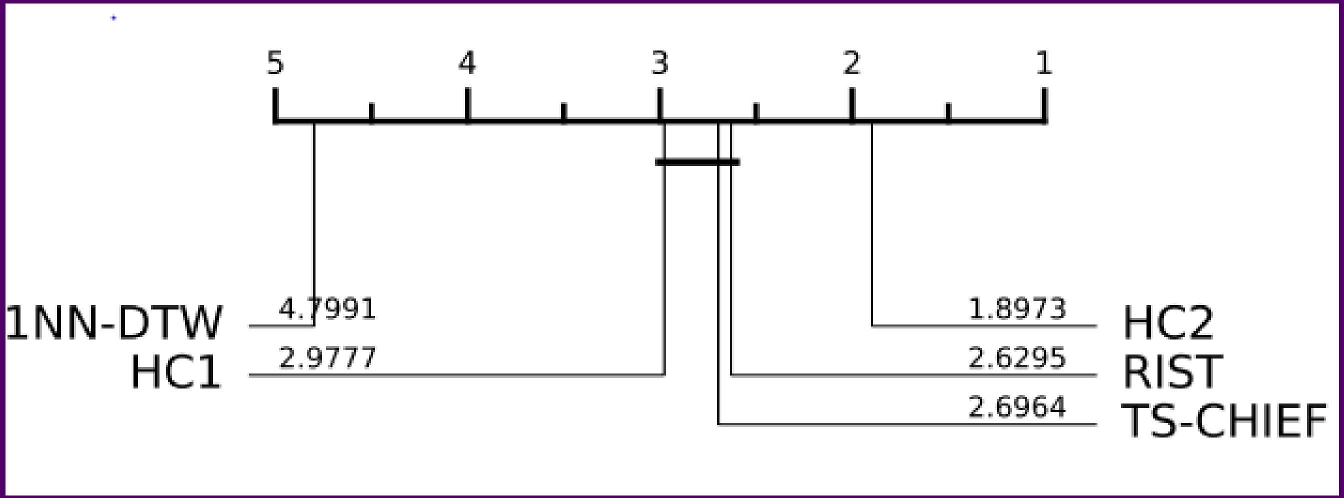


Pick best based on train accuracy

Selection and weighting based on a repeated bagging

Stacking with Random Forest

Best in class: HC2



Hierarchical Vote Collective of Transformation based Ensembles

COTE - Flat COTE: ensemble of EE and STC (2015)

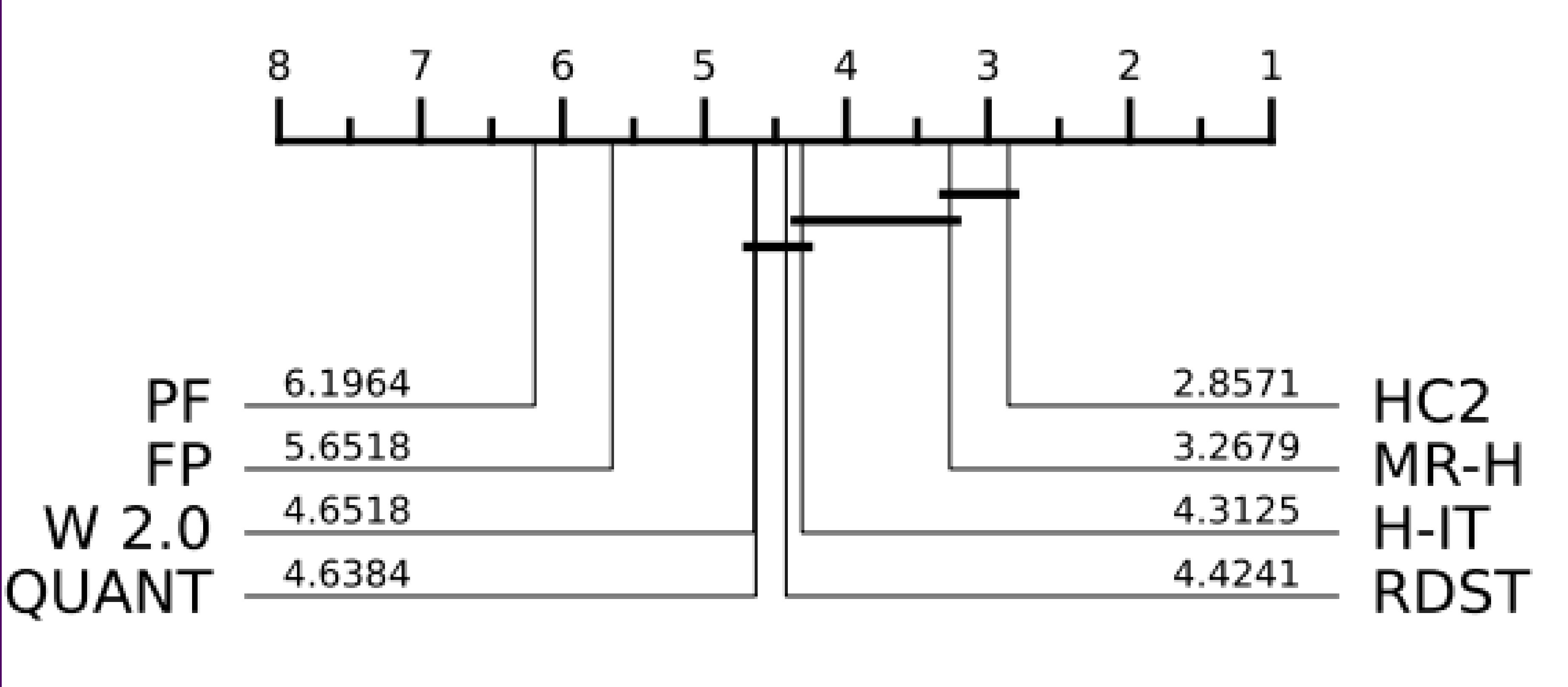
HIVE-COTE (alpha) - EE/STC/RISE/BOSS/TSF (2017)

