VistaMilk

A World Leading SFI Research Centre

Science Foundation Ireland SFI For what's next

An Examination of the State-of-the-Art for Multivariate Time Series Classification

Bhaskar Dhariyal, Thach Le Nguyen, Severin Gsponer and Georgiana Ifrim

School of Computer Science, University College Dublin, Ireland

bhaskar.dhariyal@insight-centre.org

HOST INSTITUTION

eagasc Agriculture and Food Development Authority

PARTNER INSTITUTIONS

DCU, UCD Dublin, NUI Galway OE Gaillimh, Waterford Institute of Technology, Tyndall National Institute, ICBF.com, UCC University College Cork Ireland, TSSG, Insight

FUNDED BY:

An Roinn Talmhaíochta, Bia agus Mara Department of Agriculture, Food and the Marine; Ireland's European Structural and Investment Funds Programmes 2014-2020 Co-funded by the Irish Government and the European Union; European Union European Regional Development Fund; Science Foundation Ireland SFI For what's next

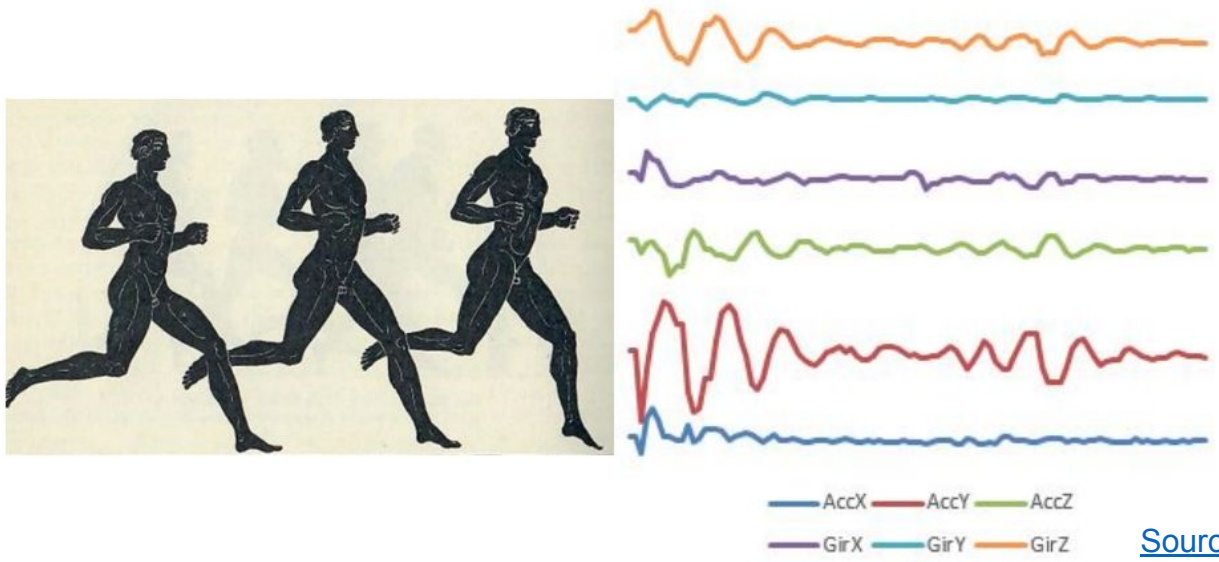
Outline

- Introduction
- Motivation
- Related Work
- Experiments
- Conclusion

Introduction

Multivariate Time Series Classification: The task of classifying an ordered temporal signal (time series) where multiple feature sources can affect the class outcome.

Below is an example time series from the Basic Motions dataset from UEA archive. The class could be: standing, walking, running or playing badminton. Each example consists of multiple time series (aka dimensions or channels).



Source

Motivation

- Less research work done on MTSC, most advances so far are in univariate TSC.
- Most MTSC problems are approached using UTSC methods
- To apply UTSC methods we can:
  - concatenate the time series dimensions
  - consider each dimension as separate (univariate) TS and collect features over the dimensions or ensemble univariate classifier predictions.
- Issue with above-mentioned methods: complex dependencies between dimensions could be missed

- **Our Goal:** Perform an empirical study of recent state-of-the-art MTSC methods on the new UEA MTSC benchmark

Motivation

- Data Availability
- Wide Range of Applications:
  - Medical
  - Sports Science
  - Agriculture
  - Manufacturing,...

- **Three Computation Challenges of MTSC**
  - Scale: Number of time series (e.g., How many people wearing sensors?)
  - Dimensions: Number of channels for each time series, dependencies between dimensions (e.g., How many sensors? How do they interact?)
  - Length: Length of the time series (e.g., What is the time duration for data collection?)

Contributions

- Empirical study of state-of-the-art MTSC methods with respect to accuracy and computation time.
- Propose a simple MTSC baseline, more accurate and faster than classic DTW.
- Simple linear classifier with stats features as accurate as complex deep learning methods.

