EIVIA 2025: Deep Learning for Time Series and Applications to Healthcare

Gonzalo Uribarri
KTH Royal Institute of Technology & SciLifeLab

@gonzauri

State-of-the-art (SOTA)

Models for TSC

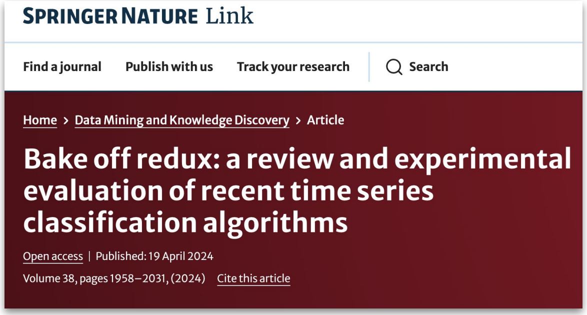
Models for time series classification are tested on the **UEA** archive and the **UCR** archive (benchmarks for multivariate and univariate TSC).

Models for TSC

Models for time series classification are tested on the **UEA** archive and the **UCR** archive (benchmarks for multivariate and univariate TSC).

UEA

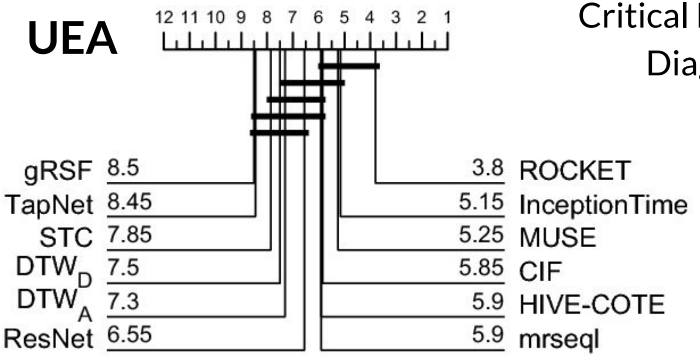
UCR



Models for TSC

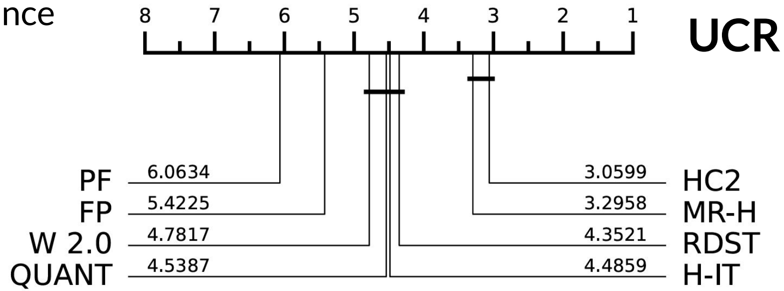
Models for time series classification are tested on the **UEA** archive and the **UCR** archive (benchmarks for multivariate and univariate TSC).

UEA



Alejandro Pasos Ruiz et al. [2021]

Critical Difference Diagrams



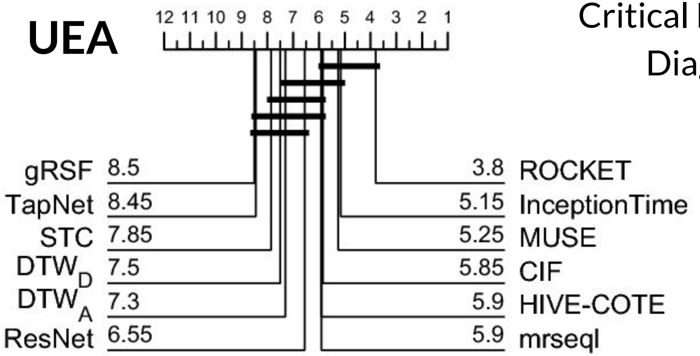
UCR

Middlehurst et al. [2024]

Models for TSC

Models for time series classification are tested on the **UEA** archive and the **UCR** archive (benchmarks for multivariate and univariate TSC).

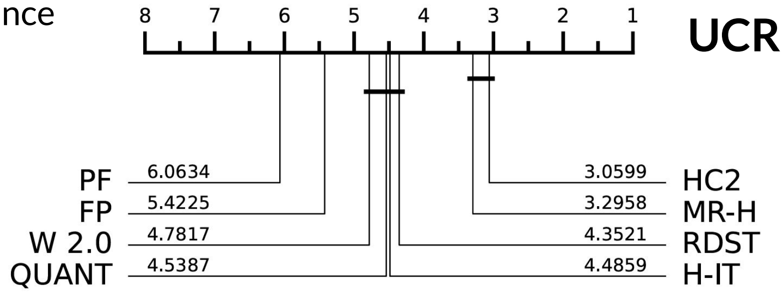
UEA



Alejandro Pasos Ruiz et al. [2021]

Critical Difference Diagrams

UCR

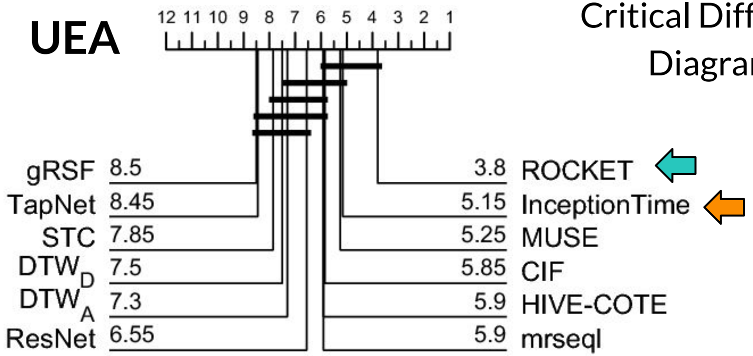


Middlehurst et al. [2024]

The position of each model represents its mean rank (lower is better)

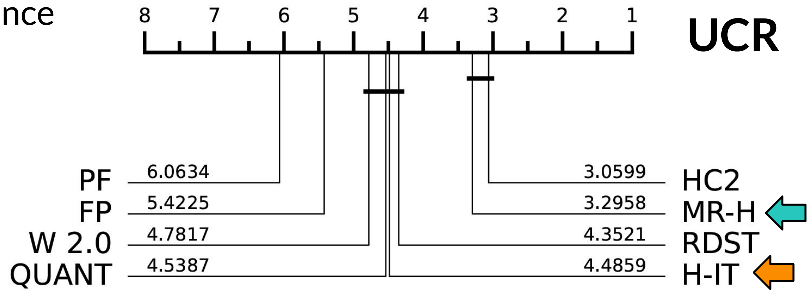
Models for TSC

Models for time series classification are tested on the **UEA** archive and the **UCR** archive (benchmarks for multivariate and univariate TSC).



Alejandro Pasos Ruiz et al. [2021]

Critical Difference Diagrams



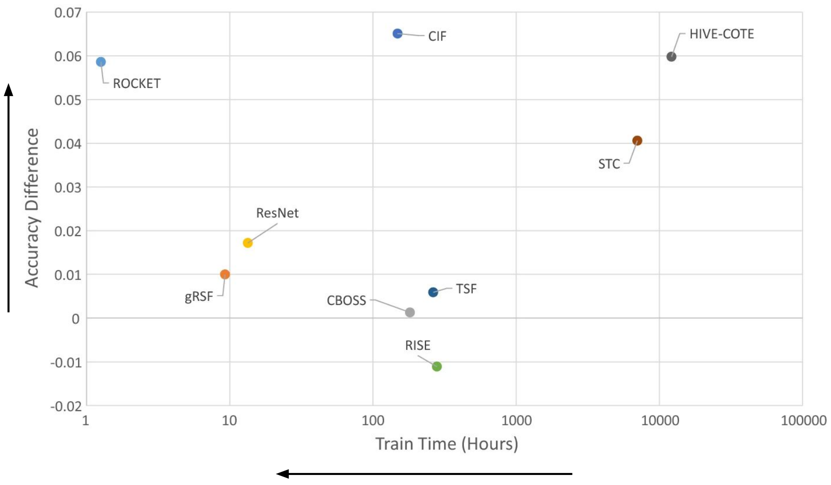
Middlehurst et al. [2024]

InceptionTime and ROCKET are two different models that performs very well.

Models for TSC

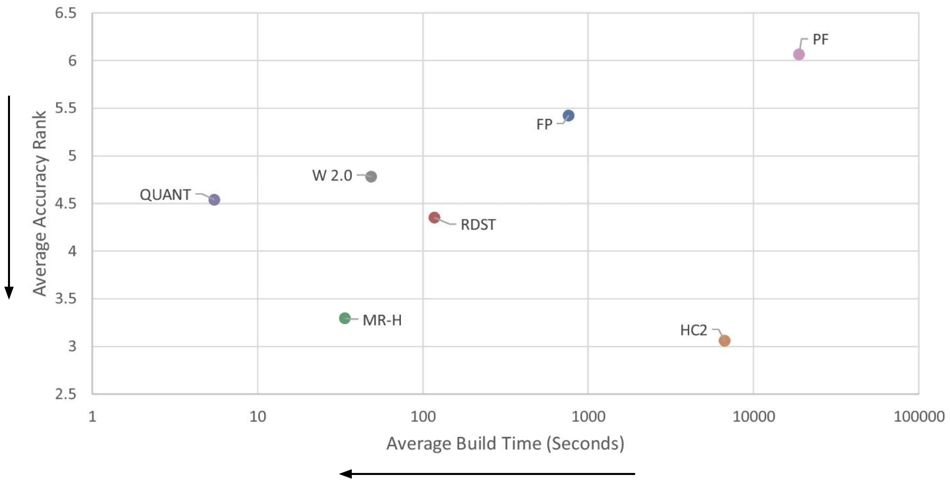
But accuracy is not all, efficiency also matters.