HIVE-COTE V1 - STC/RISE/CBOSS/TSF (2020)

HIVE-COTE V2 – STC/The Arsenal/TDE/DrCIF (2021)

Bake off Redux: compare best in class

And the winner is ….



Taxonomy of Algorithms by Feature Type

Distance based

Dynamic time warping (DTW)

Elastic Ensemble (EE)

Proximity Forest (PF)

Interval based

Time Series Forest (TSF)

Random Interval Spectral Ensemble (RISE)

Canonical Interval Forest (CIF)

Diverse Representation CIF (DrCIF)

Deep Learning

CNN
FCN
LSTM
ResNet
TapNet
MCNN
CNTC
InceptionT
MCDCNN
MLP
MACNN
TLENET
TWIESN

Hybrids

TS-CHIEF BOSS EE RISE

HC1 ST cBOSS TSF RISE

HC2 STC TDE DrCIF Arsenal

Shapelet/kernel based

Shapelet Transform Classifier (ST)

ROCKET/MiniROCKET/ MultiROCKET/ Arsenal

Dictionary based

Bag of SFA Symbols (BOSS)

Contractable BOSS (cBOSS)

WEASEL

Temporal Dictionary Ensemble(TDE)

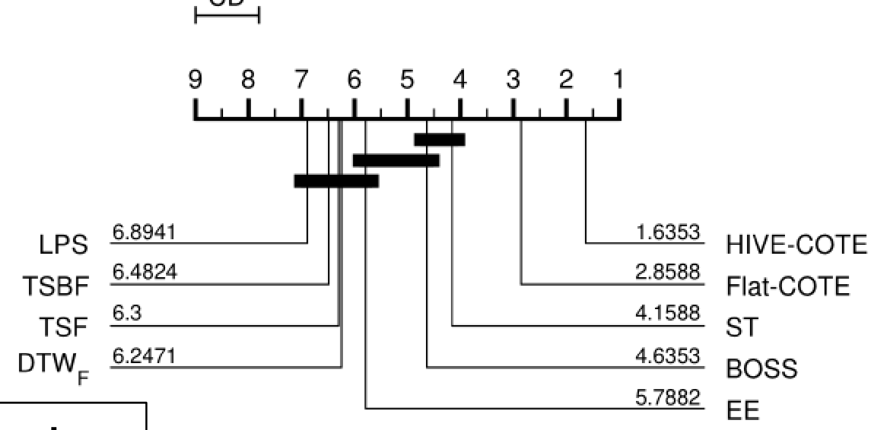
Time Series Classification (TSC) 2000-2018

Distance based:
1-NN DTW
“hard to beat”

Elastic Ensemble [1]