Related Work

UTSC Methods	MTSC Methods
Distance based (1NN-DTW)	1NN-DTW-D
Shapelet-based (Shapelet Transform, Learning Shaplets)	MrSEQL
Symbol-based (SAX-VSM, BOSS, WEASEL, MrSEQL)	WEASEL-MUSE
Ensemble Classifiers (EE, COTE, HIVE-COTE, TS-CHIEF)	ROCKET
Deep Learning (Resnet, FCN)	TapNet
ROCKET	MLSTM-FCN

UEA MTSC Benchmark: Problem Types (30 datasets)

http://www.timeseriesclassification.com

Human Activity Recognition • BasicMotions • Cricket • Epilepsy • ERing • Handwriting • Libras • NATOPS • UWaveGestureLibrary	**Motion Classification** • ArticularyWordRecognition • CharacterTrajectories • EigenWorms • PenDigits	**ECG classification** • AtrialFibrillation • StandWalkJump
EEG/MEG • FingerMovements • MotorImagery • SelfRegulationSCP1 • SelfRegulationSCP2 • FaceDetection • HandMovementDirection	**Audio Spectra Classification** • DuckDuckGeese • Heartbeat • InsectWingbeat • Phoneme • SpokenArabicDigits • JapaneseVowels	**Others** • EthanolConcentration • PEMS-SF • LSST

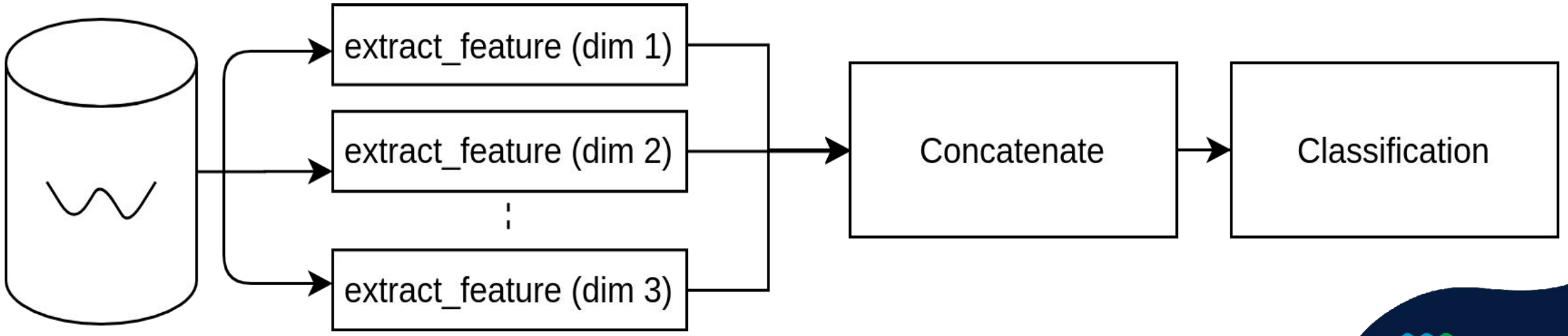
Source

UEA MTSC Benchmark: Data-Dictionary & Challenges

Problem	TrainSize	TestSize	NumDimensions	SeriesLength	NumClasses
InsectWingbeat	**30000**	**20000**	**200**	30	10
MotorImagery	278	100	64	**3000**	2
PEMS-SF	267	173	**963**	144	7
SpokenArabicDigits	**6599**	**2199**	13	93	10
StandWalkJump	12	15	4	**2500**	3
DuckDuckGeese	50	50	**1345**	270	5
EigenWorms	128	131	6	**17984**	5
FaceDetection	5890	3524	**144**	62	2

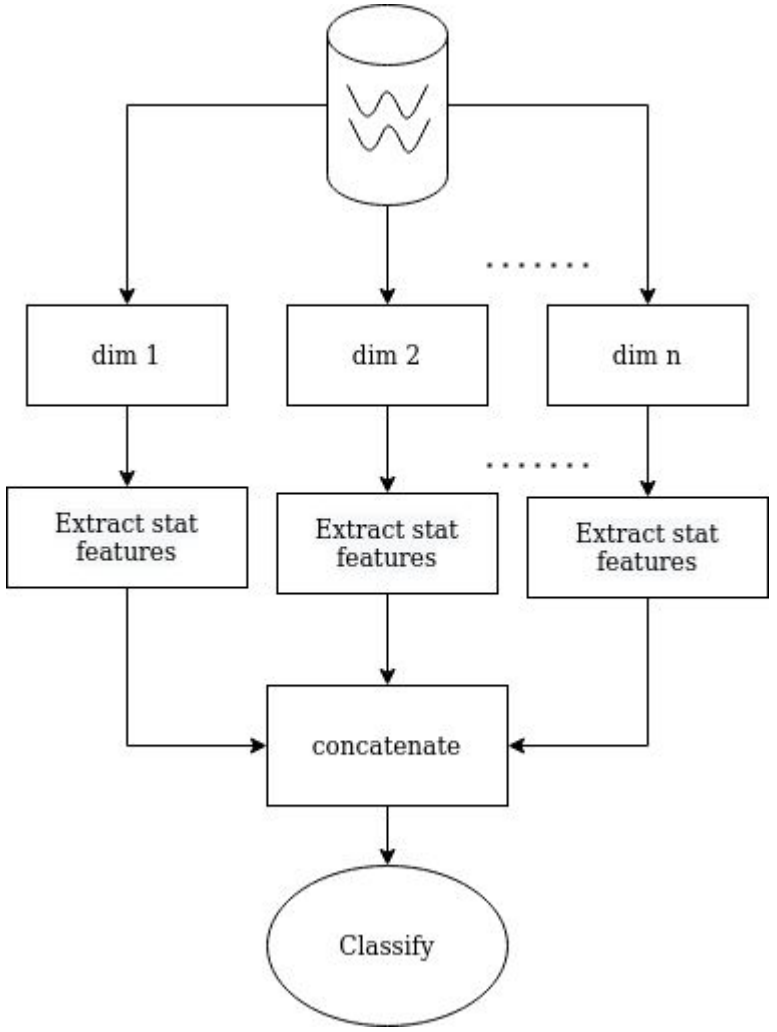
Methods

- **<u>Fundamental Principle:</u>** Extract (stats) features from each dimension and concatenate them in the same feature vector.
- The concatenation will enable the classifier to learn dependencies across dimensions.
- The extraction of features is parallelizable.
- **<u>Classifier</u>:** We use Ridge Regression to keep it simple and scalable.