UEA



Alejandro Pasos Ruiz et al. [2021]

UCR

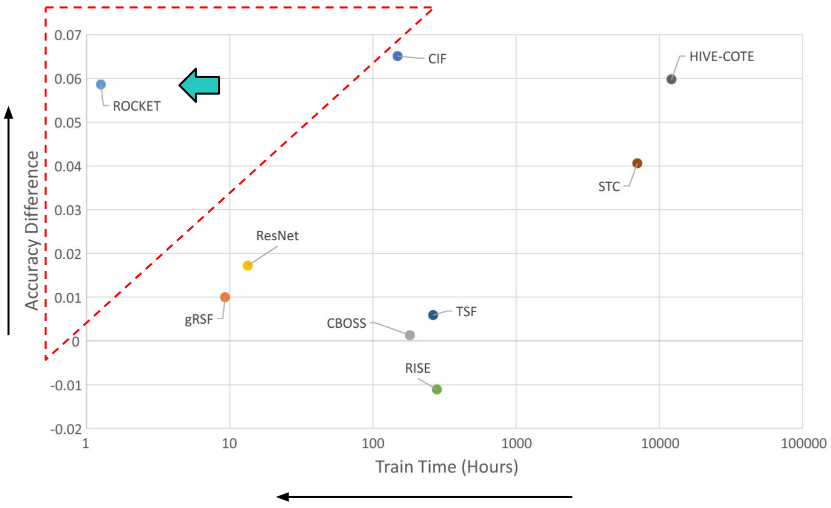


Middlehurst et al. [2024]

Models for TSC

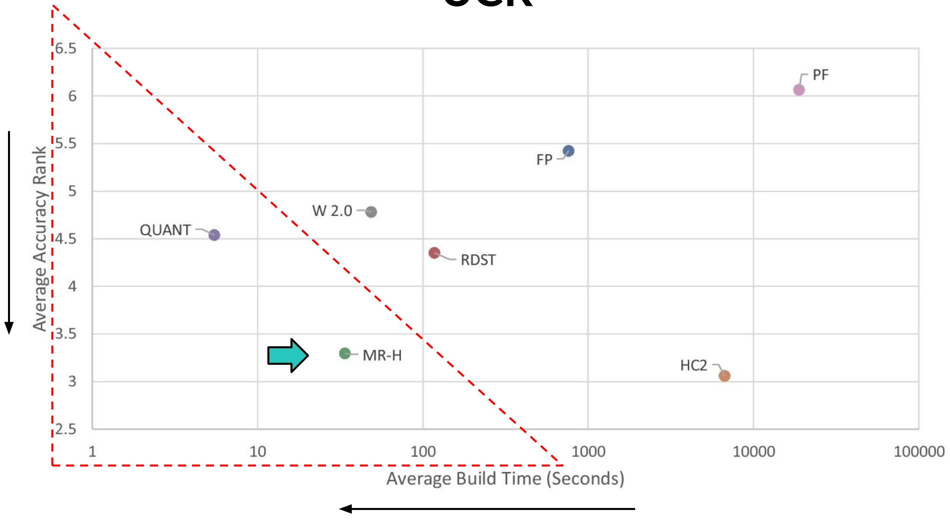
But accuracy is not all, efficiency also matters.

UEA



Alejandro Pasos Ruiz et al. [2021]

UCR



Middlehurst et al. [2024]

InceptionTime

Time Series Classification

Feature Based

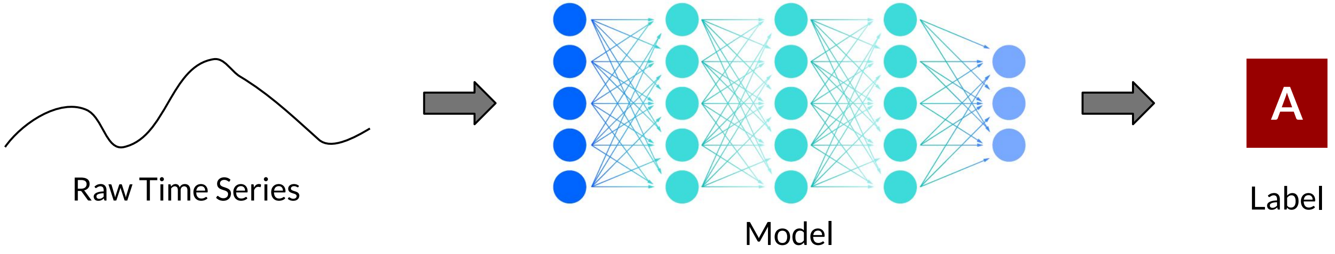
- Predefined Features
- Spectral Features
- Intervals / Dictionary

Distance Based

- Elastic distance
- Representation Learning

End-to-end

- CNNs
- RNNs
- Transformers



InceptionTime

SPRINGER NATURE Link

Find a journal Publish with us Track your research Search

Home > Data Mining and Knowledge Discovery > Article

InceptionTime: Finding AlexNet for time series classification

Published: 07 September 2020

Volume 34, pages 1936–1962, (2020) Cite this article

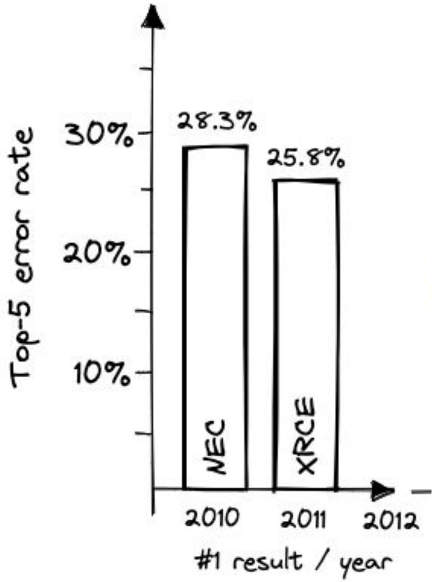
AlexNet - A little bit of history

ImageNet is an **HUGE** image dataset released in 2009, containing 12 million images in 22,000 categories.

(a) ImageNet Synset: One sample image from each category

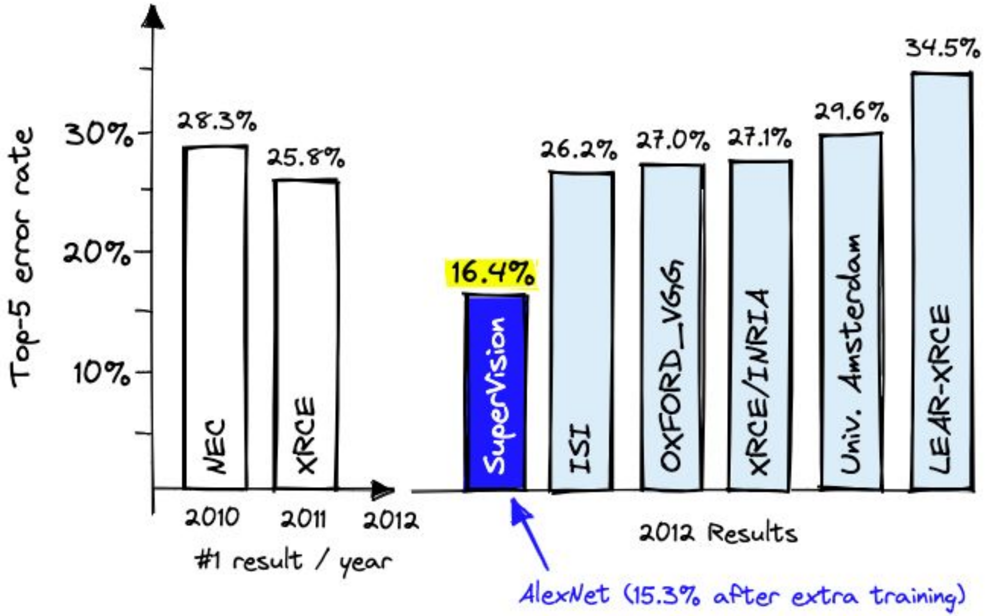
AlexNet - A little bit of history

The **ImageNet Challenge** was a competition to use 1.2 million images to classify 1000 different classes of images.



AlexNet - A little bit of history

The **ImageNet Challenge** was a competition to use 1.2 million images to classify 1000 different classes of images.

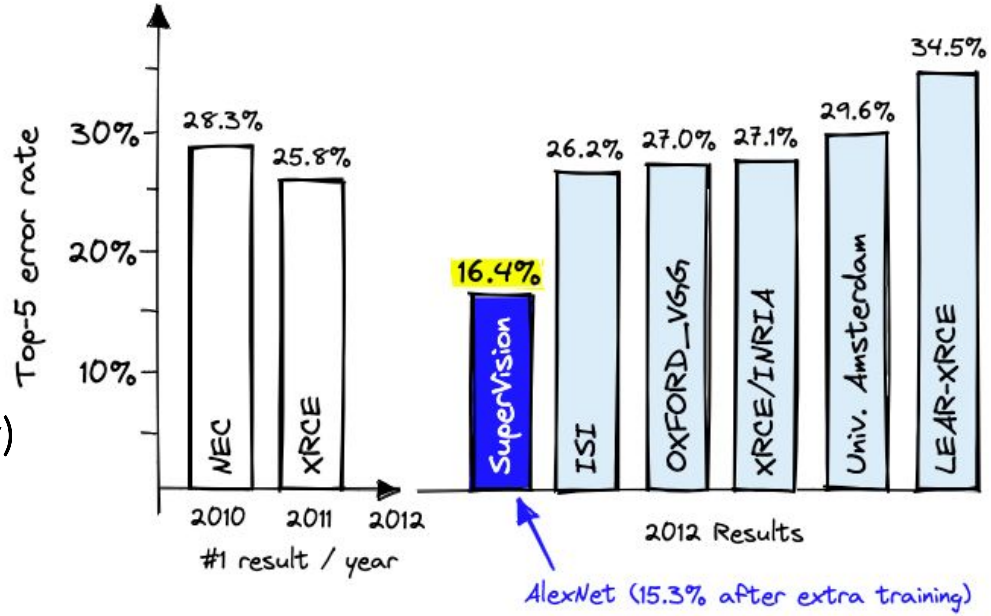


AlexNet - A little bit of history

The **ImageNet Challenge** was a competition to use 1.2 million images to classify 1000 different classes of images.

