Deep Learning For Time Series Classification

Pr. Germain Forestier

IRIMAS, Université de Haute-Alsace

germain.forestier@uha.fr

DALL-E (OpenAI) illustration of InceptionTime

January 2024

EGC 2024 - Dijon

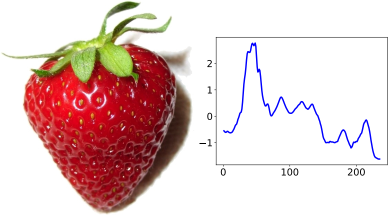
`https://germain-forestier.info/egc.pdf`

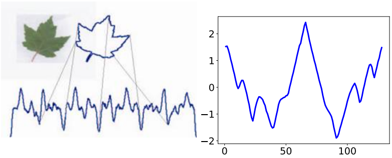
Outline

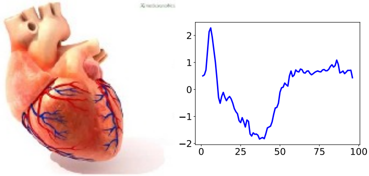
Contents

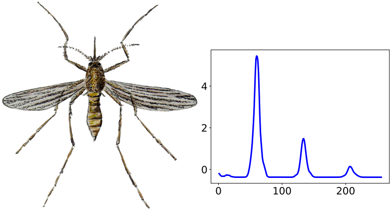
Time series are ubiquitous

- Type of data present in numerous applicative domains
- Allow to study the evolution in time of a process, a behavior, etc.





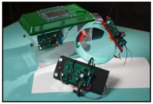


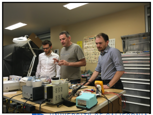


source : http://timeseriesclassification.com/

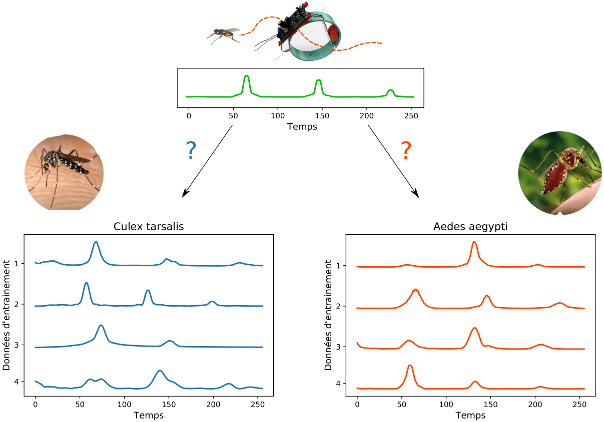
Time series classification

- Time series are regrouped in classes (e.g. Culex / Aedes)
- The goal is to assign a class to new time series





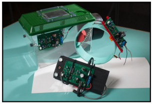
UC RIVERSIDE
UNIVERSITY OF CALIFORNIA

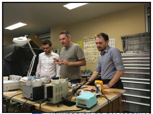


Petitjean, F., **Forestier, G.**, Webb, G. I., Nicholson, A. E., Chen, Y., & Keogh, E. (2014). Dynamic time warping averaging of time series allows faster and more accurate classification. In IEEE International Conference on Data Mining (pp. 470-479)
→ **ICDM 2023 10-year highest-impact paper Award**

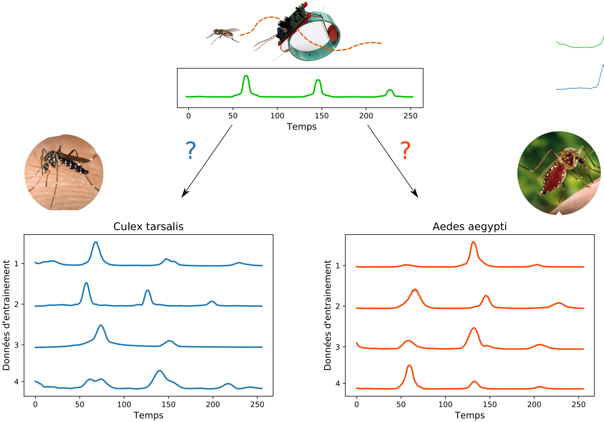
Time series classification

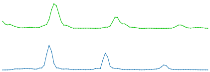
- Time series are regrouped in classes (e.g. Culex / Aedes)
- The goal is to assign a class to new time series





UC RIVERSIDE
UNIVERSITY OF CALIFORNIA

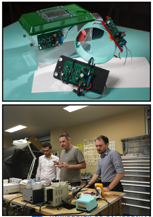


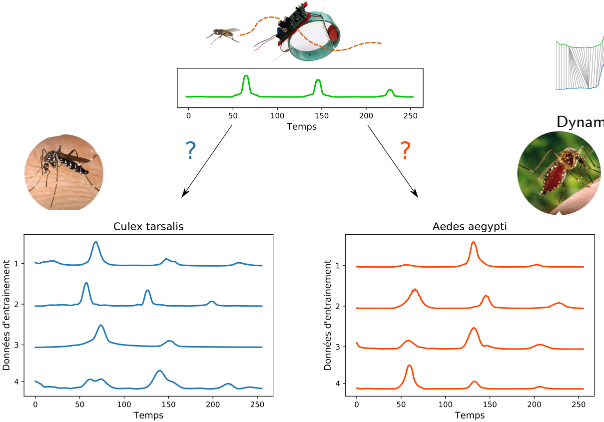


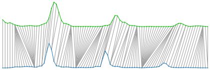
Petitjean, F., **Forestier, G.**, Webb, G. I., Nicholson, A. E., Chen, Y., & Keogh, E. (2014). Dynamic time warping averaging of time series allows faster and more accurate classification. In IEEE International Conference on Data Mining (pp. 470-479)
→ **ICDM 2023 10-year highest-impact paper Award**