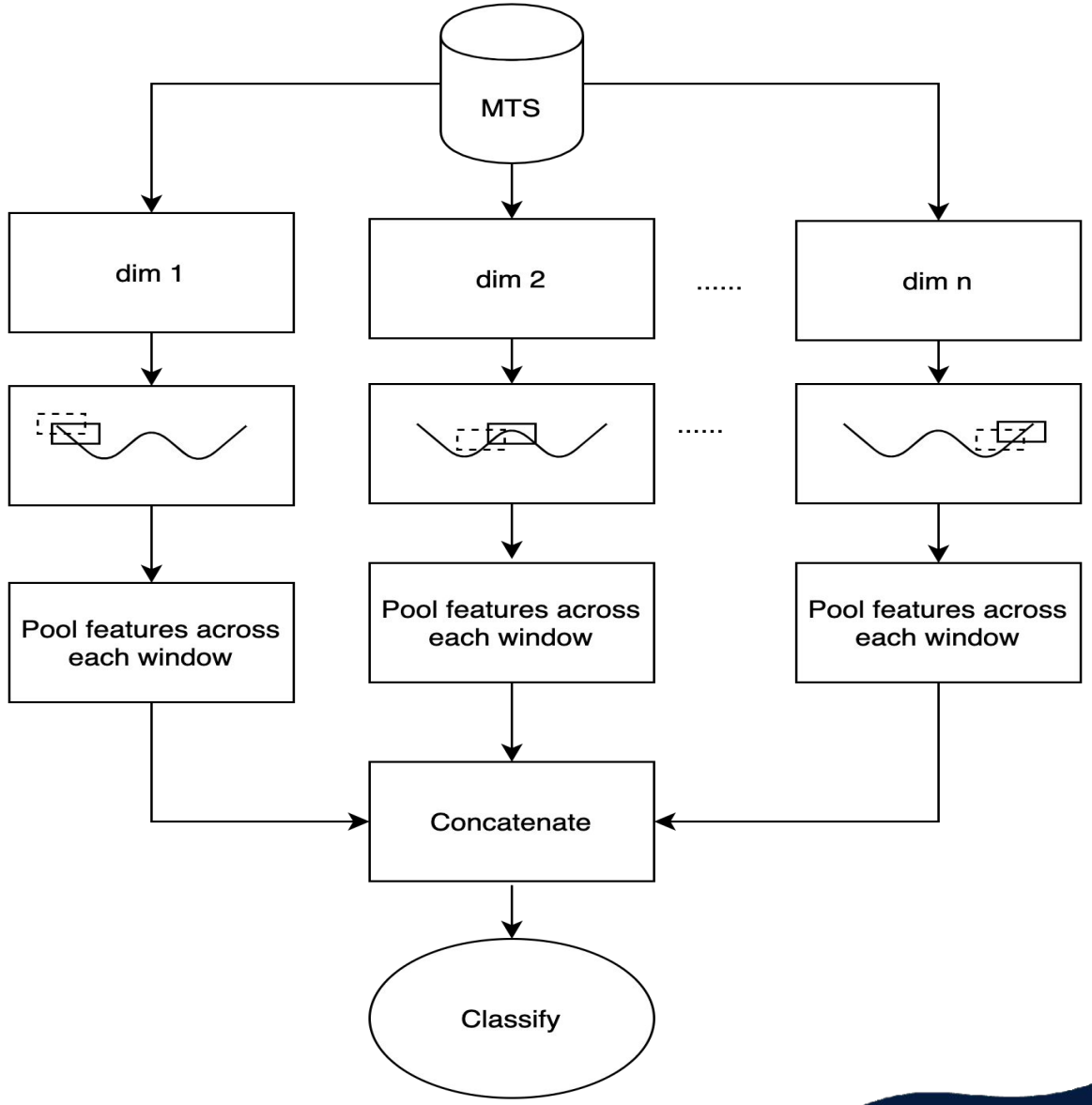
Methods: Raw Data and Stats-based Features

- From each dimension, extract 3 sets of statistics as features, resulting in 3 methods:
  - **4-stat:** 4 features (mean, min, max, std dev)
  - **9-stat:** To above, add median, kurtosis, number of peaks, 25 and 75 percentile.
  - **Catch22** features [Link]
- Concatenate all the features into a single feature vector.
- Train Classifier