The collective of transformation-based ensembles [2] flat-COTE

Bake off [3] finds nine better than DTW, and Flat-COTE SOTA



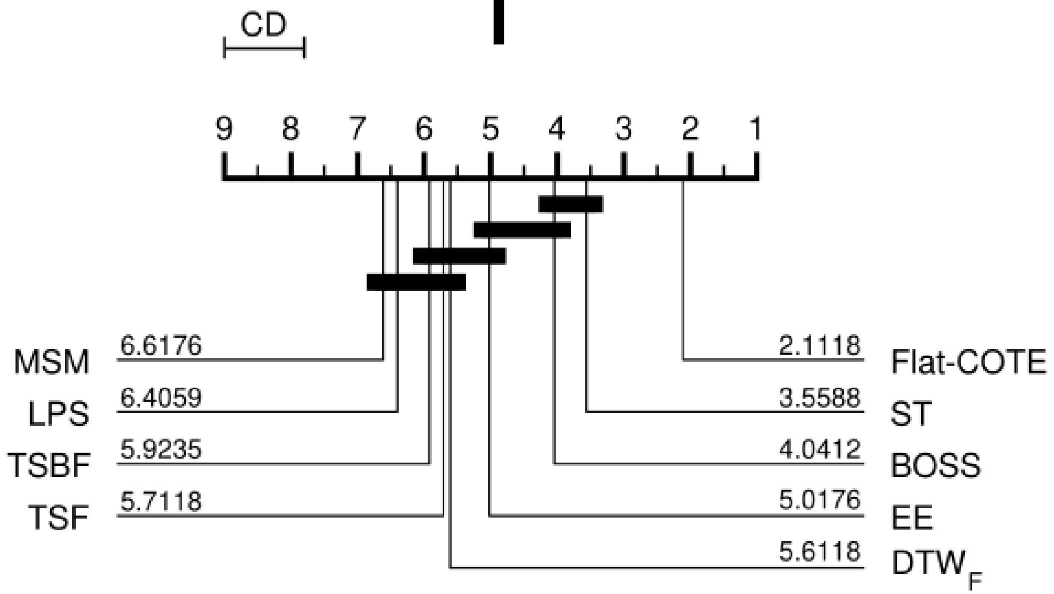
HIVE-COTE [4] new SOTA

Up to 2014 | 2015 | 2016 | 2017 | 2018

Feature based algorithms: BOP, BOSS, TSF, TSBF, DTWF, ST etc

UCR archive re-released with 85 datasets

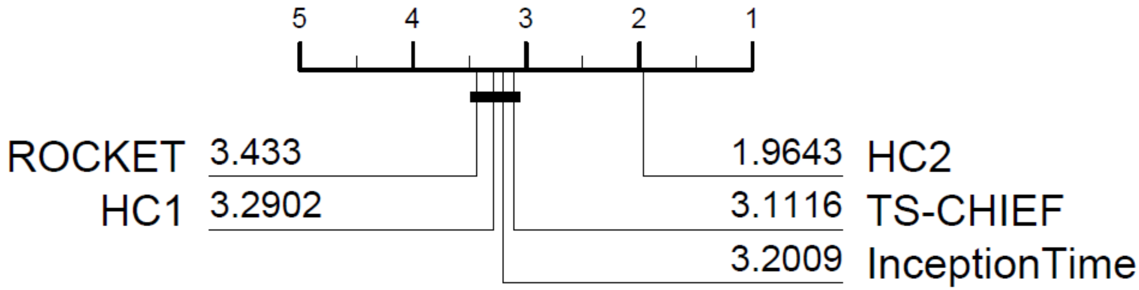
We conduct a comparative study (bake off) of 19 algorithms on 100 resamples of the new 85 datasets



TSC: recent progress 2018-2021

timeseriesclassification.com

Univariate (UCR archive 112)



UCR Archive relaunched with 128 problems [5]

Multivariate archive [10] (2020)

Kernel based ROCKET [8]

HIVE-COTE V2 (HC2) [11]

2019

2020

2021

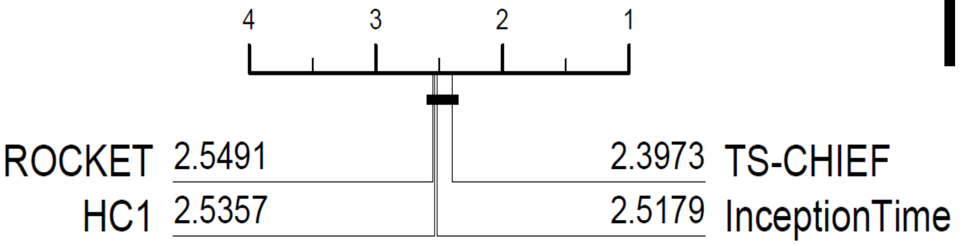
Deep learning bake off [6]

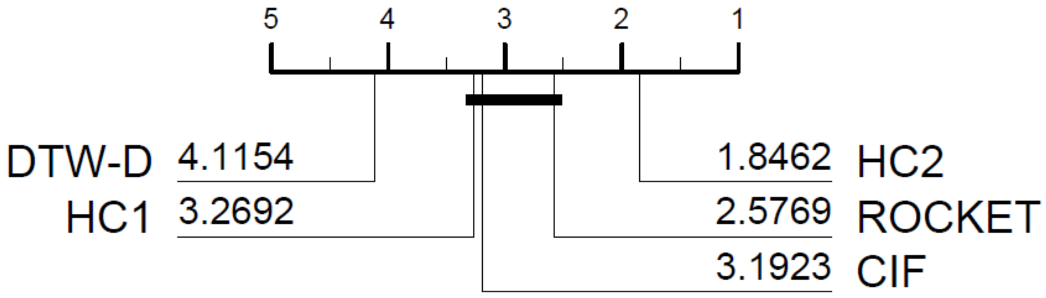
Tree based TS-CHIEF [7]

HIVE-COTE V1 (HC1) [9]

Multivariate bake off [12]

Deep learning: Inception Time [7]





Multivariate (UEA archive 26)