Time series classification

- Time series are regrouped in classes (e.g. Culex / Aedes)
- The goal is to assign a class to new time series





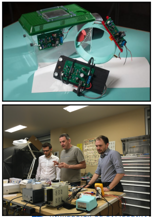


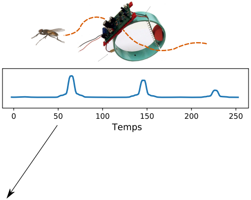
Alignement with
Dynamic Time Warping (DTW)

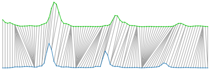
Petitjean, F., **Forestier, G.**, Webb, G. I., Nicholson, A. E., Chen, Y., & Keogh, E. (2014). Dynamic time warping averaging of time series allows faster and more accurate classification. In IEEE International Conference on Data Mining (pp. 470-479)
→ **ICDM 2023 10-year highest-impact paper Award**

Time series classification

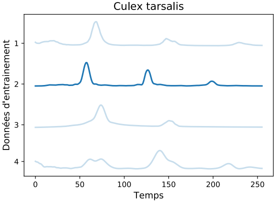
- Time series are regrouped in classes (e.g. Culex / Aedes)
- The goal is to assign a class to new time series

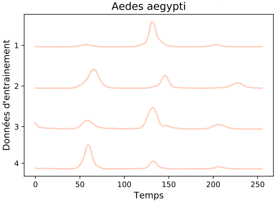






Alignement with
Dynamic Time Warping (DTW)





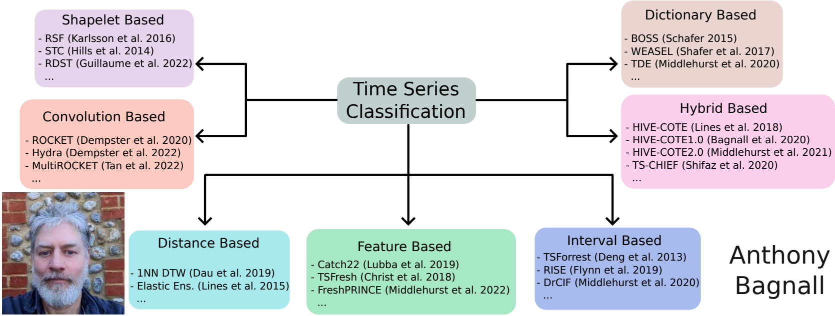
Petitjean, F., **Forestier, G.**, Webb, G. I., Nicholson, A. E., Chen, Y., & Keogh, E. (2014). Dynamic time warping averaging of time series allows faster and more accurate classification. In IEEE International Conference on Data Mining (pp. 470-479)

$\rightarrow$ **ICDM 2023 10-year highest-impact paper Award**

Contents

Taxonomy of methods

- Time Series Data can be used for various tasks including:
  - Clustering, representation learning, self-supervised learning, etc.
  - Extrinsic regression, forecasting, imputation, generation, etc.
  - Anomaly detection, similarity search, etc.
- In this talk, we focus on **classification**:

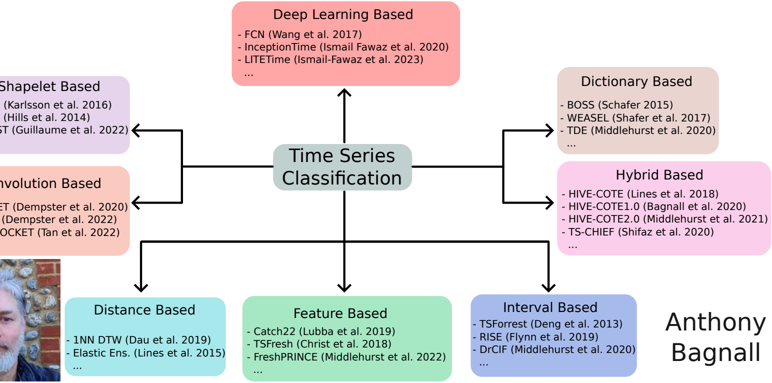


Anthony Bagnall

Middlehurst, M., Schäfer, P., & Bagnall, A. (2023). Bake off redux: a review and experimental evaluation of recent time series classification algorithms. arXiv preprint arXiv:2304.13029.

