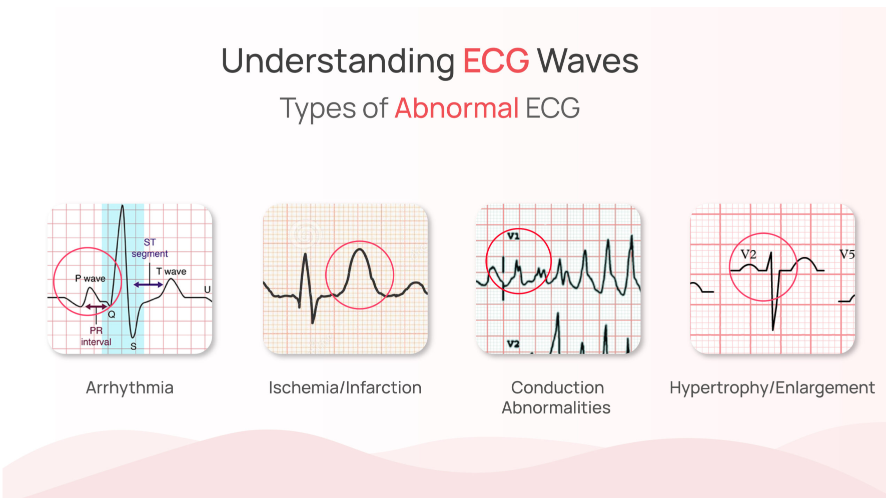
Deep Learning for Time Series

Session 4: Time Series Classification

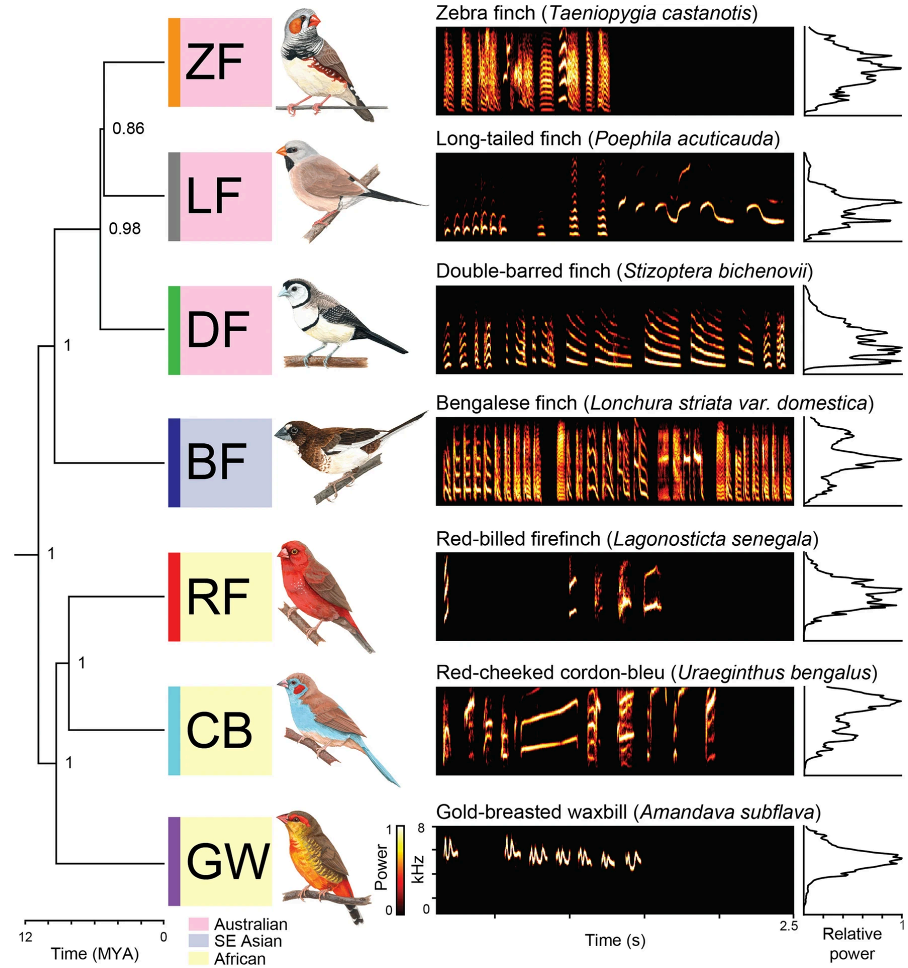
Romain Tavenard

Basics of Time Series Classification (TSC)