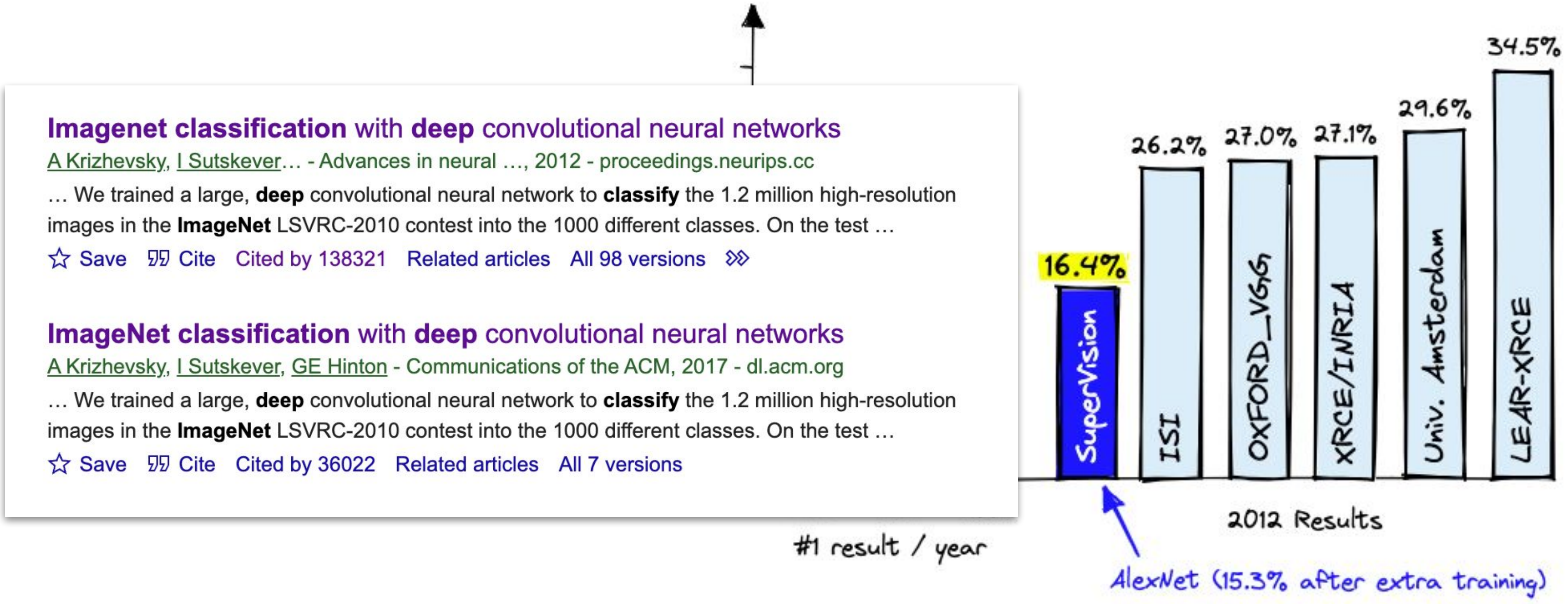
Key elements:

- Deep (8 layers)
- ReLU
- Dropout
- Augmentation (on-the-fly)
- **GPU !!!**



AlexNet - A little bit of history

The **ImageNet Challenge** was a competition to use 1.2 million images to classify 1000 different classes of images.



Imagenet classification with **deep** convolutional neural networks
A Krizhevsky, I Sutskever… - Advances in neural …, 2012 - proceedings.neurips.cc
… We trained a large, **deep** convolutional neural network to **classify** the 1.2 million high-resolution images in the **ImageNet** LSVRC-2010 contest into the 1000 different classes. On the test …
☆ Save 99 Cite Cited by 138321 Related articles All 98 versions »

ImageNet classification with **deep** convolutional neural networks
A Krizhevsky, I Sutskever, GE Hinton - Communications of the ACM, 2017 - dl.acm.org
… We trained a large, **deep** convolutional neural network to **classify** the 1.2 million high-resolution images in the **ImageNet** LSVRC-2010 contest into the 1000 different classes. On the test …
☆ Save 99 Cite Cited by 36022 Related articles All 7 versions

InceptionTime

SPRINGER NATURE Link

Find a journal　Publish with us　Track your research　Search

Home › Data Mining and Knowledge Discovery › Article

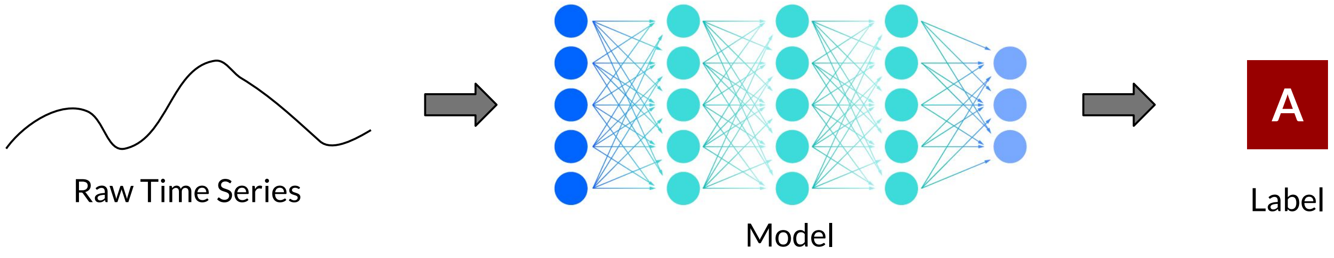
InceptionTime: Finding AlexNet for time series classification

Published: 07 September 2020

Volume 34, pages 1936–1962, (2020)　Cite this article

InceptionTime

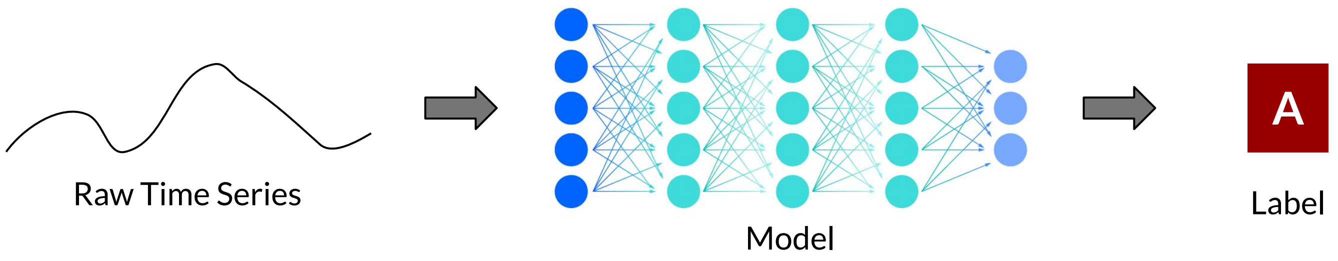
It is a **Deep 1-D Convolutional Neural Network** model inspired by the **Inception-v4** architecture.



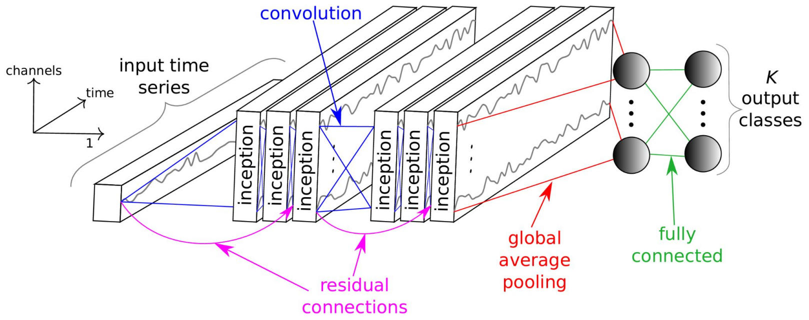
Ismail Fawaz et al. [2020]

InceptionTime

It is a **Deep 1-D Convolutional Neural Network** model inspired by the **Inception-v4** architecture.



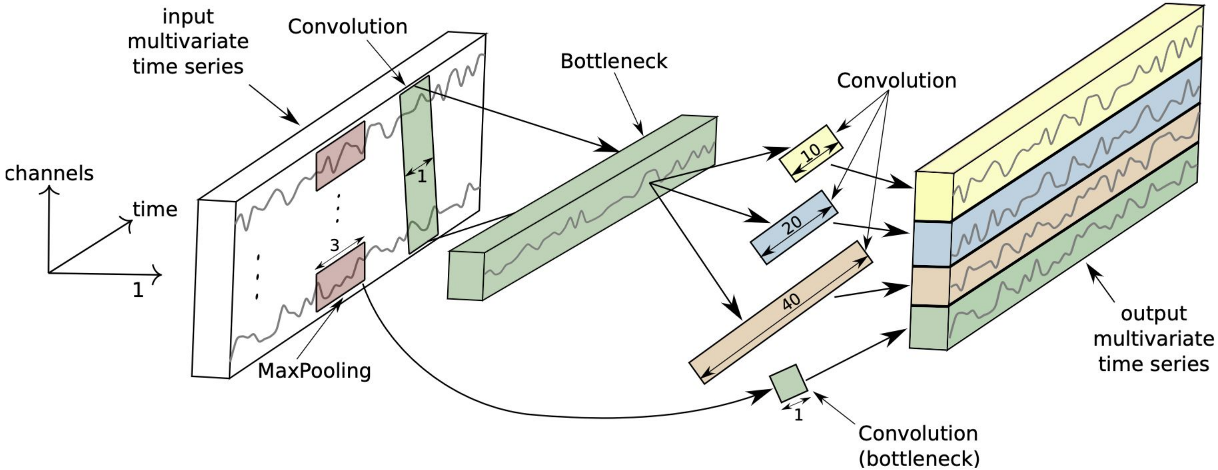
It is composed by **InceptionTime Modules**



Ismail Fawaz et al. [2020]