Taxonomy of methods

- Time Series Data can be used for various tasks including:
  - Clustering, representation learning, self-supervised learning, etc.
  - Extrinsic regression, forecasting, imputation, generation, etc.
  - Anomaly detection, similarity search, etc.
- In this talk, we focus on **classification**:



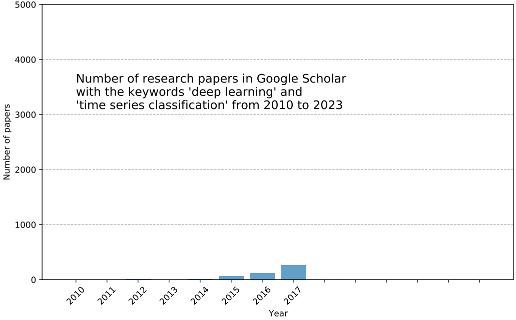
Anthony Bagnall

Middlehurst, M., Schäfer, P., & Bagnall, A. (2023). Bake off redux: a review and experimental evaluation of recent time series classification algorithms. arXiv preprint arXiv:2304.13029.

Contents

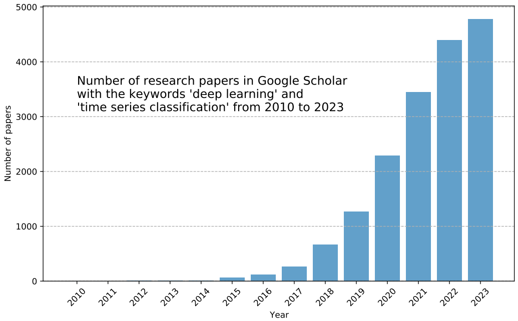
Why Deep Learning for TSC ?

- In 2017, reviewers started wondering what would be the performance of deep learning for TSC when reviewing non-deep learning TSC papers
- Deep learning had great success on other type of data (computer vision, NLP, etc.), so why not on time series ?
- Deep Learning models learn to extract features and perform classification with one set of parameters (end-to-end)



Why Deep Learning for TSC ?

- In 2017, reviewers started wondering what would be the performance of deep learning for TSC when reviewing non-deep learning TSC papers
- Deep learning had great success on other type of data (computer vision, NLP, etc.), so why not on time series ?
- Deep Learning models learn to extract features and perform classification with one set of parameters (end-to-end)

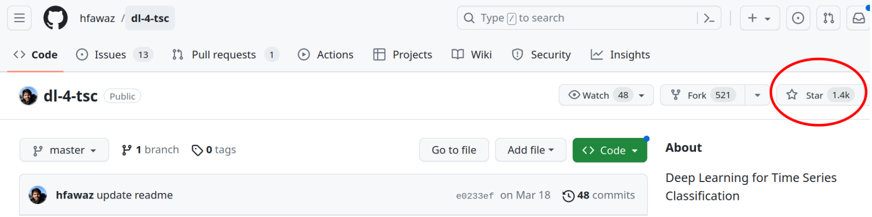


Contents

DL4TSC - 2019

In **2019**, we presented a study of **Deep Learning for Time Series Classification** (cited more than 2.7K times (GoogleScholar)) [1].

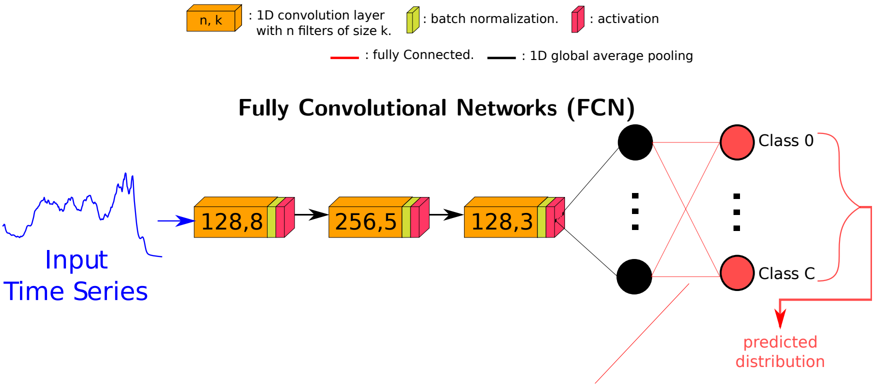
- We selected models with enough details (or available code) to reproduce the model's architecture
- We benchmarked all the models on the UCR archive [2]
- We published the code on Github for reproducibility and got very positive feedback (>1.4K stars)



[1] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019). Deep learning for time series classification: a review. Data mining and knowledge discovery, 33(4), 917-963.