2022 onwards

timeseriesclassification.com

MultiRocket

MiniRocket

Hydra

QUANT

Bake off redux

30 new archive problems

2022

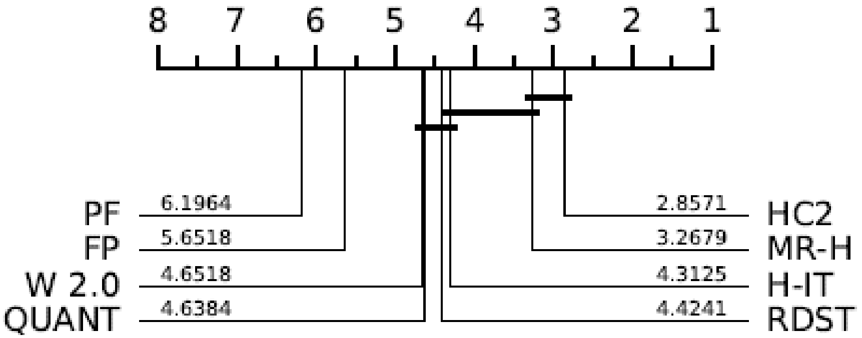
2023

2024

RDST

FreshPRINCE

WEASEL v2.0

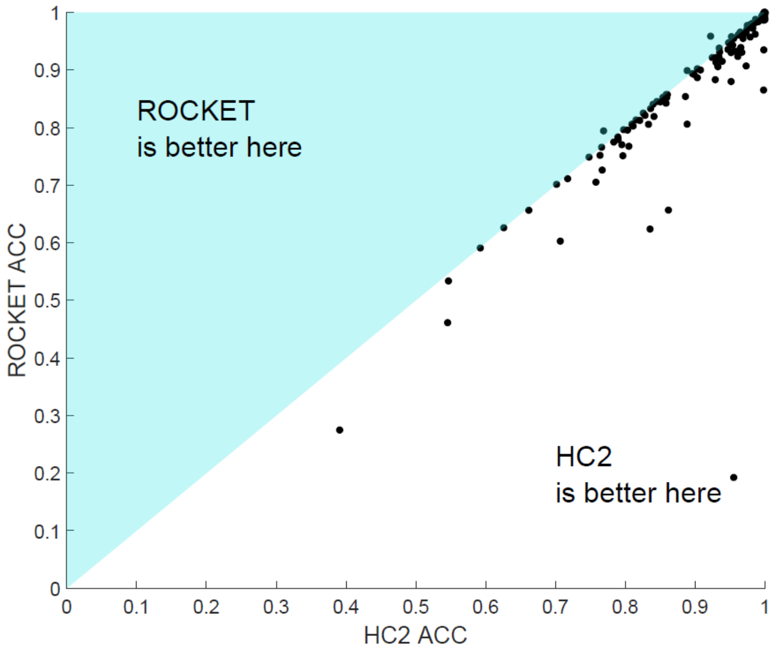


Explainable, ordinal, multivariate, channel selection, scalable, streaming, applications

And of course a LOT of deep learning research ….. often self contained

Is the Progress Real? Performance on UCR datasets vs HC2

Algorithm	Mean Accuracy Diff	HC2 Wins	HC2 Loses
HC1	-1.06%	77	29
ROCKET	-2.49%	74	32
InceptionTime	-1.69%	82	25
TS-CHIEF	-1.36%	97	11

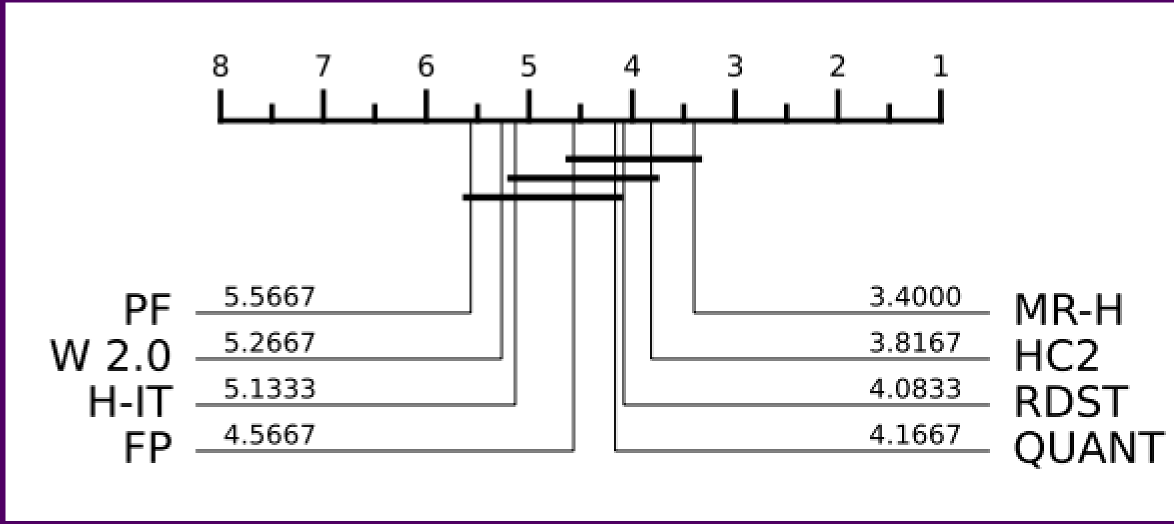


HC2 average accuracy is 89.11%

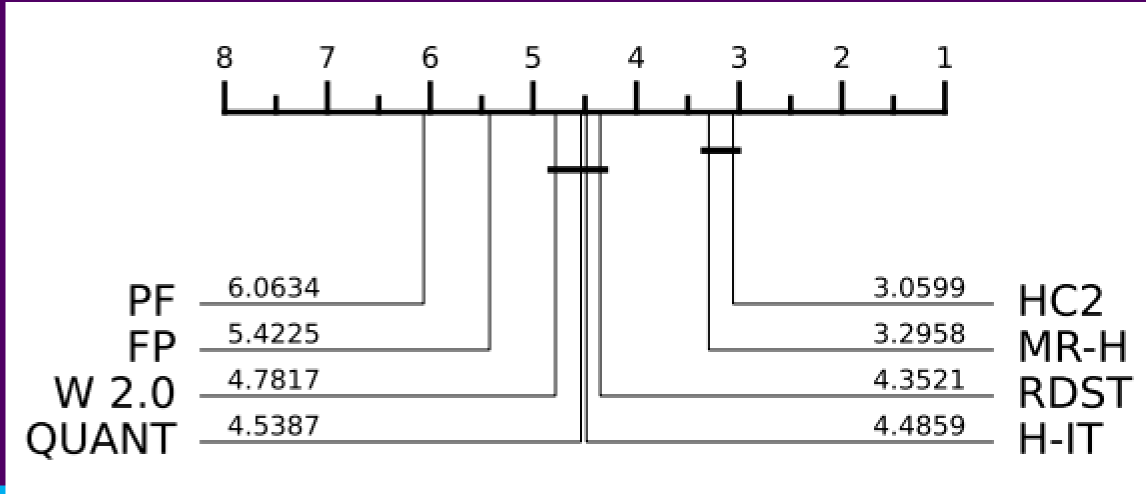
On average, over 112 UCR problems, HC2 is 12.37% more accurate than 1-NN DTW (wins on 107, ties on 2, loses on 3)

Are we all just overfitting the UCR archive?