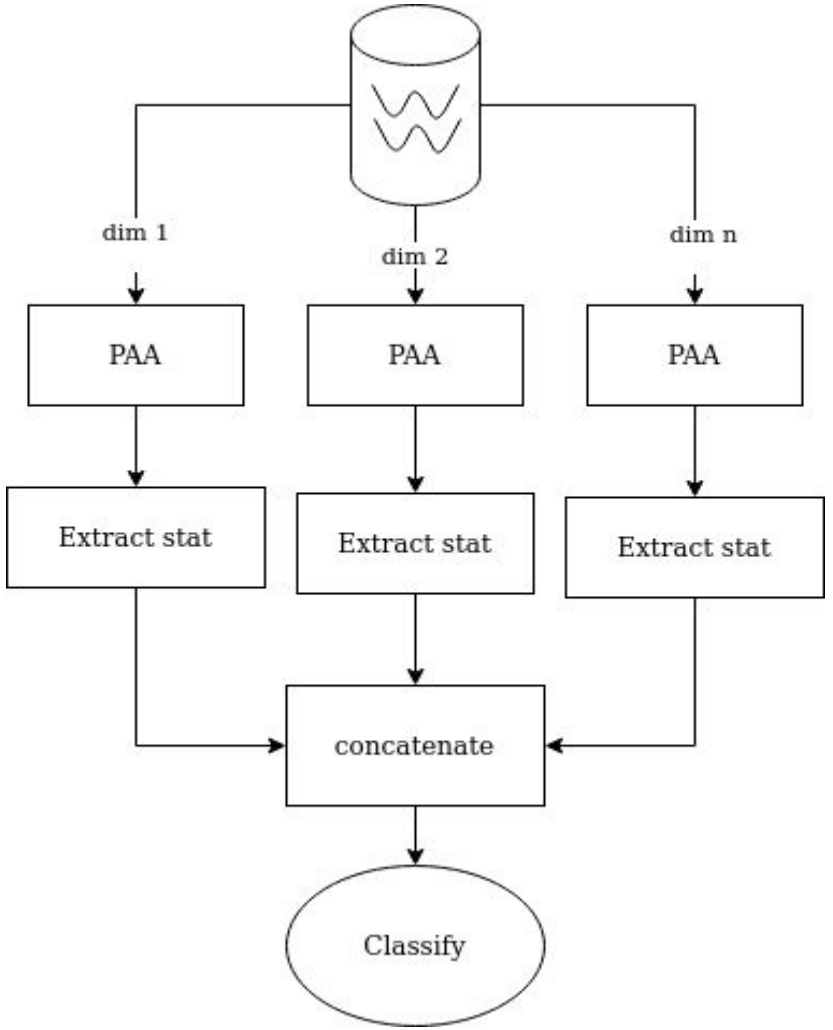
Methods: Raw Data and Window-based Features

- Extract features from every window in each dimension.
- Perform average pooling over features for each dimension.
- Concatenate features across dimensions and classify.



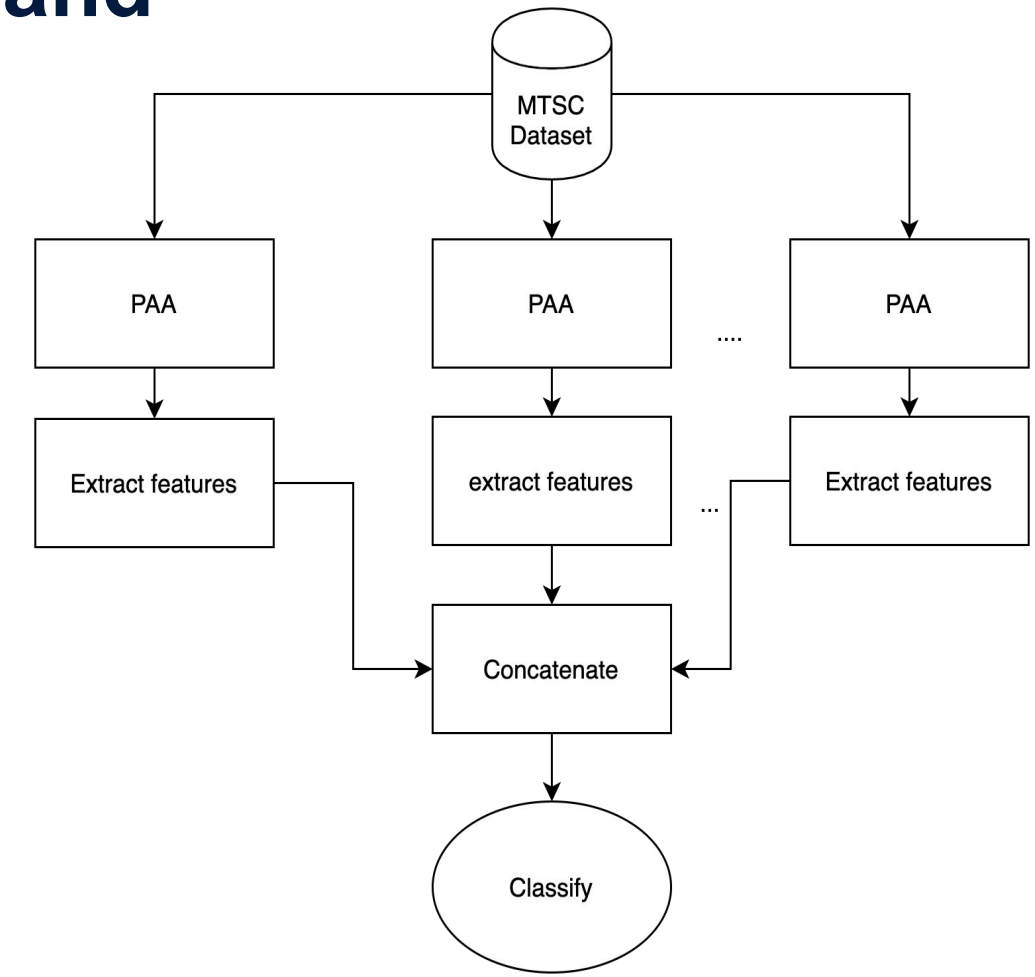
Methods: PAA Data and Stats-based Features

- PAA shrinks and smooths the raw time series [Link]
- Main PAA parameter is the output length (e.g., PAA_30 shrinks the time series to 30% of its original length)
- Perform PAA on each dimension of the time-series.
- Extract all 3 sets of features (4 stat, 9 stat and catch22) from each dimension.
- Concatenate features and classify.