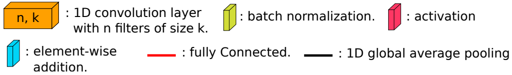
[2] Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C. C. M., Zhu, Y., Gharghabi, S. & Keogh, E. (2019). The UCR time series archive. IEEE/CAA Journal of Automatica Sinica

DL4TSC - Some Architectures

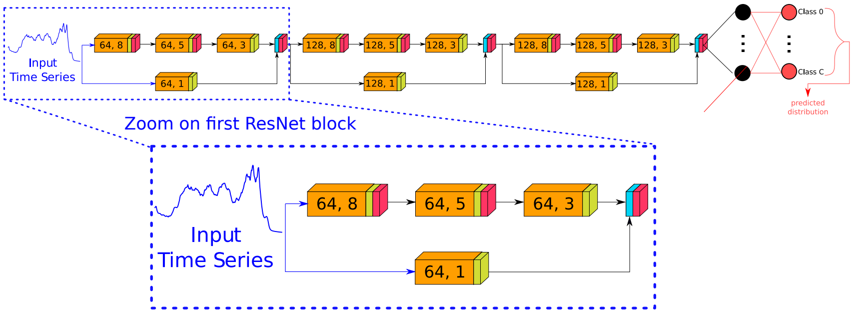


Wang, Z., Yan, W., & Oates, T. (2017, May). Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International joint conference on neural networks (IJCNN) (pp. 1578-1585). IEEE.

DL4TSC - Some Architectures



Inside each ResNet block:

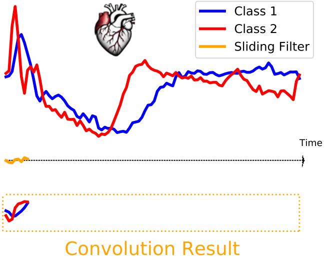


Wang, Z., Yan, W., & Oates, T. (2017, May). Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International joint conference on neural networks (IJCNN) (pp. 1578-1585). IEEE.

DL4TSC - Some Architectures

Convolutions on Images vs Time Series

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

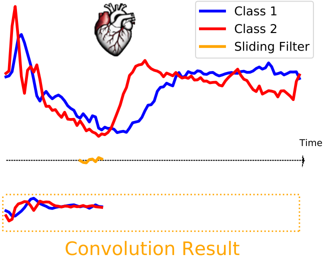


The result of applying a learned discriminative convolution on the ECG200 dataset

DL4TSC - Some Architectures

Convolutions on Images vs Time Series

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

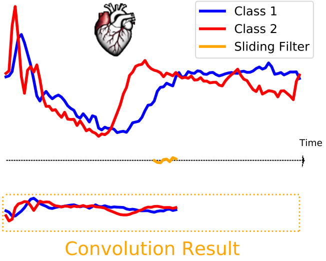


The result of applying a learned discriminative convolution on the ECG200 dataset

DL4TSC - Some Architectures

Convolutions on Images vs Time Series

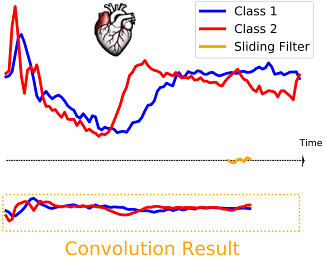
The result of a applying an edge detection convolution on an image



The result of applying a learned discriminative convolution on the ECG200 dataset

Convolutions on Images vs Time Series

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

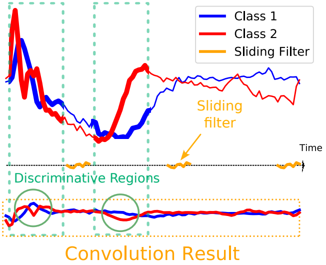


The result of applying a learned discriminative convolution on the ECG200 dataset

DL4TSC - Some Architectures

Convolutions on Images vs Time Series

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



The result of applying a learned discriminative convolution on the ECG200 dataset

DL4TSC - Experiments: Hyperparameters

Methods	Architecture							
	#layers	#conv	#invar	normalize	pooling	feature	activate	regularize
MLP	4	0	0	none	none	FC	ReLU	dropout
FCN	5	3	4	batch	none	GAP	ReLU	none
ResNet	11	9	10	batch	none	GAP	ReLU	none
Encoder	5	3	4	instance	max	Att	PReLU	dropout
MCNN	4	2	2	none	max	FC	sigmoid	none
t-LeNet	4	2	2	none	max	FC	ReLU	none
MCDCNN	4	2	2	none	max	FC	ReLU	none
Time-CNN	3	2	2	none	avg	Conv	sigmoid	none

Methods	Optimization						
	algorithm	valid	loss	epochs	batch	learning rate	decay
MLP	AdaDelta	train	entropy	5000	16	1.0	0.0
FCN	Adam	train	entropy	2000	16	0.001	0.0
ResNet	Adam	train	entropy	1500	16	0.001	0.0
Encoder	Adam	train	entropy	100	12	0.00001	0.0
MCNN	Adam	$\text{split}_{20\%}$	entropy	200	256	0.1	0.0
t-LeNet	Adam	train	entropy	1000	256	0.01	0.005
MCDCNN	SGD	$\text{split}_{33\%}$	entropy	120	16	0.01	0.0005
Time-CNN	Adam	train	mse	2000	16	0.001	0.0

DL4TSC - Experiments: Setup

- For univariate TSC: validation on the UCR archive [1]

- Each model was evaluated with the accuracy at test time
- We used the original train/test split for comparison
- We tested 8 models on 128 datasets with 5 random initializations
- A total of 5120 experiments ran on a cluster of 60+ GPUs

- For comparing classifiers, we performed a Friedman test followed by a Wilcoxon Signed Rank Test with Holm's alpha correction [1]
- Critical difference diagrams were used to visualize the classifiers' rank [3] and highlight statistical differences

[1] Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C. C. M., Zhu, Y., Gharghabi, S. & Keogh, E. (2019). The UCR time series archive. IEEE/CAA Journal of Automatica Sinica

[2] Garcia, S., & Herrera, F. (2008). An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. Journal of Machine Learning Research.