30 new datasets



142 old+new datasets



Is the Progress Real? Case Study: Detecting Fraudulent Alcohol

Can we detect the methanol contents of spirits bottles non-invasively?

Algorithm	Accuracy and standard error (LOBO)
HC2	63.82% +/- 2.72%
ROCKET	52.72% +/- 2.69%
InceptionTime	44.93% +/- 2.67%
PLS	13.14% +/- 0.9%

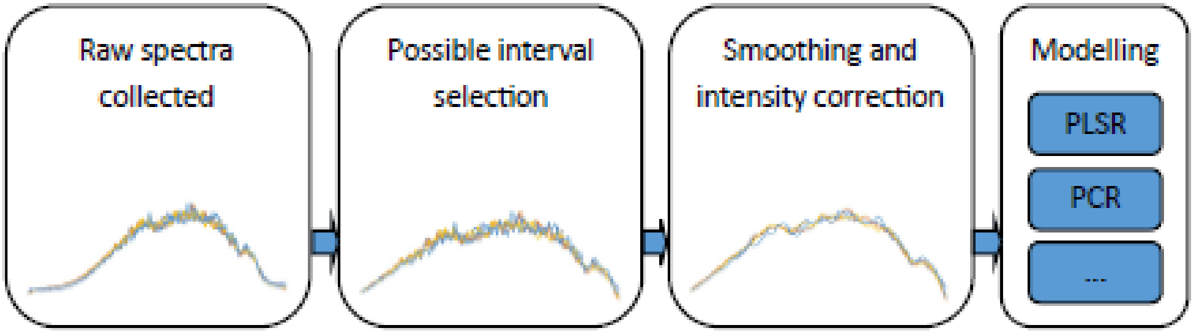
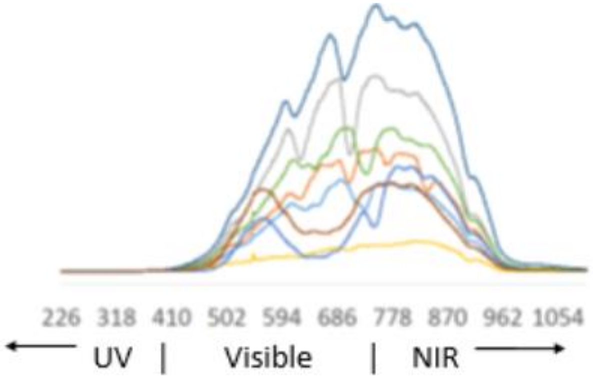


Fig. 2.3 An overview of a standard chemometric pipeline, applied to example spectra.

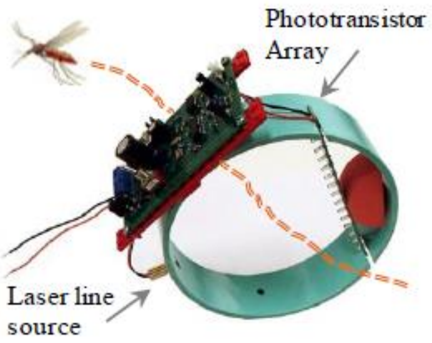


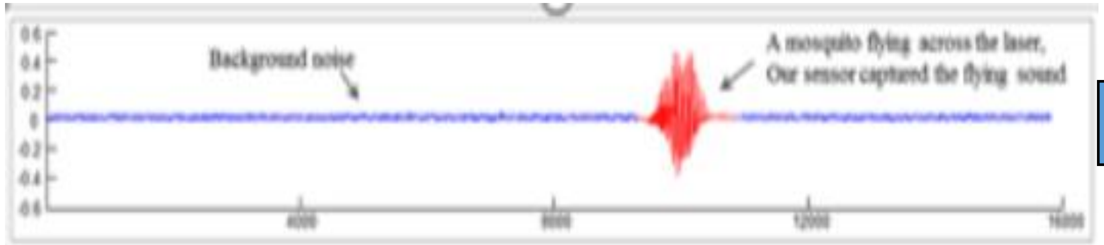
THE Scotch Whisky RESEARCH INSTITUTE

Application 2: Insect Classification

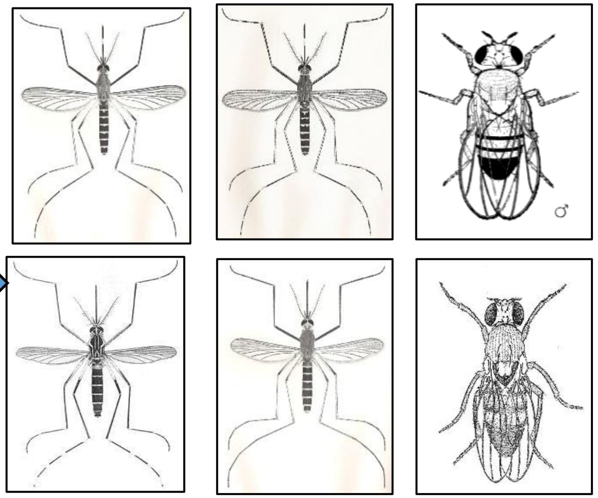
icipe
INTERNATIONAL CENTRE OF INSECT PHYSIOLOGY AND ECOLOGY
African Insect Science for Food and Health