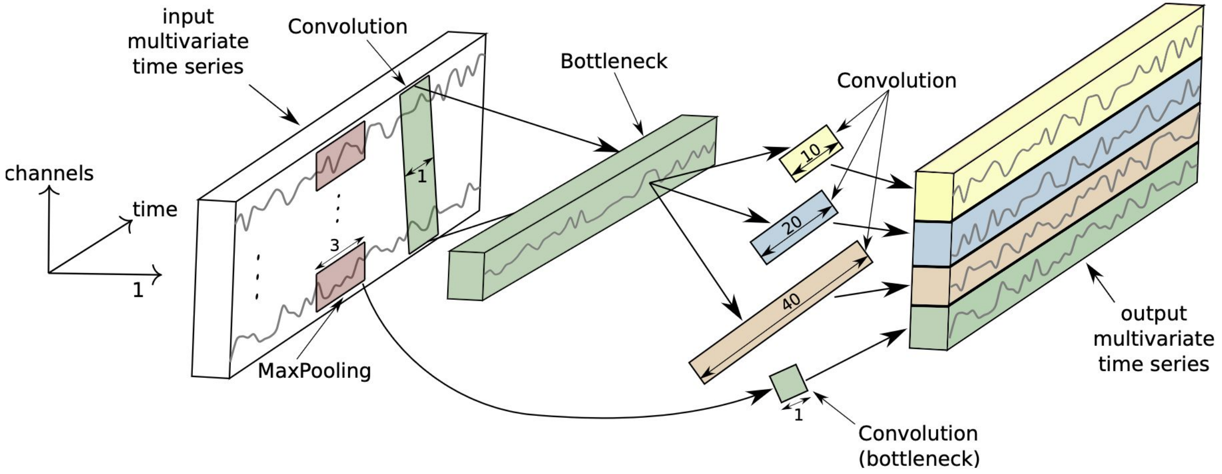
InceptionTime

InceptionTime
Module



InceptionTime

InceptionTime Module

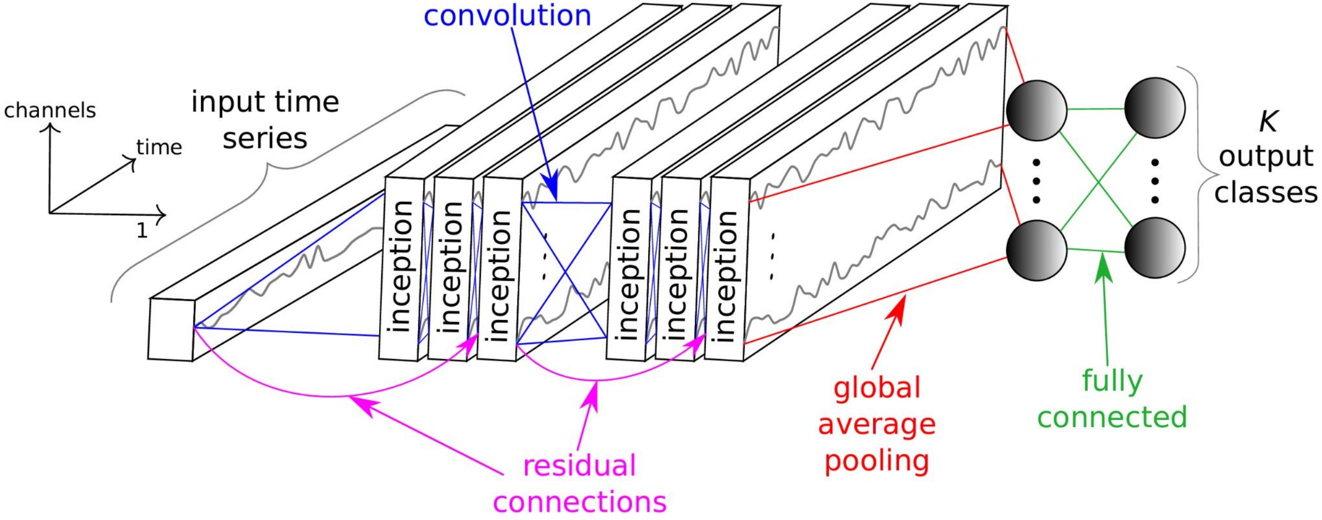


Key elements:

- **Multi-Scale Feature Extraction:** Filter of different sizes.
- **Large Kernels:** Larger than ones used for images
- **Bottlenecks:** 1x1 Convolutions (mixing channels)

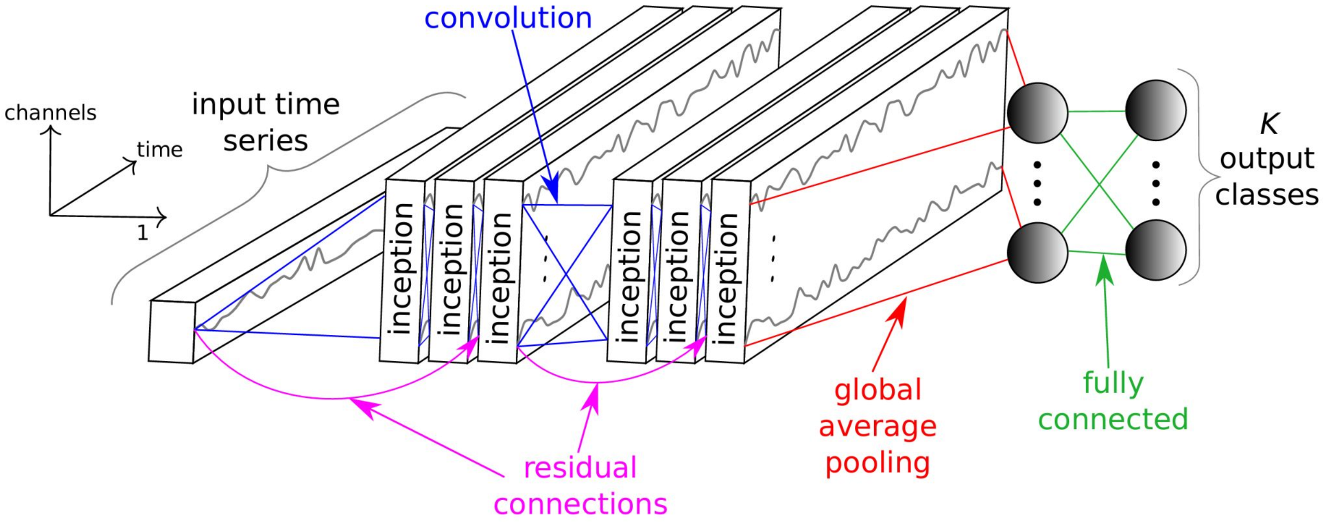
InceptionTime

InceptionTime Architecture



InceptionTime

InceptionTime Architecture



Key elements:

- **Residual Connections:** Improve training stability.
- **Global Average Pooling:** Time pooling operation before the fully connected layer.

InceptionTime

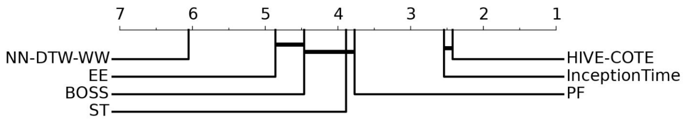


Fig. 5: Critical difference diagram showing the performance of InceptionTime compared to the current state-of-the-art classifiers of time series data.

Performance on UCR

InceptionTime

InceptionTime: Finding AlexNet for Time Series Classification 9

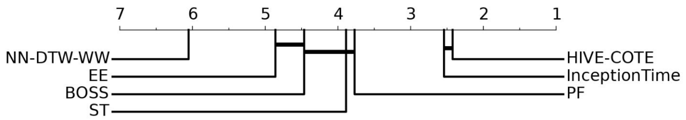


Fig. 5: Critical difference diagram showing the performance of InceptionTime compared to the current state-of-the-art classifiers of time series data.

Performance on UCR

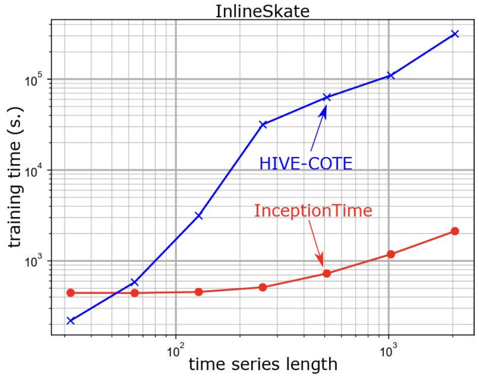


Fig. 7: Training time as a function of the series length for the InlineSkate dataset.

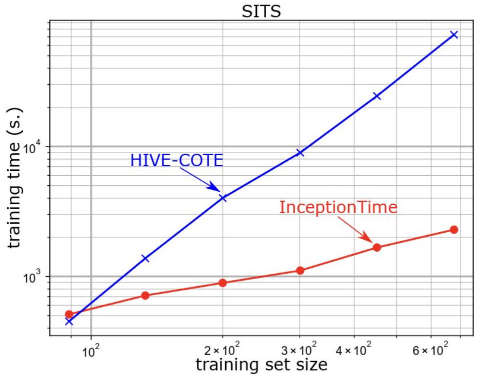


Fig. 8: Training time as a function of the training set size for the SITS dataset.

InceptionTime Strengths and Weaknesses

- State-of-the-art performance for TSC
- Fast inference
- Good Scaling
- **Truly Multivariate**

InceptionTime Strengths and Weaknesses

- State-of-the-art performance for TSC
- Fast inference
- Good Scaling
- **Truly Multivariate**

- Complex Training
- High Variance (Counter using an ensemble)
- Difficult to interpret

ROCKET

Time Series Classification

Feature Based

- Predefined Features
- Spectral Features
- Intervals / Dictionary

Distance Based

- Elastic distance
- Representation Learning

End-to-end

- CNNs
- RNNs
- Transformers

Time Series Classification

Feature Based

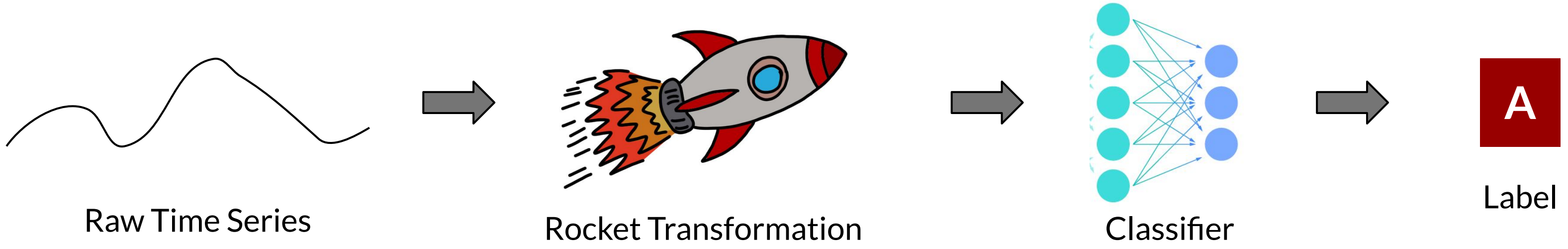
- Predefined Features
- Spectral Features
- Intervals / Dictionary

Distance Based

- Elastic distance
- Representation Learning

End-to-end

- CNNs
- RNNs
- Transformers



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

Published: 13 July 2020

Volume 34, pages 1454–1495, (2020) Cite this article

Angus Dempster, François Petitjean & Geoffrey I. Webb

SPRINGER NATURE Link

Find a journal   Publish with us   Track your research   Search

Home > Data Mining and Knowledge Discovery > Article

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

Published: 13 July 2020

Volume 34, pages 1454–1495, (2020)   Cite this article

Download PDF

Access provided by KTH Royal Institute of Technology

Angus Dempster, François Petitjean & Geoffrey I. Webb

RESEARCH-ARTICLE

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

Authors: Angus Dempster, Daniel F. Schmidt, Geoffrey I. Webb   Authors Info & Claims

KDD '21: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining • Pages 248 - 257
https://doi.org/10.1145/3447548.3467231

Published: 14 August 2021 Publication History

SPRINGER NATURE Link

Find a journal   Publish with us   Track your research   Search

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

Volume 36, pages 1623–1646, (2022)   Cite this article

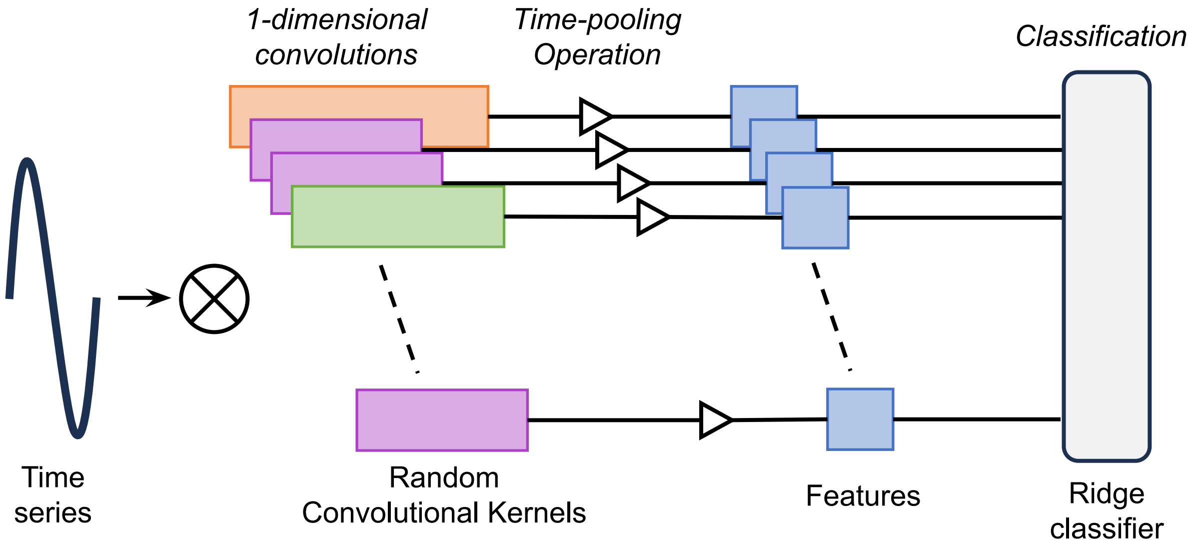
ROCKET models

Random Convolutional Kernel Transform (ROCKET)* is a **transformation stage** which can be applied to **time-series data**.