[3] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research

DL4TSC - Experiments: Setup

- For univariate TSC: validation on the UCR archive [1]

- Each model was evaluated with the accuracy at test time
- We used the original train/test split for comparison
- We tested 8 models on 128 datasets with 5 random initializations
- A total of 5120 experiments ran on a cluster of 60+ GPUs

- For comparing classifiers, we performed a Friedman test followed by a Wilcoxon Signed Rank Test with Holm's alpha correction [1]
- Critical difference diagrams were used to visualize the classifiers' rank [3] and highlight statistical differences

[1] Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C. C. M., Zhu, Y., Gharghabi, S. & Keogh, E. (2019). The UCR time series archive. IEEE/CAA Journal of Automatica Sinica

[2] Garcia, S., & Herrera, F. (2008). An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. Journal of Machine Learning Research.

[3] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research

DL4TSC - Experiments: Setup

- For univariate TSC: validation on the UCR archive [1]

- Each model was evaluated with the accuracy at test time
- We used the original train/test split for comparison
- We tested 8 models on 128 datasets with 5 random initializations
- A total of 5120 experiments ran on a cluster of 60+ GPUs

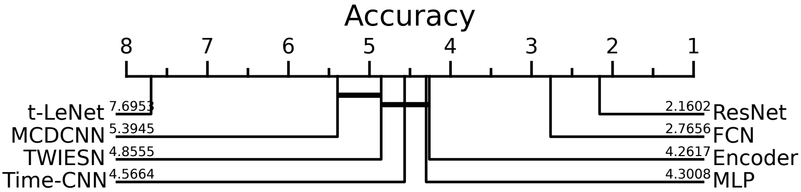
- For comparing classifiers, we performed a Friedman test followed by a Wilcoxon Signed Rank Test with Holm's alpha correction [1]
- Critical difference diagrams were used to visualize the classifiers' rank [3] and highlight statistical differences

[1] Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C. C. M., Zhu, Y., Gharghabi, S. & Keogh, E. (2019). The UCR time series archive. IEEE/CAA Journal of Automatica Sinica

[2] Garcia, S., & Herrera, F. (2008). An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. Journal of Machine Learning Research.

[3] Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research

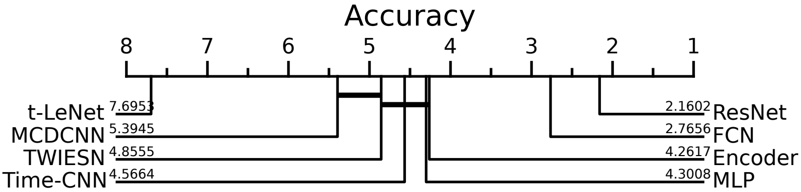
DL4TSC - Experiments: Results



- A Critical Difference Diagram is a visualization technique that displays the performance of multiple classifiers on multiple datasets
- The classifiers in a CDD are ordered following their **average rank**
- Ranking is made following the accuracy on all test sets of the 128 datasets
- Each clique (black line between classifiers) represent a statistically non-significant difference in performance between linked classifiers on the 128 datasets of the UCR archive (using Wilcoxon Signed-Rank test + Holm correction)

Benavoli, Alessio, Giorgio Corani, and Francesca Mangili. "Should we really use post-hoc tests based on mean-ranks?." The Journal of Machine Learning Research 17.1 (2016): 152-161.

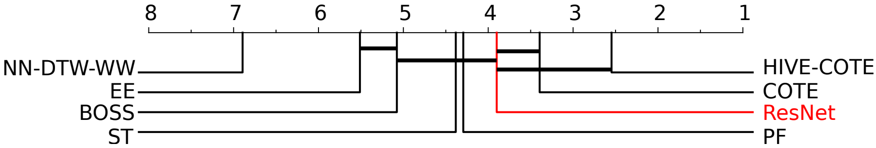
DL4TSC - Experiments: Results



- A Critical Difference Diagram is a visualization technique that displays the performance of multiple classifiers on multiple datasets
- The classifiers in a CDD are ordered following their **average rank**
- Ranking is made following the accuracy on all test sets of the 128 datasets
- Each clique (black line between classifiers) represent a statistically non-significant difference in performance between linked classifiers on the 128 datasets of the UCR archive (using Wilcoxon Signed-Rank test + Holm correction)

Benavoli, Alessio, Giorgio Corani, and Francesca Mangili. "Should we really use post-hoc tests based on mean-ranks?." The Journal of Machine Learning Research 17.1 (2016): 152-161.