UC RIVERSIDE
UNIVERSITY OF CALIFORNIA

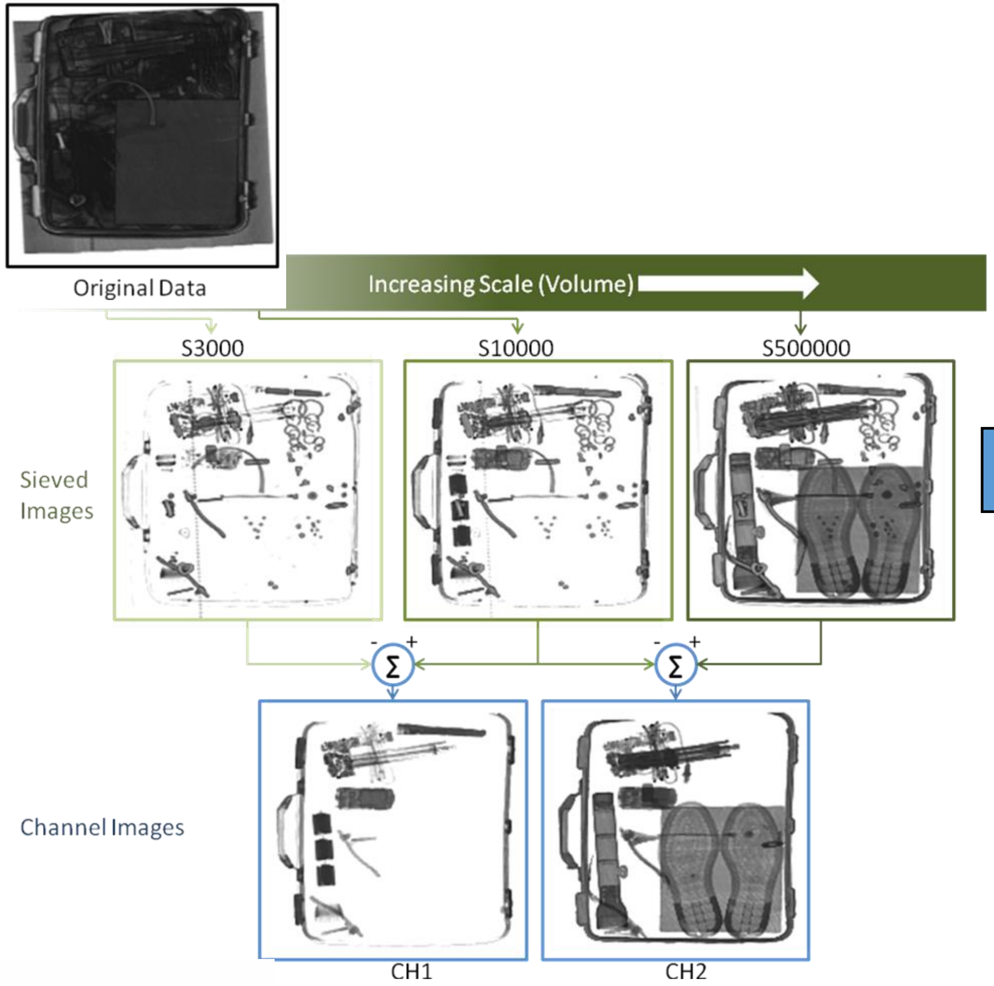




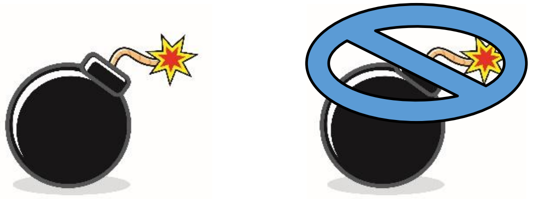
?



Application 3: Detecting Explosives in Electric Devices



feature maps to time series



Engineering and Physical Sciences Research Council

Defence and Security Accelerator

aeon tsml toolkit

https://www.aeon-toolkit.org/

- aeon
  - anomaly_detection
  - base
  - benchmarking
  - classification
  - clustering
  - datasets
  - datatypes
  - distances
  - forecasting
  - local
  - networks
  - performance_metrics
  - pipeline
  - registry
  - regression
  - segmentation
  - similarity_search
  - testing
  - transformations
  - utils
  - visualisation

The Alan Turing Institute

NUMFOCUS

OPEN CODE = BETTER SCIENCE

aeon-toolkit.org/en/latest/

aeon

Search

USING AEON

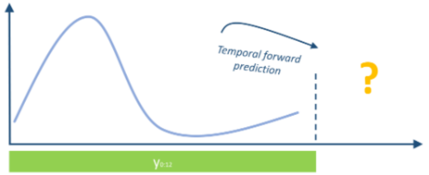
Installation

Getting Started

Welcome to aeon

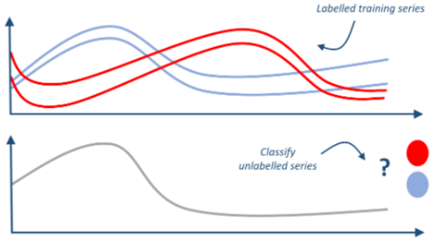
`aeon` is a scikit-learn compatible toolkit for time series tasks such as forecasting, classification and clustering.