* Dempster et al. [2020]

ROCKET models

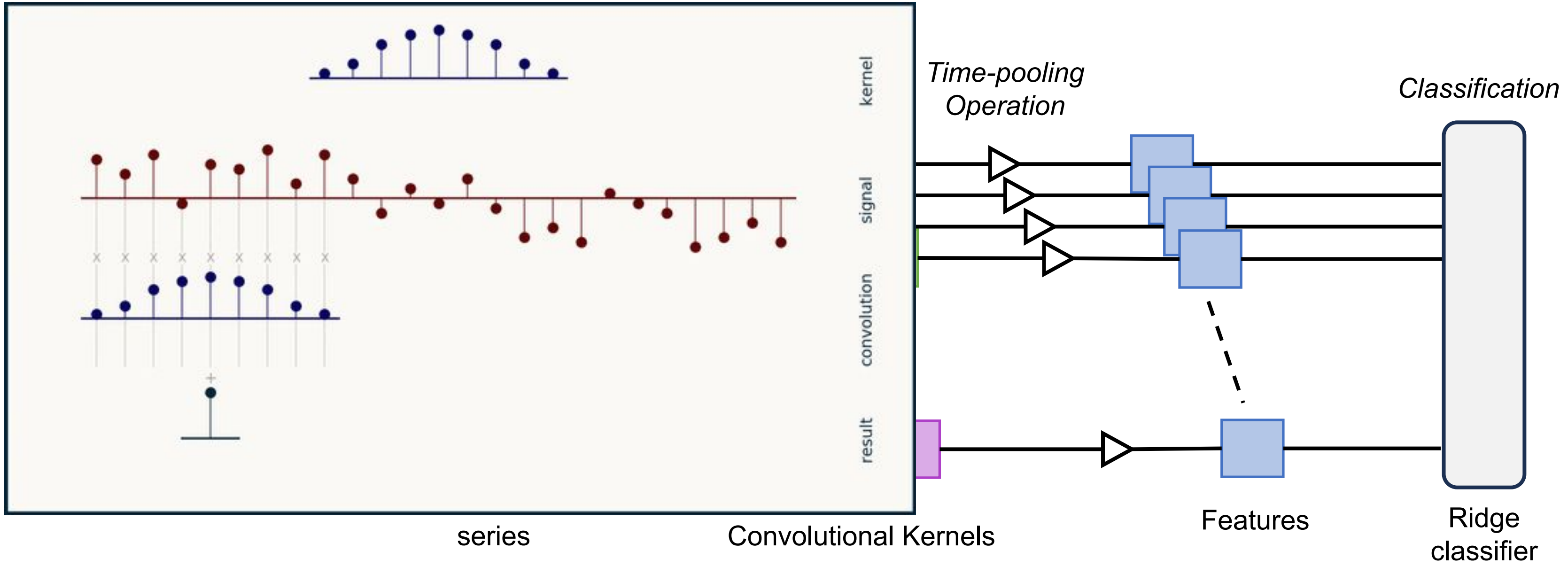
Random Convolutional Kernel Transform (ROCKET)* is a **transformation stage** which can be applied to **time-series data**.



* Dempster et al. [2020]

ROCKET models

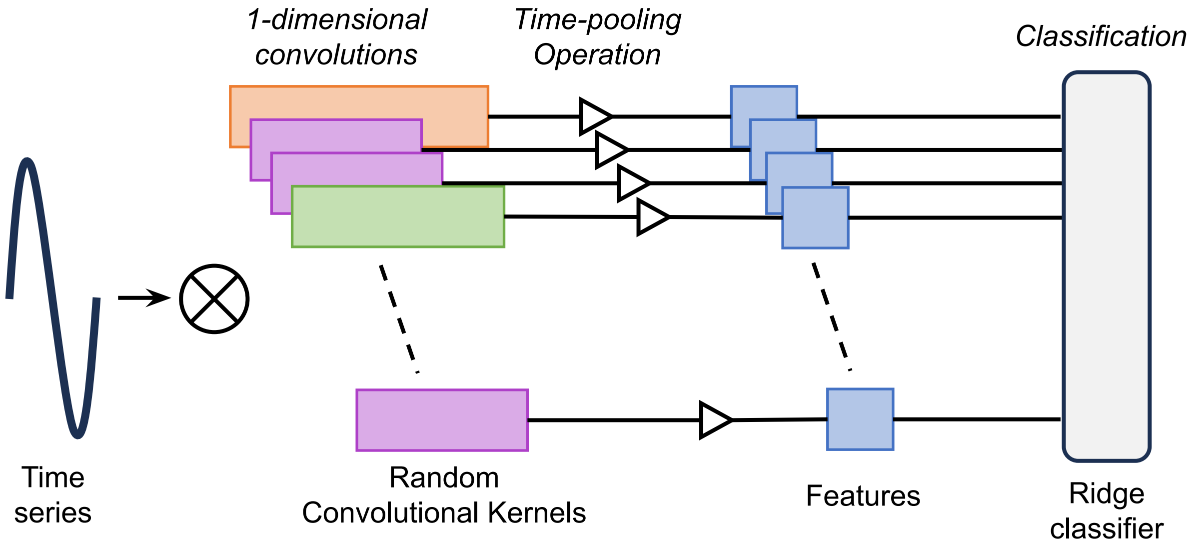
Random Convolutional Kernel Transform (ROCKET)* is a **transformation stage** which can be applied to **time-series data.**



* Dempster et al. [2020]

ROCKET models

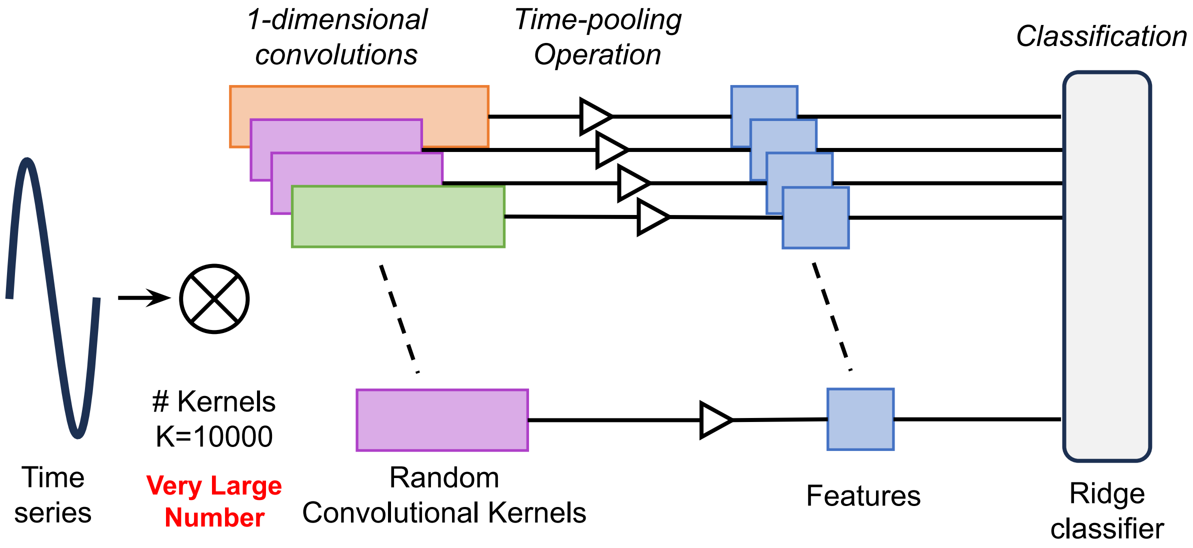
Random Convolutional Kernel Transform (ROCKET)* is a **transformation stage** which can be applied to **time-series data.**



* Dempster et al. [2020]

ROCKET models

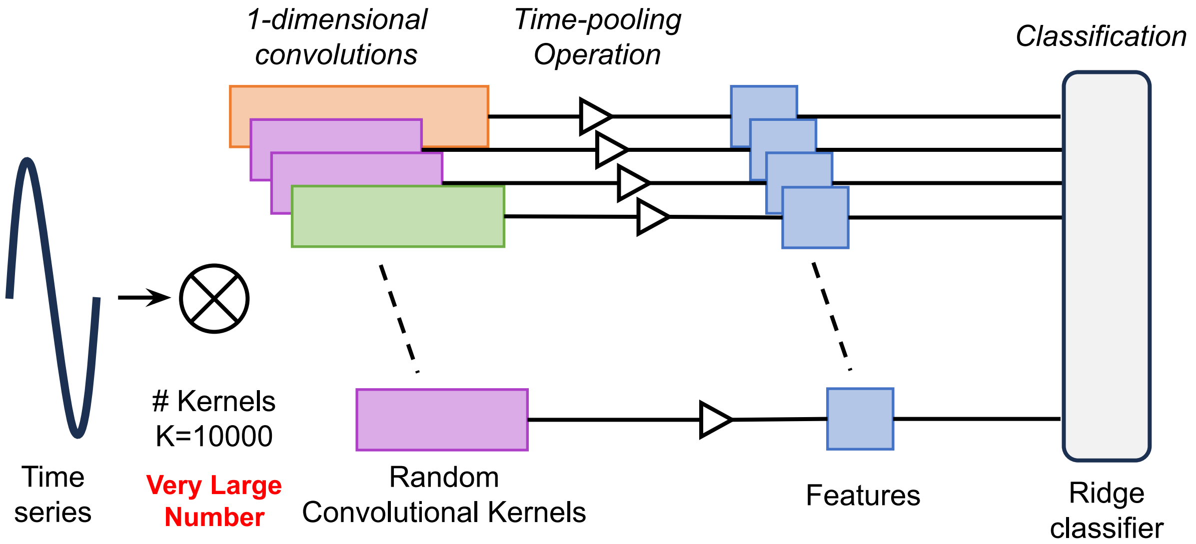
Random Convolutional Kernel Transform (ROCKET)* is a **transformation stage** which can be applied to **time-series data**.