TSC examples



Source: sunfox.in blog



Source: "Machine learning and statistical classification of birdsong link vocal acoustic features with phylogeny"

Specific challenges of Multivariate Time Series Classification

- How to combine channels?
  - Early fusion
  - Late fusion
  - Channel-wise models
- Correlations between variables matter
  - Ignoring them hurts performance

Dataset-specific challenges

- Typical TSC datasets are **small**
  - Often a few hundreds of training samples
  - High risk of overfitting for deep models
- Strong heterogeneity across datasets
  - Length: short vs. long series
  - Noise levels
  - Intra-class variability

Benchmarking practices

- UCR/UEA benchmark is widely used
- Fixed train / test splits
  - No cross-validation in standard benchmarks
  - Encourages benchmark-specific tuning
- Evaluation often relies on:
  - Accuracy
  - Average rank across (very diverse) datasets

$\Rightarrow$ Risk of overfitting to benchmarks rather than solving real-world problems

Overview of the State-of-the-art

Bird's eye view

1. Before 2016:
   - Feature-based methods
   - Distance-based methods
   - Ensembles (COTE series)
2. 2016-2020
   - DNNs used as is (Resnets, MLPs, CNNs, FCNs, InceptionTime)
   - Random convolutions: Rocket (2020), MiniRocket (2021), MultiRocket (2022)
   - Transformers
   - HIVE-COTEv1 and v2: ensembling of more methods + hierarchical vote (not covered)
3. 2022- : Foundation models (covered in the next session)

Remarks

1. Contrary to forecasting, DNNs are yet to beat other baselines
2. Transformer-based methods are mainly derivations of forecasting models
3. Ensemble methods are very competitive in benchmarks

Historical baselines

Distance-based baselines

- Key similarity measures (task-specific):
  - Euclidean distance
  - (variants of) Dynamic Time Warping
  - Longest Common Subsequence
  - *etc.*