- Provides a broad library of time series algorithms, including the latest advances.
- Efficient implementation of time series algorithms using numba.
- Interfaces with other time series packages to provide a single framework for algorithm comparison.



Get started with time series forecasting.

Forecasting



Get started with time series classification.

Classification

EPSRC

Pioneering research and skills

Engineering and Physical Sciences Research Council

GoW Search

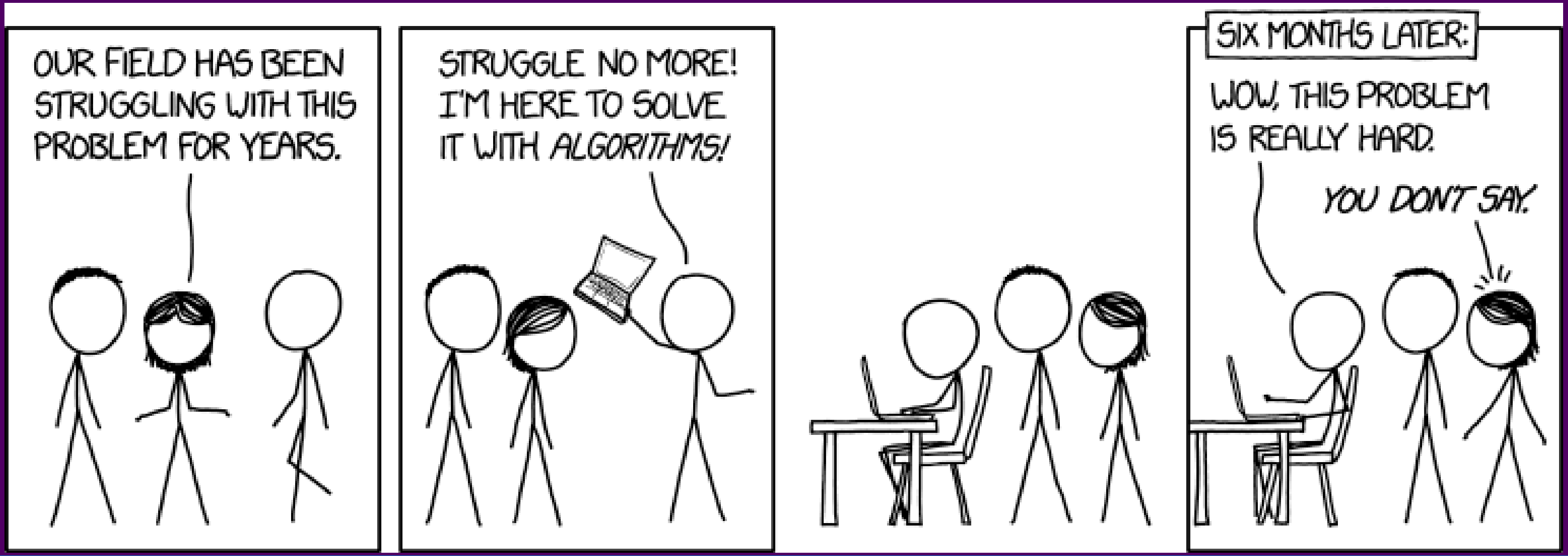
Home | GoW Home | Back | Research Areas | Topic | Sector | Scheme | Region | Theme | Organisation | Partners