* Dempster et al. [2020]

ROCKET models

Random Convolutional Kernel Transform (ROCKET)* is a **transformation stage** which can be applied to **time-series data.**

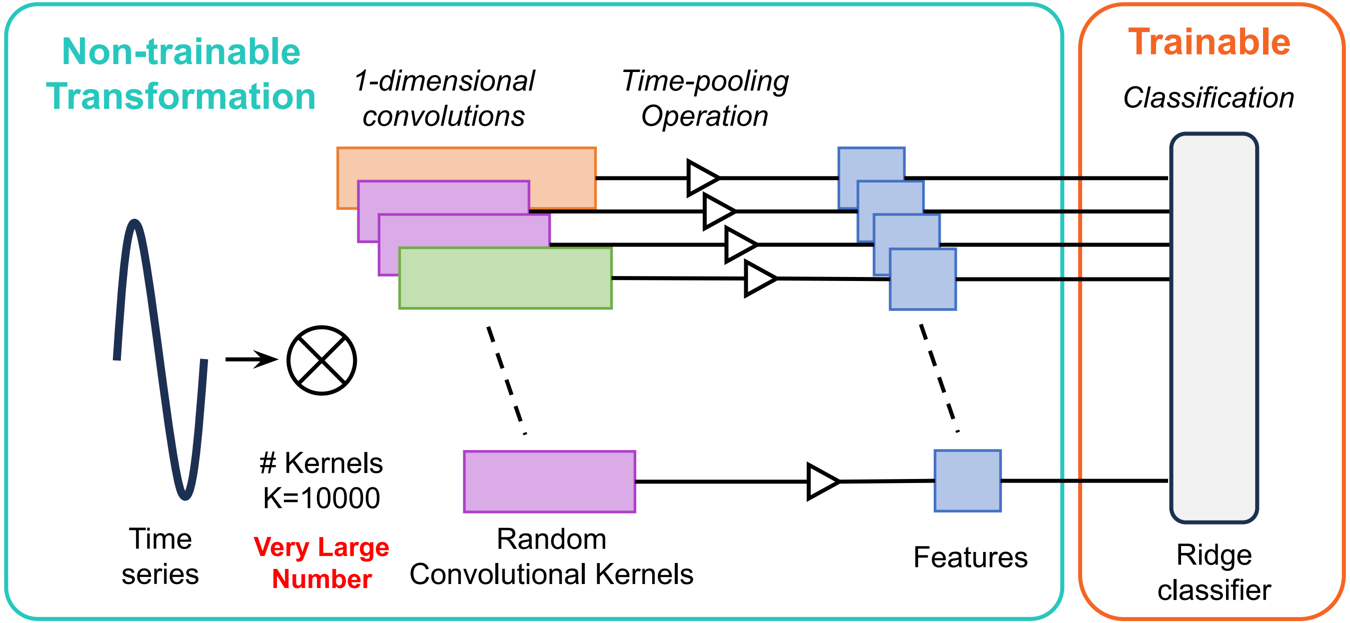


* Dempster et al. [2020]

ROCKET models

This is KEY!

Random Convolutional Kernel Transform (ROCKET)* is a **transformation stage** which can be applied to **time-series data.**



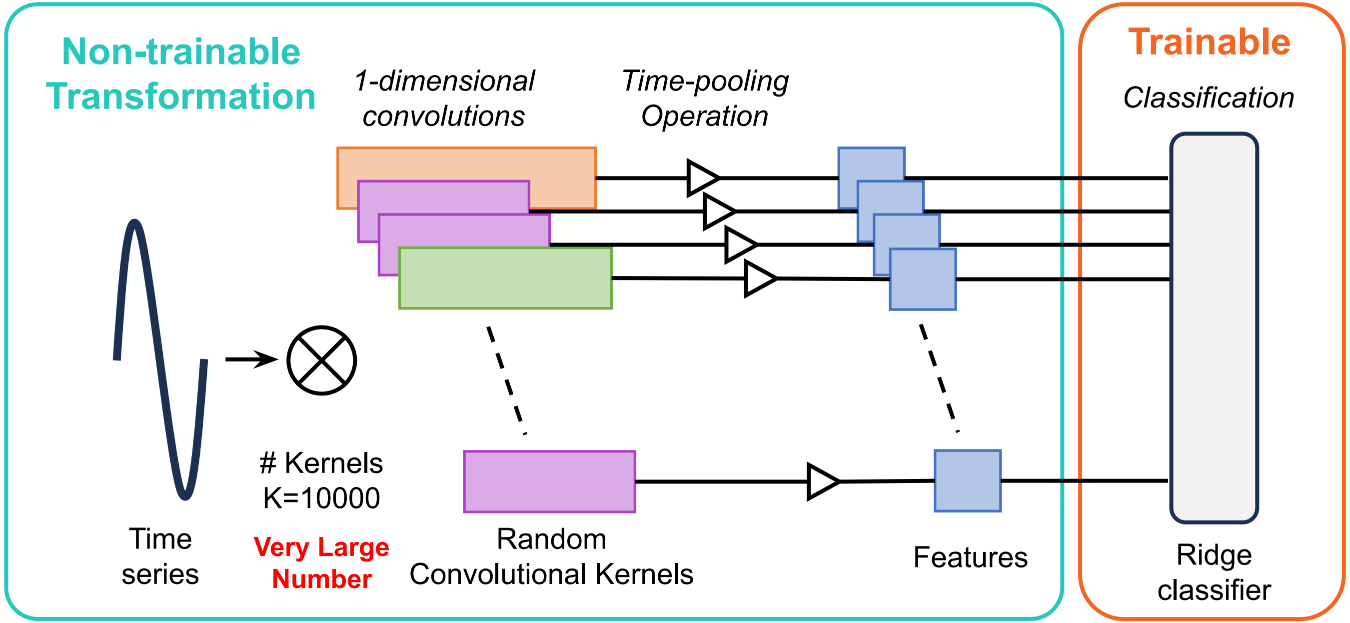
* Dempster et al. [2020]

ROCKET models

Random Convolutional Kernel Transform (ROCKET)* is a **transformation stage** which can be applied to **time-series data.**

This is KEY!

ROCKET
Kernels: Random
Pooling: MAX + PPV
 # Features: 20000



* Dempster et al. [2020]

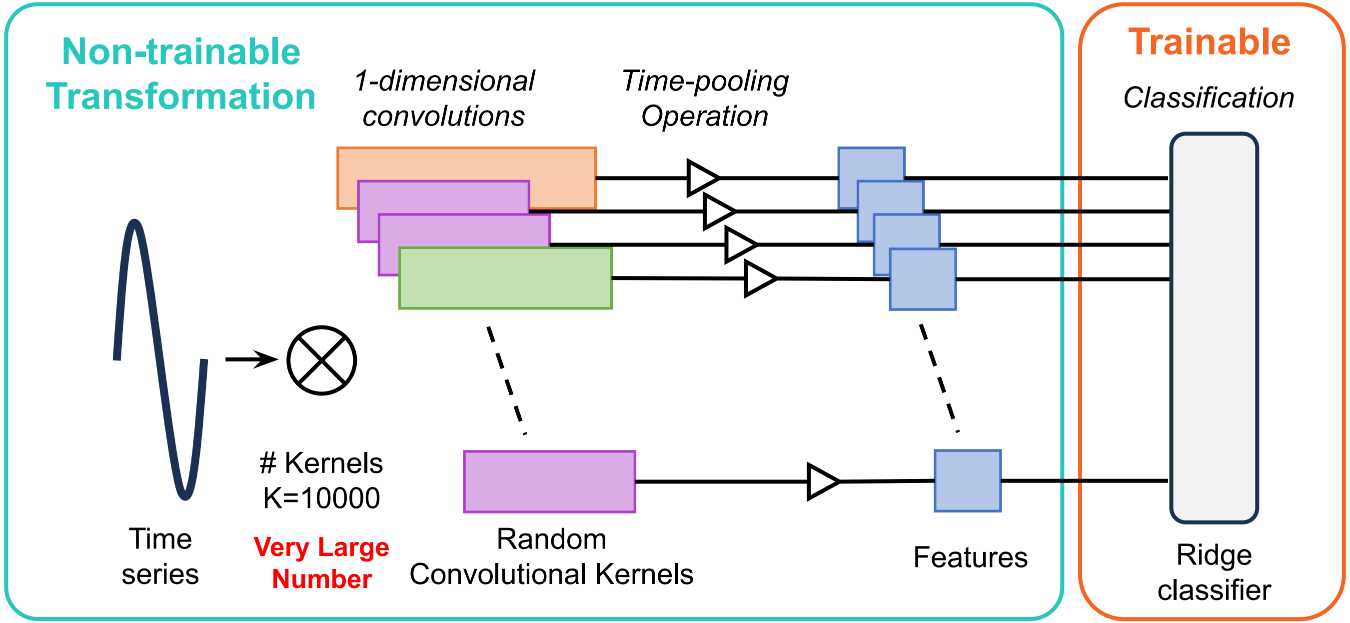
ROCKET models

Random Convolutional Kernel Transform (ROCKET)* is a **transformation stage** which can be applied to **time-series data.**

This is KEY!