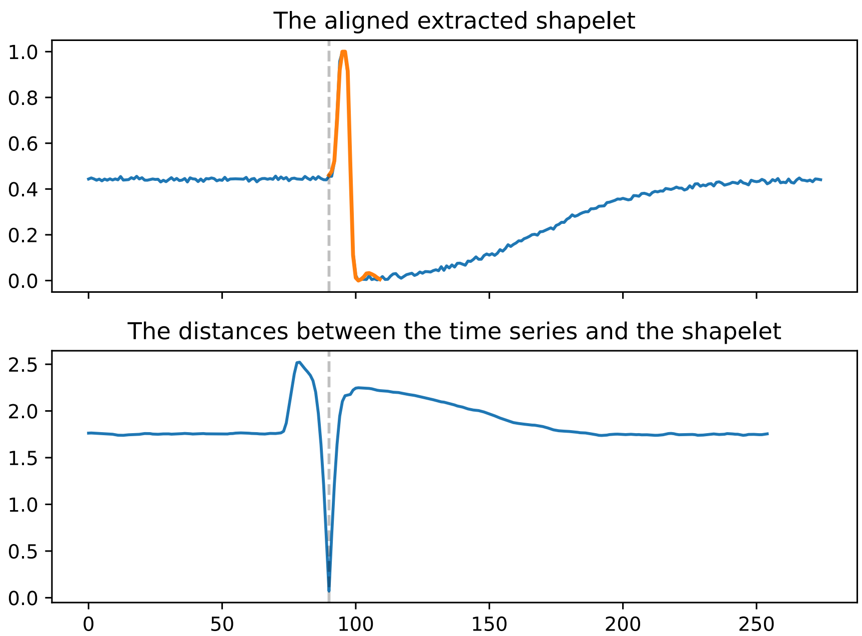
Feature-based baselines

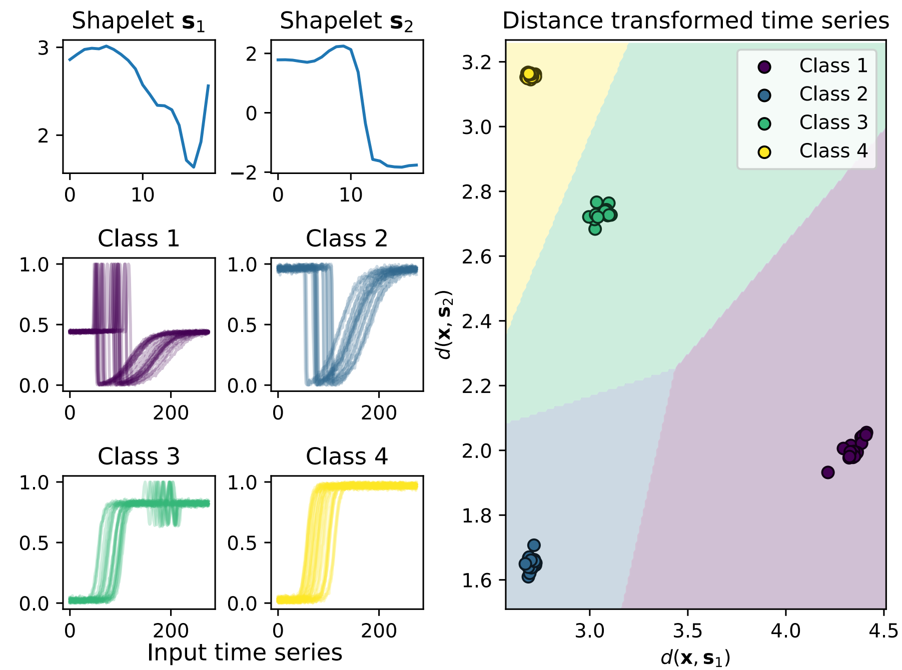
- Extract numerous features and plug a simple classifier
  - Often a strong baseline in benchmark evaluations!

- Example features:
  - `sum_of_reoccurring_values` - sum of all values present in the time series more than once
  - `longest_strike_above_mean` - length of the longest consecutive subsequence that is bigger than the mean

Shapelets

- Shapelet = a subsequence of consecutive observations from a time series
  - Can be chosen or learned
  - Goal: Choose/learn a pool of K shapelets that are discriminative for a given task





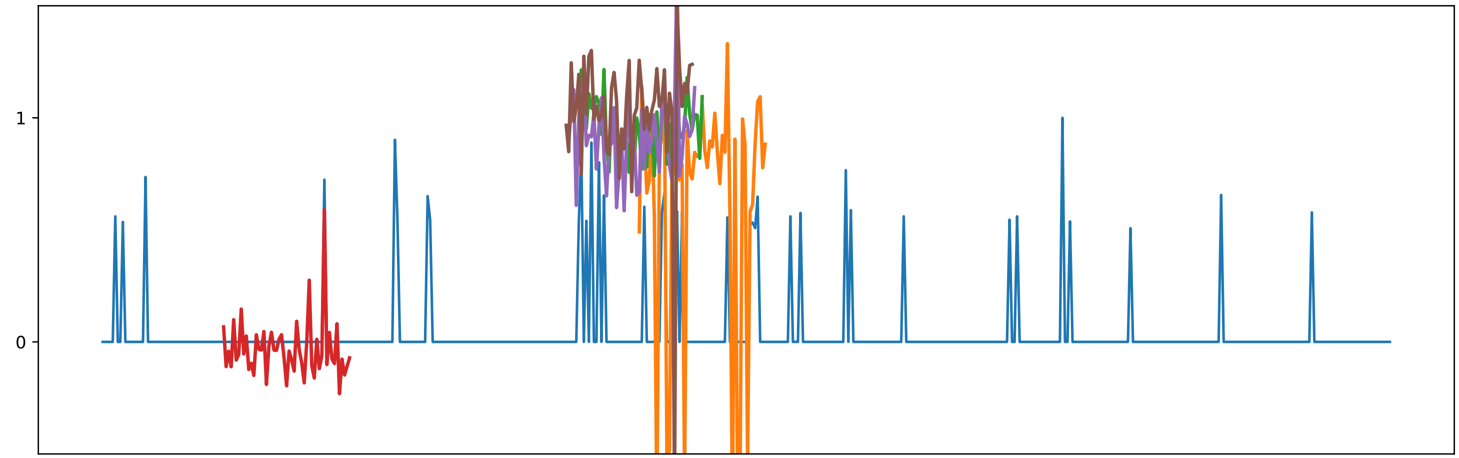
Illustrations from tslearn docs

From classic to learned shapelets

- Original shapelets:
  - Exhaustive or heuristic search
  - Expensive but interpretable
- Learned shapelets:
  - Learning Shapelets (LS)
  - Shapelet layers in neural networks
  - Joint optimization with classifier
- Advantages:
  - Interpretability
  - Competitive performance on small datasets

A word on Shapelet interpretability

- Shapelets are often selected for their interpretability
- What about learned shapelets?