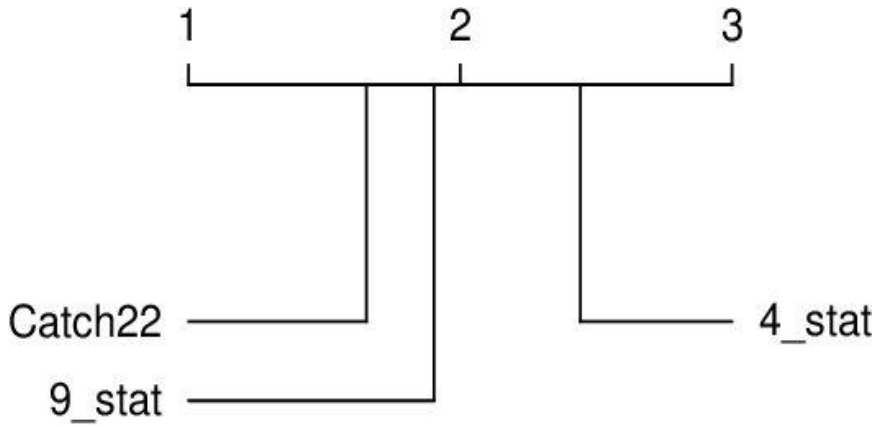
Methods: Multiple PAA Data and Stats-based Features

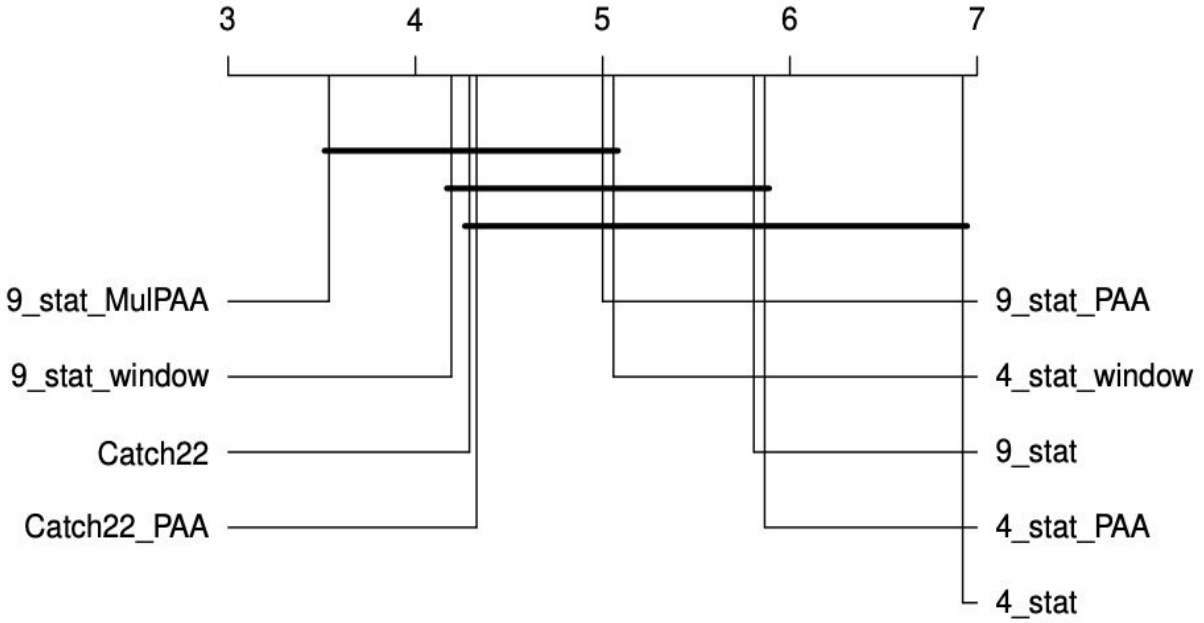
- To reduce the impact of the PAA parameter, use multiple PAA transformations
- Perform PAA at different levels, on each dimension of time-series (e.g., PAA_30, PAA_60, PAA_90)
- Extract features (4 stat and 9 stat) from each dimension of time series.
- Concatenate features and classify.