ROCKET
Kernels: Random
Pooling: MAX + PPV
 # Features: 20000

MiniRocket
Kernels: Dictionary
Pooling: PPV
 # Features: 10000



* Dempster et al. [2020]

ROCKET models

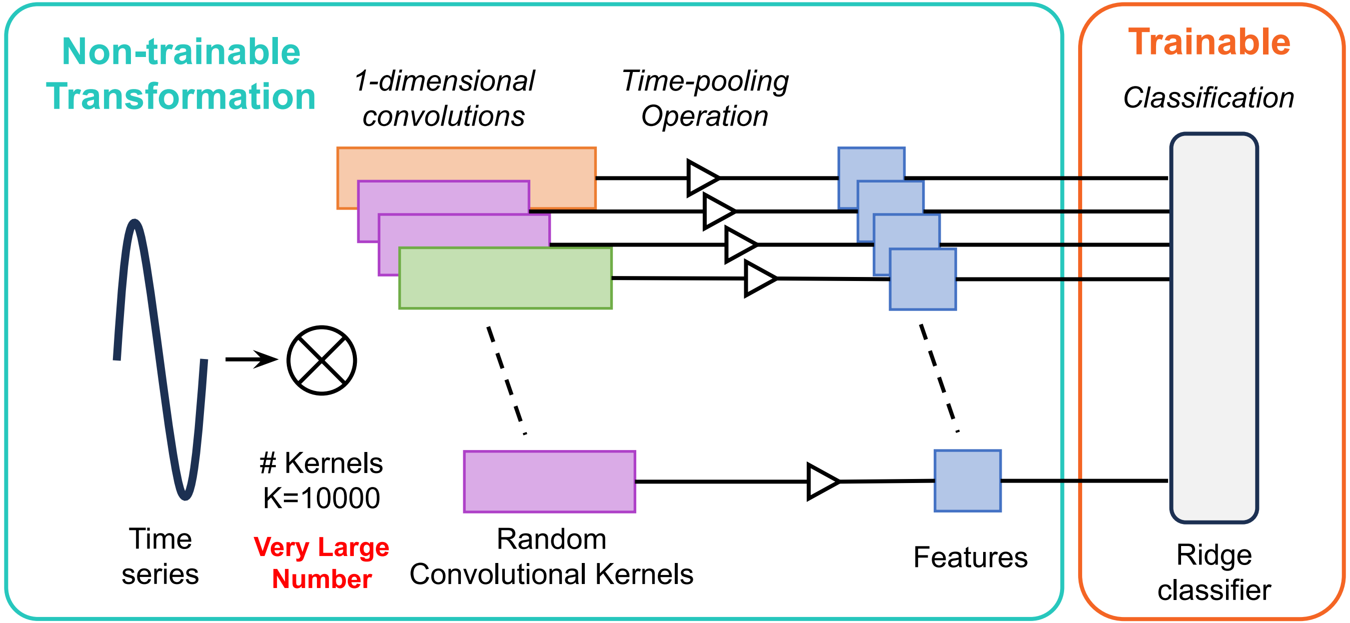
Random Convolutional Kernel Transform (ROCKET)* is a **transformation stage** which can be applied to **time-series data.**

This is KEY!

ROCKET
Kernels: Random
Pooling: MAX + PPV
 # Features: 20000

MiniRocket
Kernels: Dictionary
Pooling: PPV
 # Features: 10000

MultiRocket
Kernels: Dictionary
Pooling: PPV+MPV+MIPV+LSPV
 # Features: 50000



* Dempster et al. [2020]

ROCKET Strengths and Weaknesses

- State-of-the-art performance for TSC
- Fast and simple training
- Less prone to overfitting (less parameters)

ROCKET Strengths and Weaknesses

- State-of-the-art performance for TSC
- Fast and simple training
- Less prone to overfitting (less parameters)

- It produces many features (many useless)
- Scales poorly with the number of channels
- Difficult to interpret

Detach-ROCKET

SPRINGER NATURE Link

Find a journal   Publish with us   Track your research   Search

Home › Data Mining and Knowledge Discovery › Article

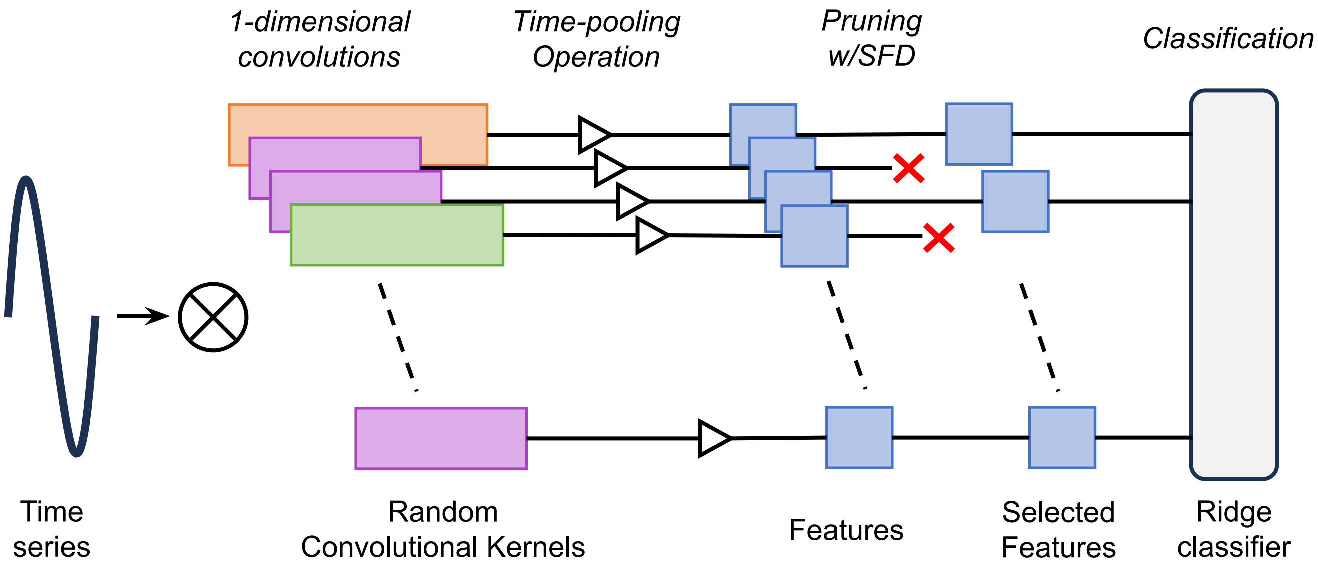
Detach-ROCKET: sequential feature selection for time series classification with random convolutional kernels

Open access | Published: 20 August 2024

Volume 38, pages 3922–3947, (2024)   Cite this article

Pruning ROCKET with SFD

We propose an algorithm to select the most relevant features called Sequential Feature Detachment (SFD)*.



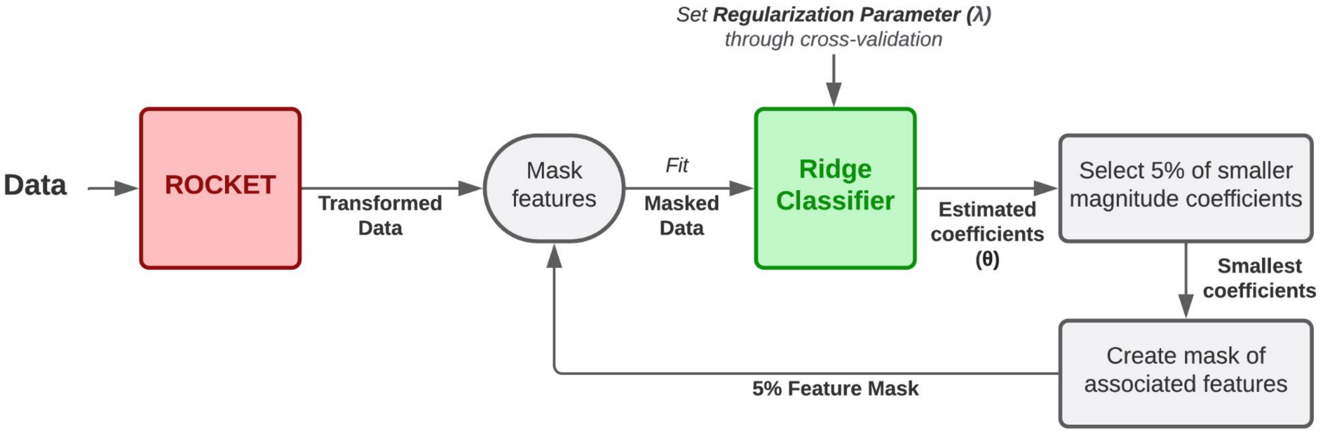
* Recently published in Data Min. Knowl. Discov., Uribarri et al. [2024]

Sequential Feature Detachment (SFD) algorithm

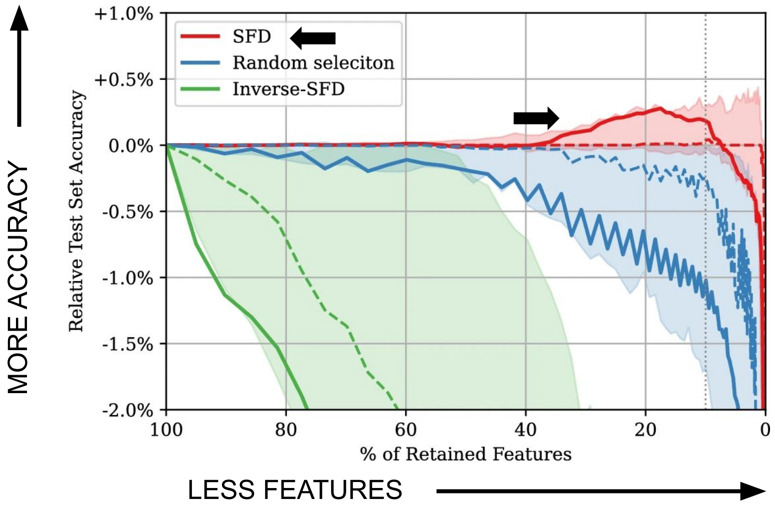
Algorithm 1 Sequential Feature Detachment

Parameters:
M: Number of steps
N: Number of initial features
K: Number of kernels
p: Proportion of eliminated features at each step

initialize: ROCKET model with K kernels
initialize: Active feature set $\mathbb{S}$ with $F = 2K$ features
Train ridge classifier with LOOCV to find λ value
for $t = 1$ to M **do**
  Train ridge classifier on $\mathbb{S}$ and obtain optimal coefficient $\hat{\theta}_k$ for each active feature
  Rank features based on $|\hat{\theta}_k|$
  Discard lowest p fraction of ranked features
  Update active feature set $\mathbb{S}$ with retained features
return Selected features at each step $\mathbb{S}_t$



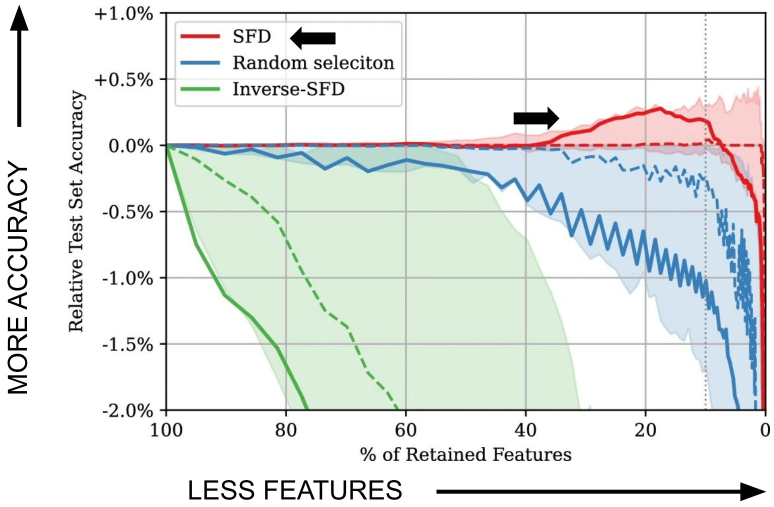
Detach-ROCKET: Pruned is better



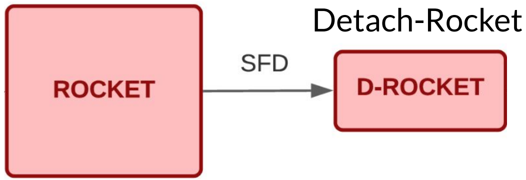
Testing on the **UCR** archive shows that our pruning can **improve model generalization** while drastically reducing model size.