DL4TSC - Experiments: Comparison



- ResNet (best DL model) performed as good as state-of-the-art non-deep models for Time Series Classification on the UCR archive (in 2019).
- No significance was found in difference in performance between ResNet and Hive-COTE (ensemble of five classifiers).
- Showcasing the high performance of deep models for TSC

Bagnall, Anthony, et al. "The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances." Data mining and knowledge discovery 31 (2017): 606-660.

Contents

InceptionTime

Inception:

- Originally proposed by Google for image recognition problems [1]
- Further developed to reach state-of-the-art results on ImageNet [2]
- Apply convolutions of different length to capture various patterns
- Use a bottleneck layer in order to reduce the number of parameters

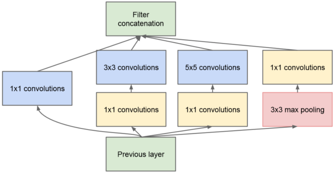
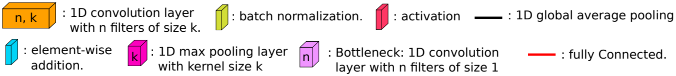


Figure: Inception module for image recognition [1]

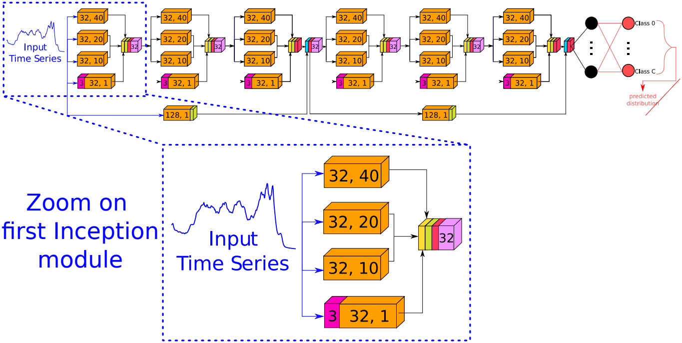
[1] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-9).

[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).

InceptionTime



Inception architecture for TSC



InceptionTime is an ensemble of five Inception models.

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

InceptionTime: Training time comparison (HIVE-COTE)

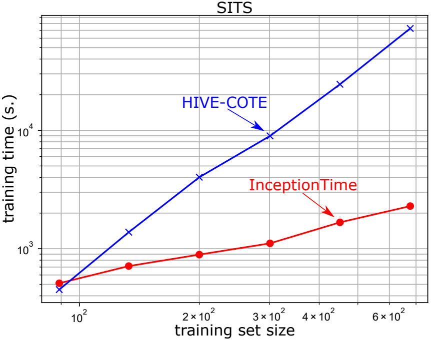


Figure: Training time as a function of the training set size for the SITS dataset

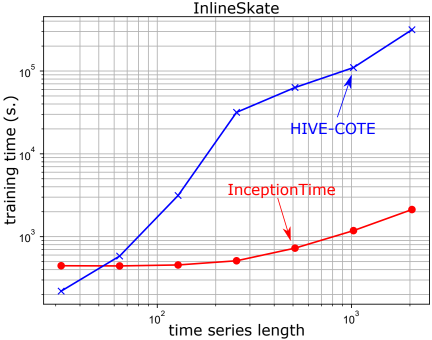
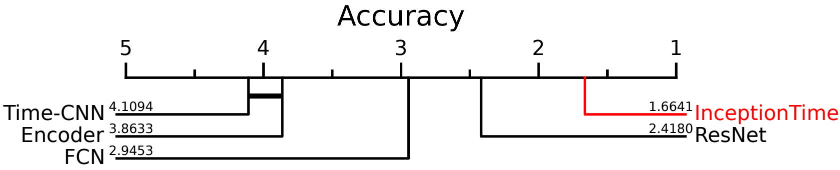


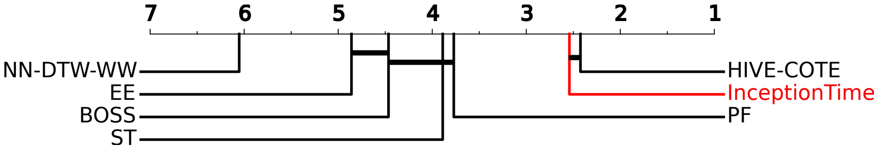
Figure: Training time as a function of the series length for the InlineSkate dataset

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

InceptionTime: Performance with Best Deep Learners



In 2020, InceptionTime became the state-of-the-art Deep Learning model for Time Series Classification on the UCR archive.



InceptionTime performed significantly better than most non-deep models with no significant difference with HIVE-COTE

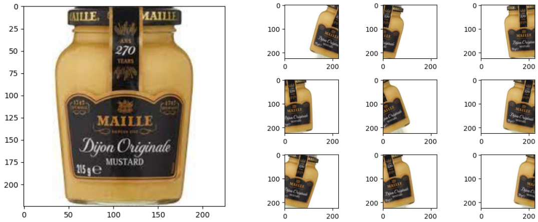
Ismail Fawaz, Hassan, et al. "InceptionTime: Finding alexnet for time series classification." Data Mining and Knowledge Discovery 34.6 (2020): 1936-1962.