Details of Grant

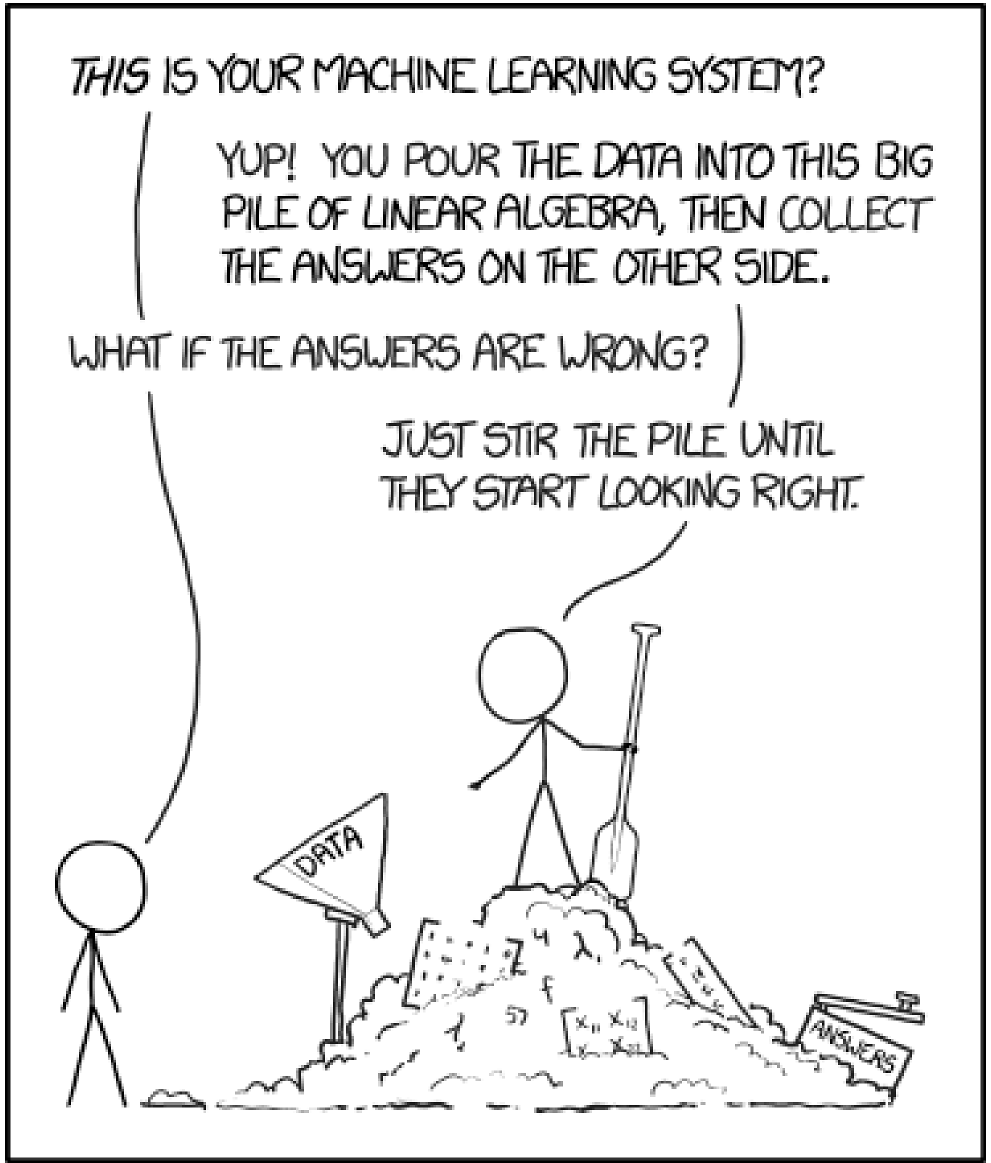
EPSRC Reference:	**EP/W030756/2**		
Title:	**aeon: a toolkit for machine learning with time series**		
Principal Investigator:	**Bagnall, Professor A**		
Other Investigators:	**Sambrook, Dr TD**	**Sami AK, Dr S**	**Renoult, Dr L**
Researcher Co-Investigators:			
Project Partners:	**GlaxoSmithKline plc (GSK)**	**Mercedes-Benz AG**	**Monash University**
	The Alan Turing Institute	**UCL**	**University of California Riverside**
	University of Cambridge		
Department:	**Electronics and Computer Science**		
Organisation:	**University of Southampton**		
Scheme:	**Standard Research**		

What classifier to use for a new TSC problem?

What is the best approach with no prior knowledge?
What is the best approach for a specific problem domain?
What is the best approach for a specific data set?



Thank you for listening,
any questions?



TSC References

[1] J. Lines and A. Bagnall. Time series classification with ensembles of elastic distance measures. *Data Mining and Knowledge Discovery*, 29:565–592, 2015.

[2] A. Bagnall, J. Lines, J. Hills, and A. Bostrom. Time-series classification with COTE: The collective of transformation-based ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 27:2522–2535, 2015.

[3] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 31(3):606–660, 2017.

[4] J. Lines, S. Taylor, and A. Bagnall. Time series classification with HIVE-COTE: The hierarchical vote collective of transformation-based ensembles. *ACM Transactions Knowledge Discovery from Data*, 12(5):1–36, 2018.

[5] H. Dau, A. Bagnall, K. Kamgar, M. Yeh, Y. Zhu, S. Gharghabi, C. Ratanamahatana, A. Chotirat, and E. Keogh. The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, 6(6):1293–1305, 2019.

[6] H. Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.

[7] Ahmed Shifaz, Charlotte Pelletier, François Petitjean, and Geoffrey I Webb. TS-CHIEF: a scalable and accurate forest algorithm for time series classification. *Data Mining and Knowledge Discovery*, 34(3):742–775, 2020.

[8] Angus Dempster, François Petitjean, and Geoffrey Webb. ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34:1454–1495, 2020.

[9] A. Bagnall, M. Flynn, J. Large, J. Lines, and M. Middlehurst. On the usage and performance of HIVE-COTE v1.0. In *proceedings of the 5th Workshop on Advanced Analytics and Learning on Temporal Data*, volume 12588 of *Lecture Notes in Artificial Intelligence*, 2020.

[10] A. Pasos Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 35(2):401—449, 2021.

[11] Matthew Middlehurst, James Large, Michael Flynn, Jason Lines, Aaron Bostrom, and Anthony Bagnall. HIVE-COTE 2.0: a new meta ensemble for time series classification. *Machine Learning*, 110:3211–3243, 2021.

[12] Angus Dempster, D Schmidt, and Geoffrey Webb. Minirocket: A very fast (almost) deterministic transform for time series classification. In *proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2021.

[13] Chang Wei Tan, Angus Dempster, Christoph Bergmeir, and Geoffrey Webb. MultiRocket: multiple pooling operators and transformations for fast and effective time series classification. *Data Mining and Knowledge Discovery*, Online first:3211–3243, 2022.

[14] Angus Dempster, Daniel F Schmidt, and Geoffrey I Webb. HYDRA: Competing convolutional kernels for fast and accurate time series classification. *arXiv preprint arXiv:2203.13652*, 2022.