Source: “Localized Random Shapelets”, AALTD’19

Ensembles

- Collective of Transformation-based Ensembles (COTE)
  - if there is no prior knowledge, ensemble different representations

1. Flat-COTE (2016): 35 classifiers over four data representations
   - shapelets, DTWs, *etc*
2. Hive-COTE-alpha,v1,v2 (2018, 2020, 2022):
   - more representations (forests, spectral) + hierarchical voting procedure

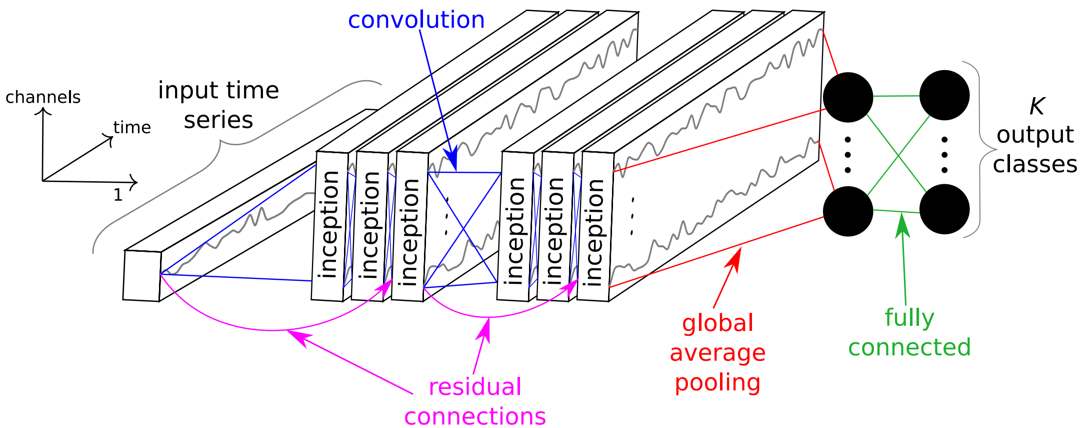
HC2 is one of the best methods on open public benchmarks but very slow

From traditional models to deep learning

Take 1: just try classic vision models on TS

- Many standard Conv-based architectures can be adapted for TSC (eg. InceptionTime: a Resnet with inception module)
  - a stack of convolutions of different sizes