Results: Proposed Methods

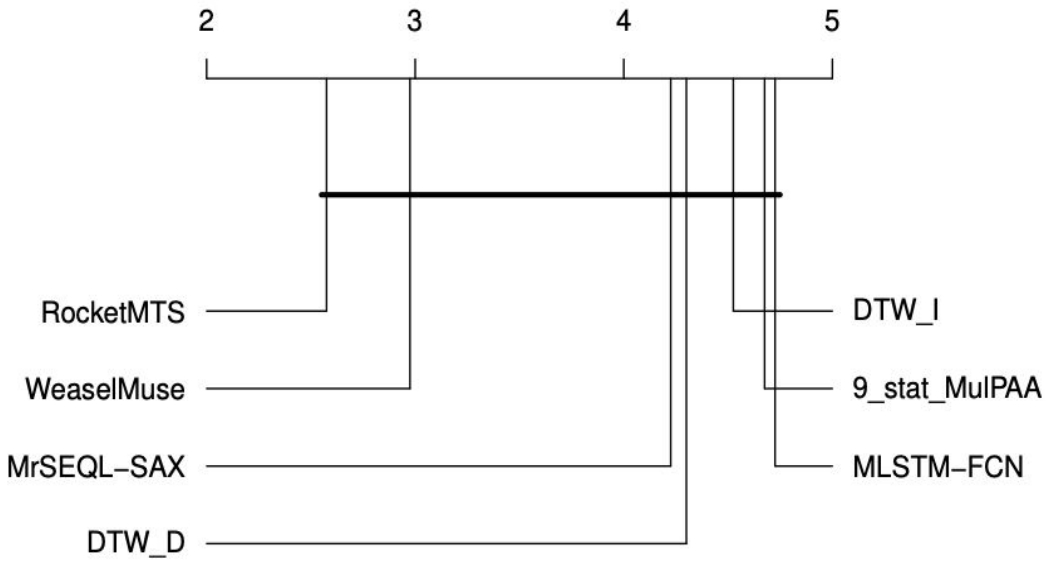
- Comparison of all proposed methods.
- Results on 26 equal length datasets from UEA MTSC benchmark.
- Visualise and summarise results with the Critical Difference diagram.





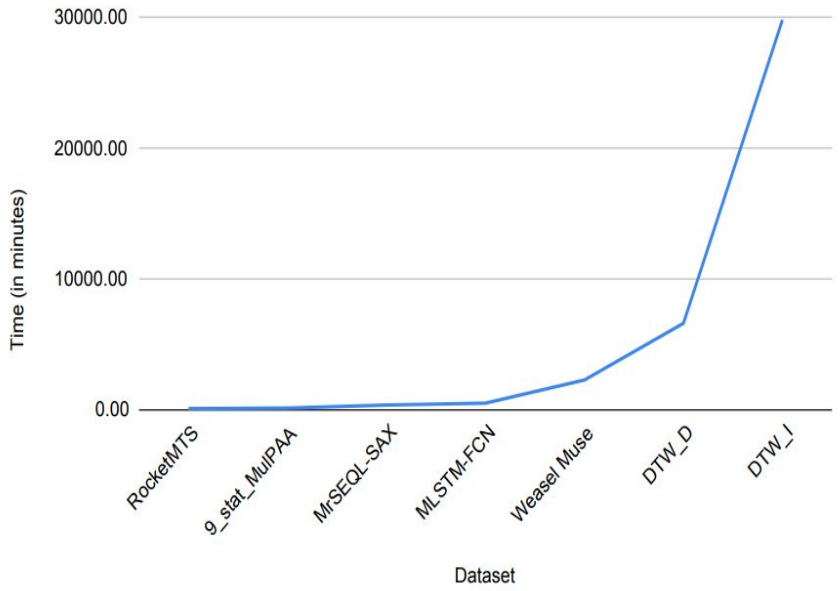
Results: State-of-the-art Accuracy

- Results are based on 20 datasets.
- RocketMTS outperforms all other methods, but no statistically significant difference to the other methods, on these datasets
- Deep Learning methods MLSTM-FCN and TapNet do not perform as well as expected