* Recently published in Data Min. Knowl. Discov., Uribarri et al. [2024]

Detach-ROCKET: Pruned is better

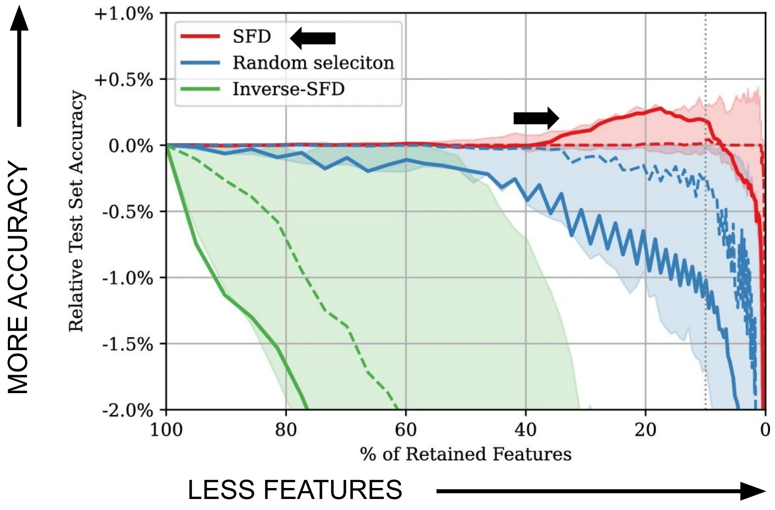


Testing on the **UCR** archive shows that our pruning can **improve model generalization** while drastically reducing model size.

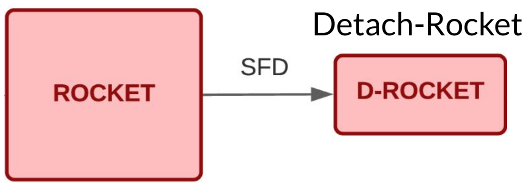


* Recently published in Data Min. Knowl. Discov., Uribarri et al. [2024]

Detach-ROCKET: Pruned is better



Testing on the **UCR** archive shows that our pruning can **improve model generalization** while drastically reducing model size.



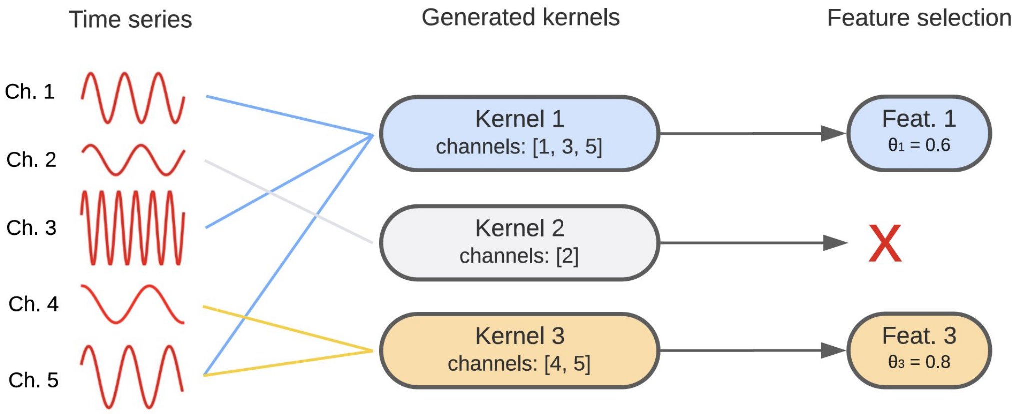
We also propose a methodology to **automatically select** the **optimal percentage of pruning.**

Original ROCKET (Dempster et al. 2020)	D-ROCKET (fixed 10%)		D-ROCKET (c=1)	
Test Acc.(%)	Features (%)	Test Acc.(%)	Features (%)	Test Acc.(%)
84.74 ± 0.62	10	**85.15 ± 0.78**	2.06	82.90 ± 2.74

* Recently published in Data Min. Knowl. Discov., Uribarri et al. [2024]

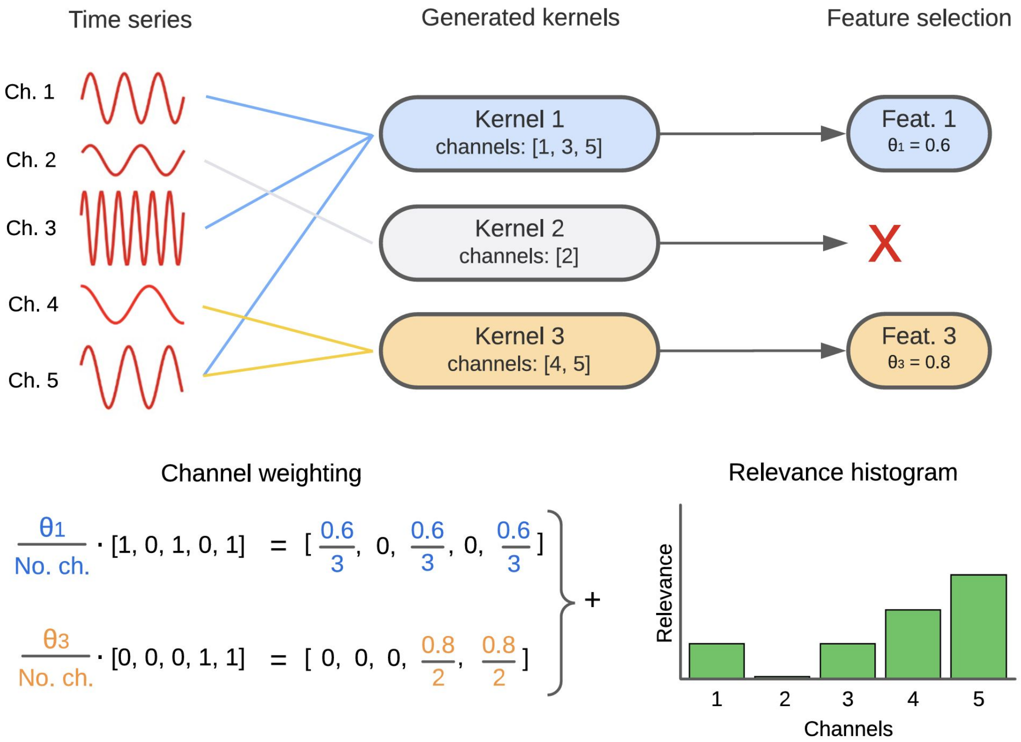
Detach-ROCKET: Channel Relevance

Detach-ROCKET: Channel Relevance



Selected features reveal the **relevant kernels** for classification.

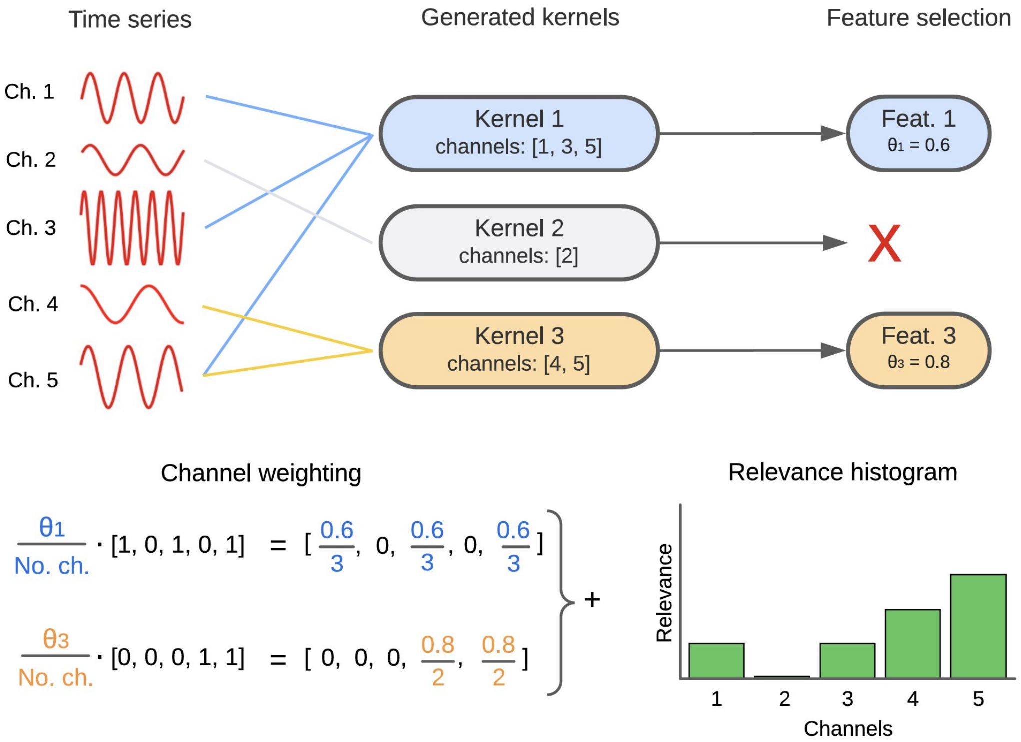
Detach-ROCKET: Channel Relevance



Selected features reveal the **relevant kernels** for classification.

A weighted sum on all selected kernels is the estimation of **channel relevance.**

Detach-ROCKET: Channel Relevance



Selected features reveal the **relevant kernels** for classification.

A weighted sum on all selected kernels is the estimation of **channel relevance.**

We add **post-hoc interpretability** to the model "for free".

ROCKET Strengths and Weaknesses

- State-of-the-art performance for TSC
- Fast and simple training
- Less prone to overfitting (less parameters)

- It produces many features (many useless)
- Scales poorly with the number of channels
- Difficult to interpret

Hands-on Time: Notebook 4