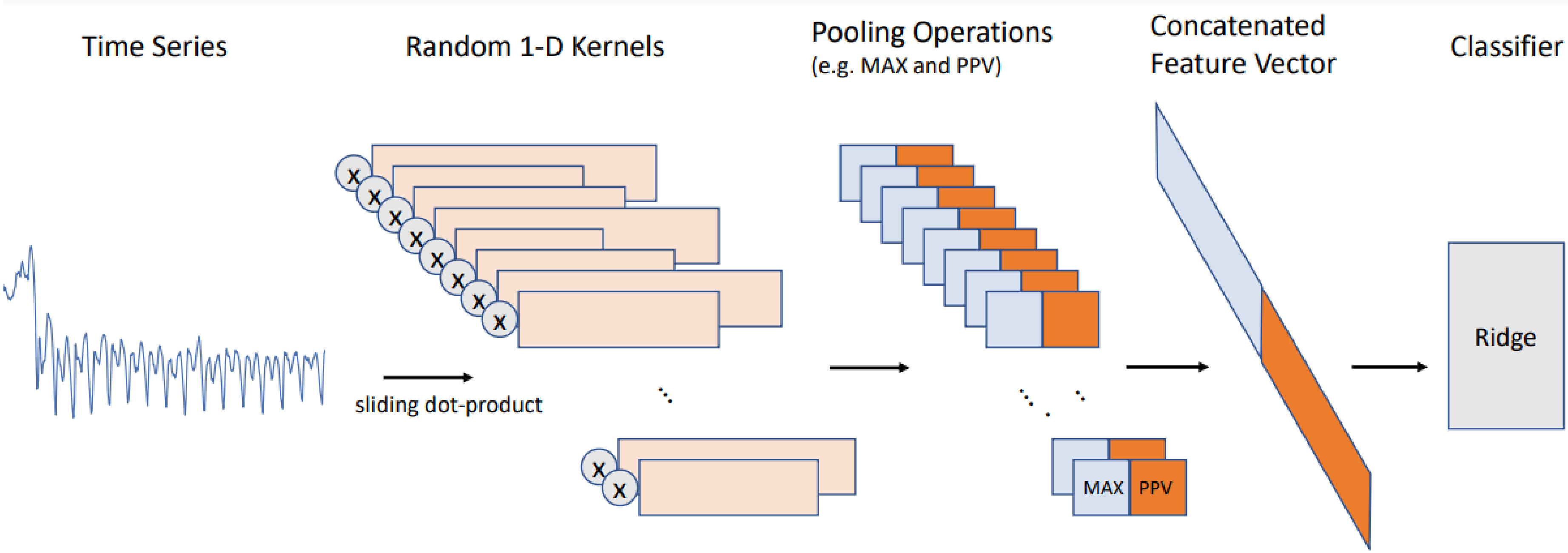
$\rightarrow$ multi-resolution analysis



Source: “InceptionTime: Finding AlexNet for Time Series Classification”, DMKD’20

Take 2: simple models can be strong baselines

- ROCKET: use random 1D convolutions as feature extractors
  - Use maxpooling and PPV (proportion of positive values) as aggregators
  - Apply ridge regressor to the obtained embedding



Source: aeon docs

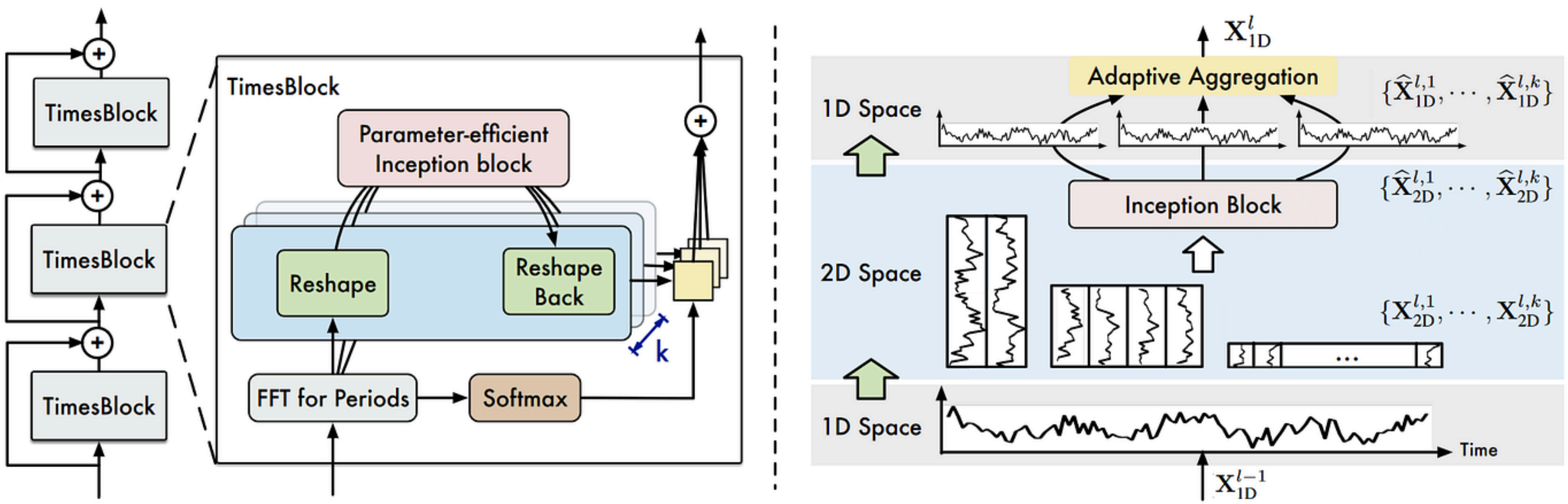
Take 2: simple models can be strong baselines

- ROCKET extensions:
  - hard-coded convolutions (MiniRocket)
  - more aggregators (MultiRocket)
  - random convolutions + dictionary learning (MR-HYDRA)