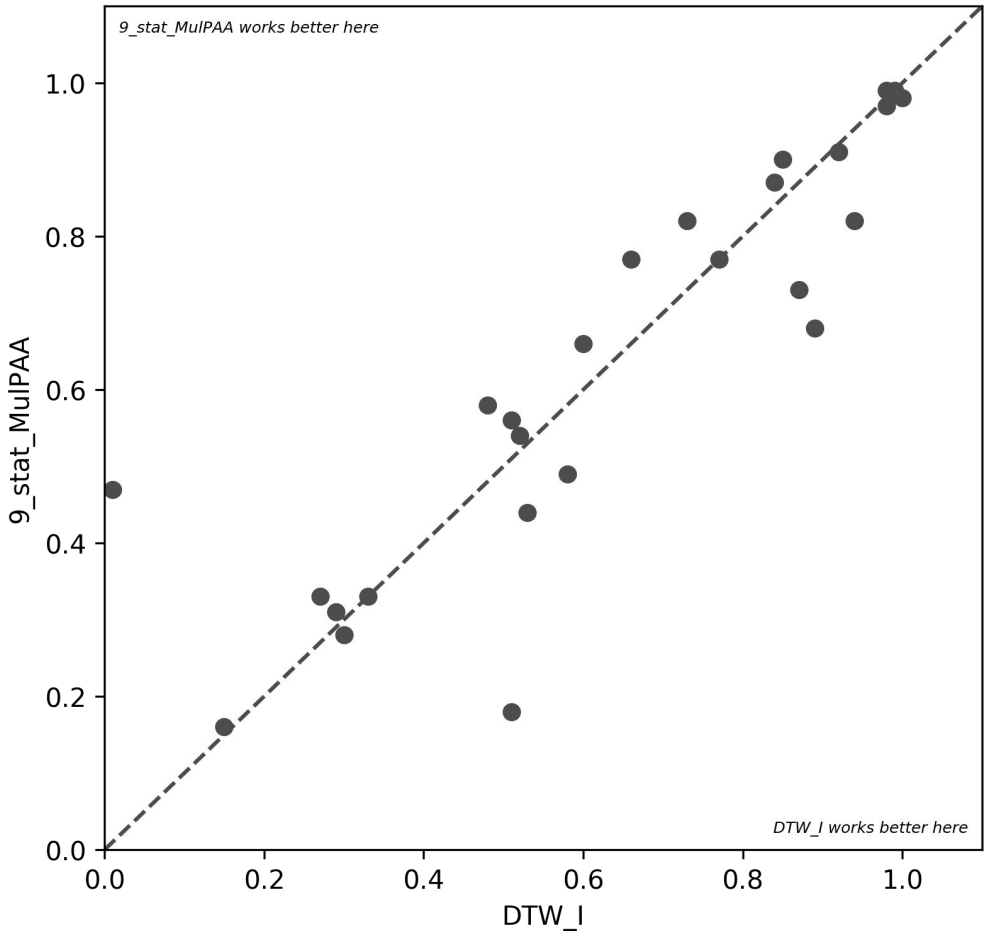
Results: State-of-the-art Runtime

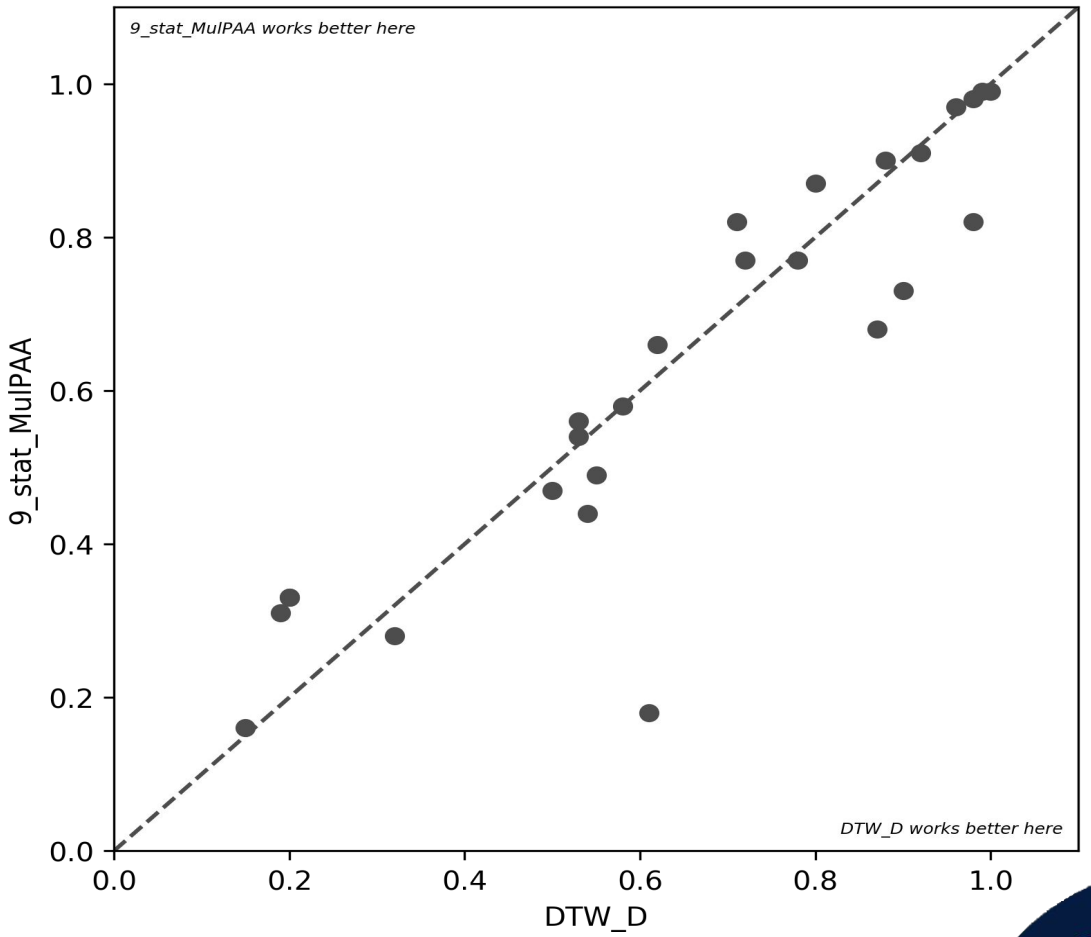
- RocketMTS takes the least amount of time to train (34 mins)
- DTW (I and D) are slowest, on some datasets even running for days.
- 9_stat_MulPAA takes 208 minutes, fastest method after ROCKET.
- System Config: 512GB memory with 72 cores and 32GB VRAM Nvidia GTX1080Ti. It operates on a Xeon E5-2695 processor with Ubuntu 18.04LTS.



Results: Accuracy Comparison

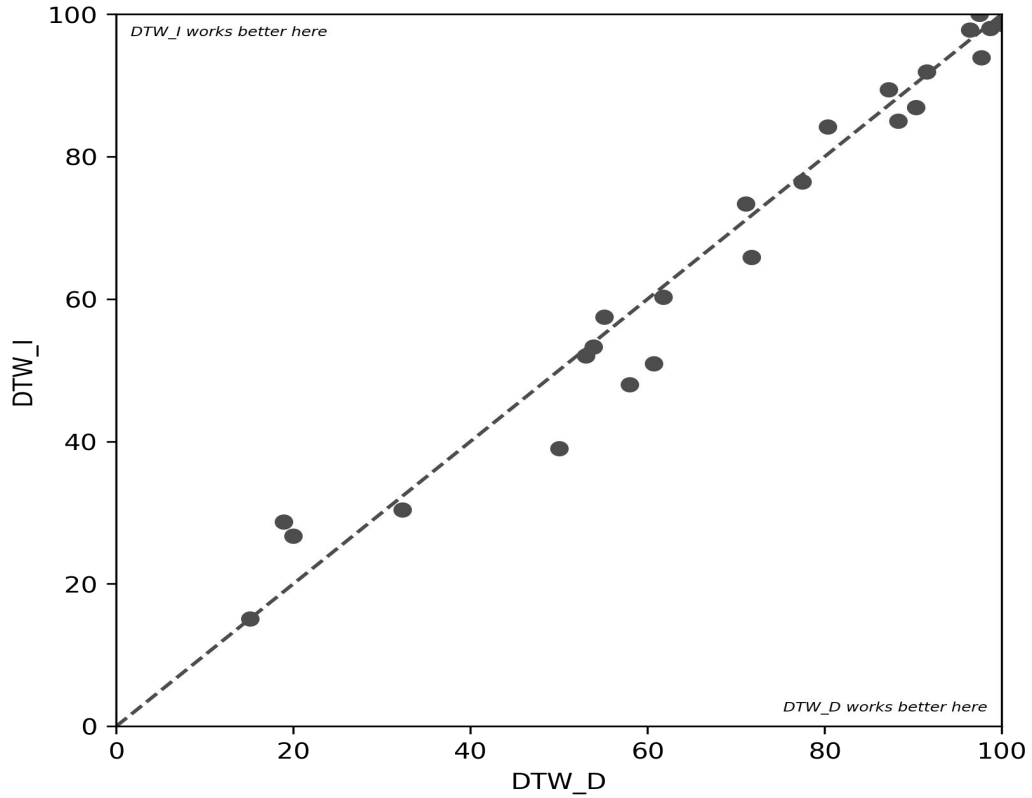
9_stat_MulPAA and DTW

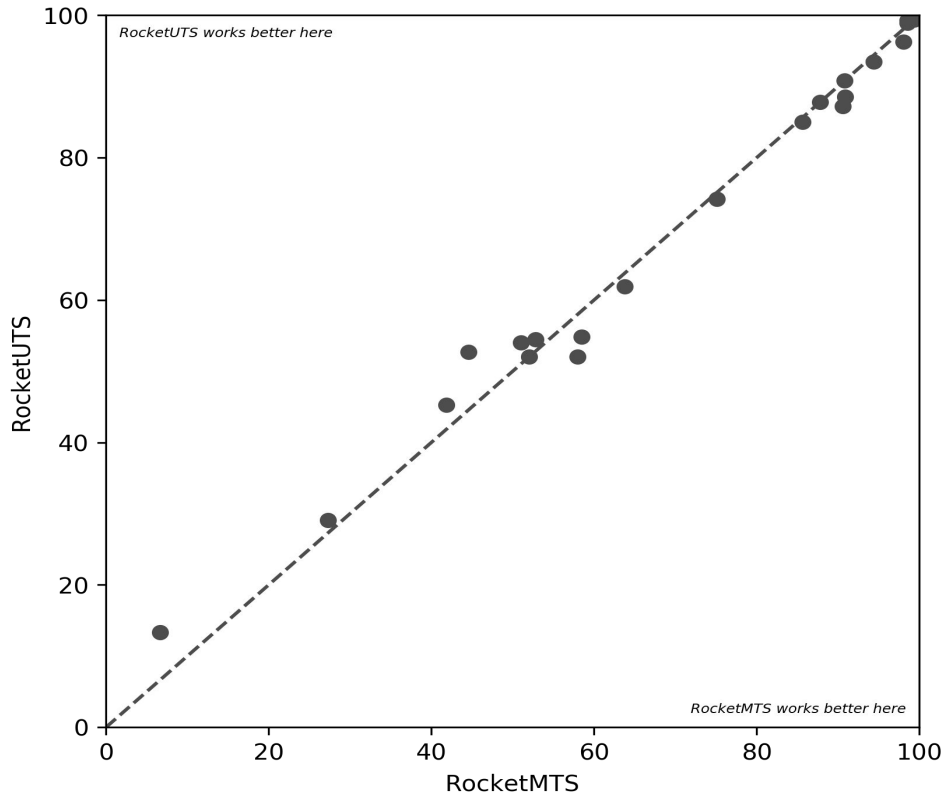




Results: Inter-dimension Dependency Interaction

DTW_I vs DTW_D and ROCKET_UTS vs ROCKET_MTS





Key Take-away: Extracting features independently from each dimension works as well as extracting features across dimensions

Conclusion

- We proposed an alternate MTSC baseline which is simple to implement and faster than DTW to train.
- We found that ROCKET, recently proposed classifier for UTSC, also works well for MTSC
- Weasel-Muse and MrSEQL also perform well but are more challenged by scale
- Variable-length TS handling an issue for many methods
- Deep learning methods do not achieve higher accuracy than the baselines.

- Scalability of methods for MTSC still a big challenge

		ROCKET-MTS	MrSEQL-SAX	DTW_I	DTW_D	WEASEL-MUSE	MLSTM FCN
Challenges	Variable-length	C		C	C		C
	Scalability		C	C	C	C	
	Memory Consumption					C	
	Overfitting						C

Thank you!
Questions?

Code in Github:
https://github.com/mlgig/mtsc_benchmark

Further Information

Bhaskar Dhariyal
University College Dublin
Email: bhaskar.dhariyal@insight-centre.org