$\rightarrow$ On par with HC2 but much faster

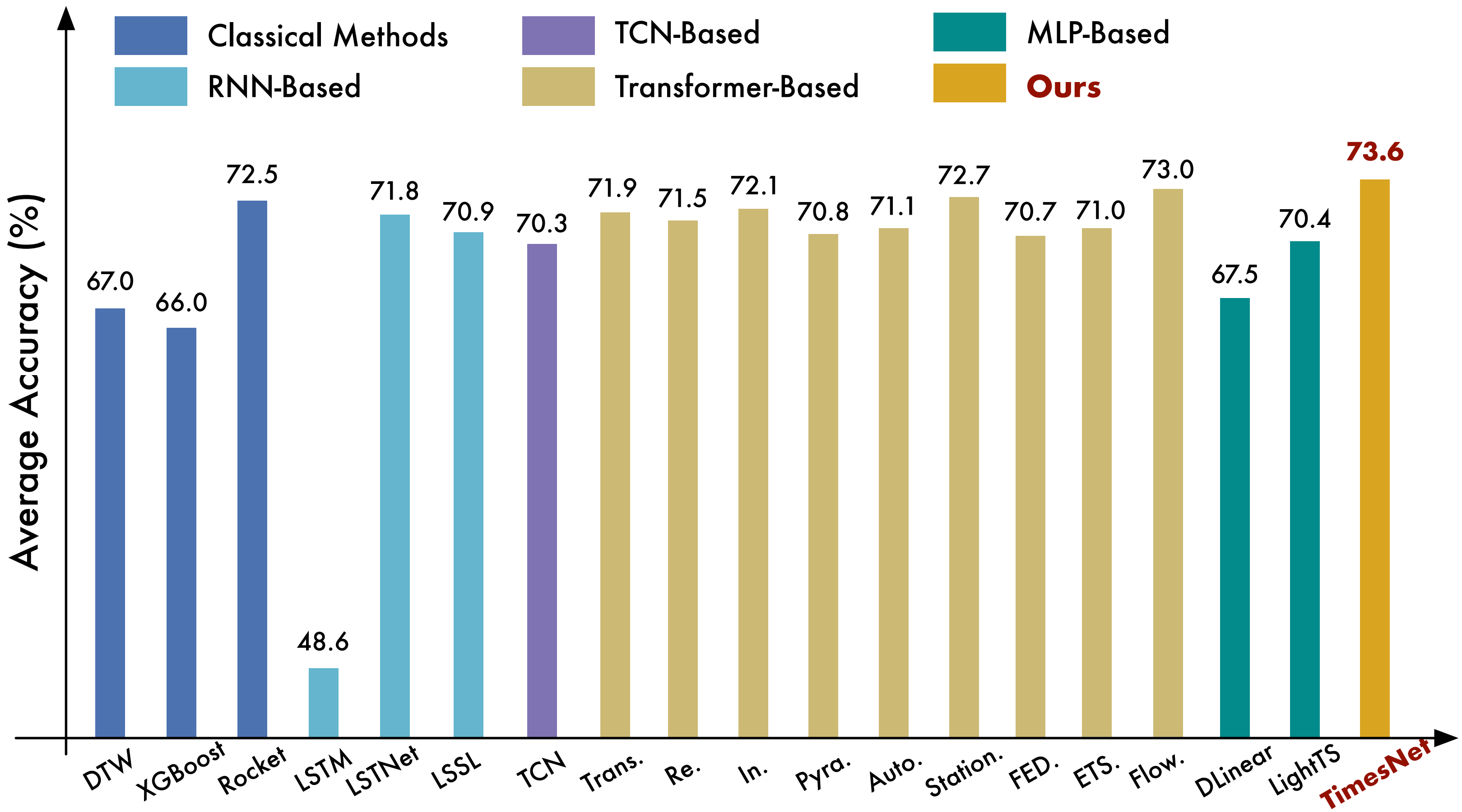
TimesNet: a CNN with inception module and 2D kernels

- Assumption: the signal is periodic
- Motivation: we want to capture intra-period AND inter-period variations



Source: “TimesNet: […]”, ICLR’23

TimesNet: a CNN with inception module and 2D kernels



Source: “TimesNet: […]”, ICLR’23

Accuracy vs. Efficiency Trade-offs: Computational considerations

- COTE / HC2:
  - Very strong accuracy (favored by the diversity of the benchmarks)
  - Extremely expensive computationally
- ROCKET-based methods:
  - Fast training and inference
  - Excellent accuracy-efficiency trade-off
- Deep models:
  - GPU-friendly, data-hungry