Contents

Regularization Techniques: Data Augmentation

Data augmentation:

- Data augmentation aims at increasing the diversity of the train set
- In computer vision it consists in altering input images
- Alteration can be: cropping, padding, horizontal flipping, etc.

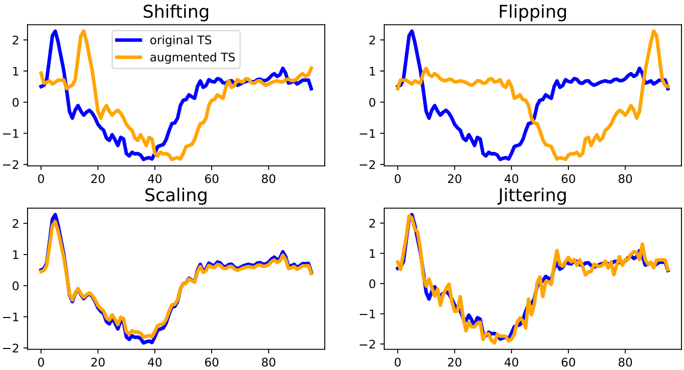


[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems

[2] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In Proceedings of International Conference on Learning

Regulatization Techniques: Data Augmentation

Augmentation on images can be adapted to 1D time series:



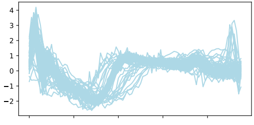
- How to design augmentation techniques **specific** for time series data ?

Pialla, Gautier, et al. "Data augmentation for time series classification with deep learning models." AALTD/ECML-PAKDD 2022.

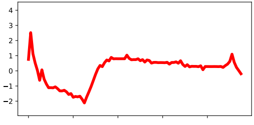
Regularization Techniques: Data Augmentation

How to create synthetic time series ?

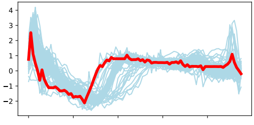
- We averaged a set of time series and took the average as a new synthetic object
- We used weighted averages to generate multiple synthetic objects



Set of time series



Average time series



Augmented dataset

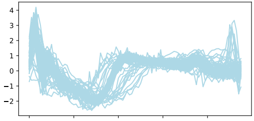
Petitjean, F., Ketterlin, A., & Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. Pattern Recognition, 44(3), 678-693.

Forestier, Germain, et al. "Generating synthetic time series to augment sparse datasets." 2017 IEEE International Conference on Data Mining (ICDM). IEEE, 2017.

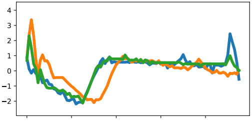
Regularization Techniques: Data Augmentation

How to create synthetic time series ?

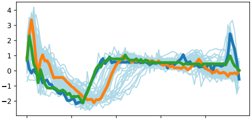
- We averaged a set of time series and took the average as a new synthetic object
- We used weighted averages to generate multiple synthetic objects



Set of time series



Weighted average time series



Augmented dataset

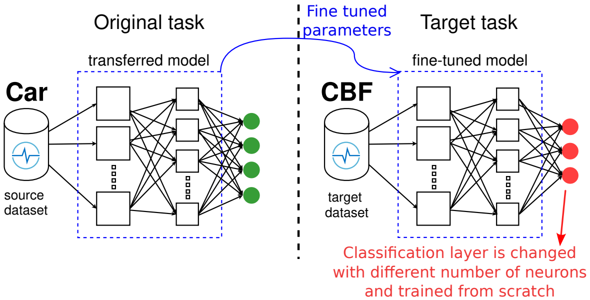
Petitjean, F., Ketterlin, A., & Gançarski, P. (2011). A global averaging method for dynamic time warping, with applications to clustering. Pattern Recognition, 44(3), 678-693.

Forestier, Germain, et al. "Generating synthetic time series to augment sparse datasets." 2017 IEEE International Conference on Data Mining (ICDM). IEEE, 2017.

Regularization Techniques: Transfer Learning

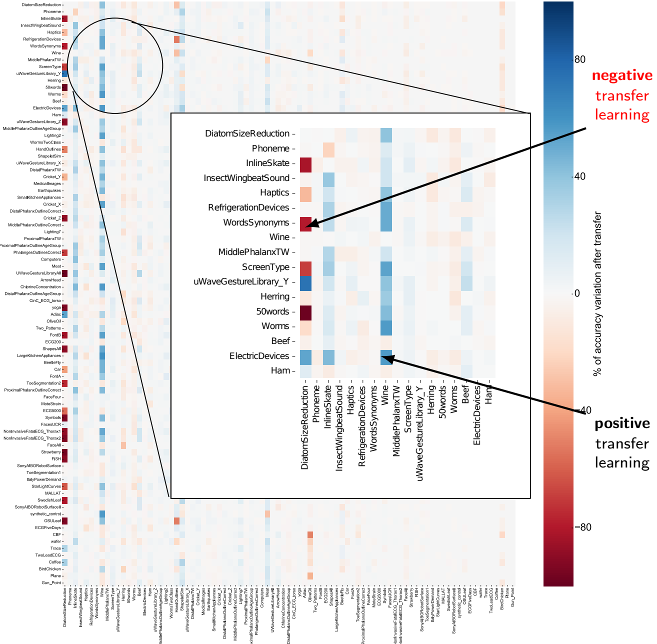
Transfer Learning:

1. Train a base network on a source dataset
2. Transfer the learned features (the network's weights) to a second network and adapt the last layer (class-dependent)
3. Re-train or fine-tune the transferred network on a target dataset

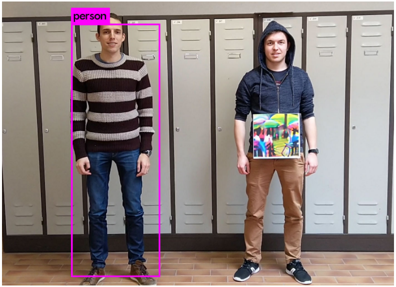


Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2018). Transfer learning for time series classification. IEEE International Conference on Big Data.

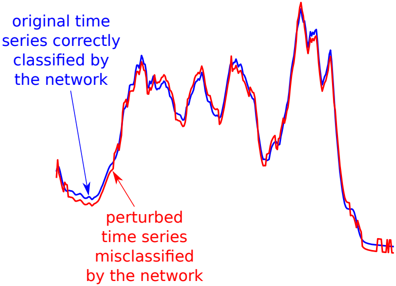
Regularization Techniques: Transfer Learning



Regularization Techniques: Adversarial Attacks



Object recognition [1]



Time series classification [2]

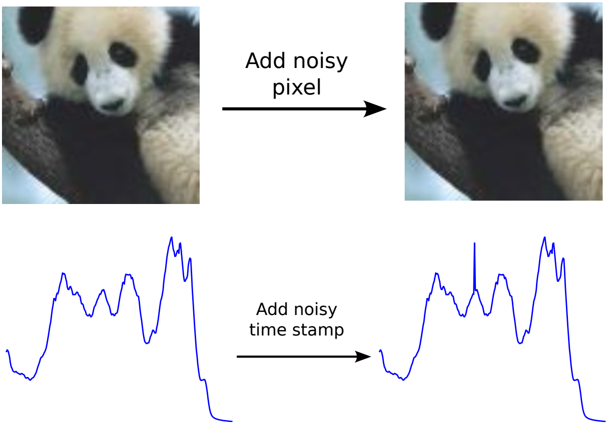
Figure: Adversarial attacks on deep learning systems.

[1] Van Ranst, W., Thys, S., & Goedemé, T. (2019). Fooling automated surveillance cameras: adversarial patches to attack person detection. In CVPR Workshop on The Bright and Dark Sides of Computer Vision: Challenges and Opportunities for Privacy and Security.

[2] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. (2019). Adversarial Attacks on Deep Neural Networks for Time Series Classification. International Joint Conference on Neural Networks (IJCNN).

Regularization Techniques: Adversarial Attacks

It is easier to see attacks on time series compared to images:



Goodfellow, I.; Shlens, J.; Szegedy, C. Explaining and Harnessing Adversarial Examples. In Proceedings of the International Conference on Learning Representations

Regularization Techniques: Adversarial Attacks

Adversarial attacks on deep neural networks:

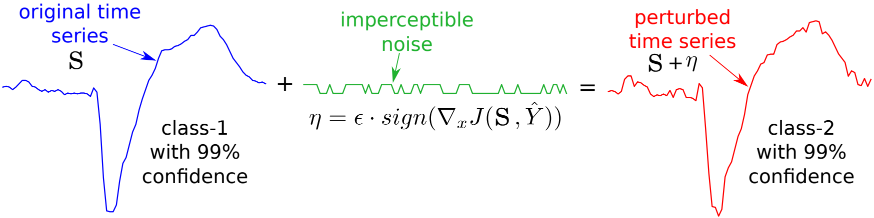


Figure: Example of perturbing the classification of an input time series (from the TwoLeadECG dataset) by adding an imperceptible noise computed using the Fast Gradient Sign Method (FGSM). Figure inspired from [1].

[1] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. International Conference on Learning Representations.

[2] Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., Muller, P. A. (2019). Adversarial attacks on deep neural networks for time series classification. In International Joint Conference on Neural Networks (IJCNN)

Regularization Techniques: Adversarial Attacks

Smooth Perturbations for Time Series Adversarial Attacks

- A novel adversarial attack that produces smooth perturbations

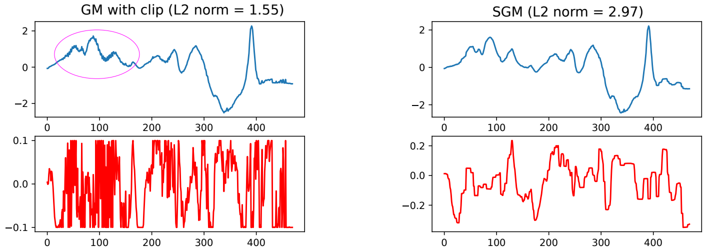


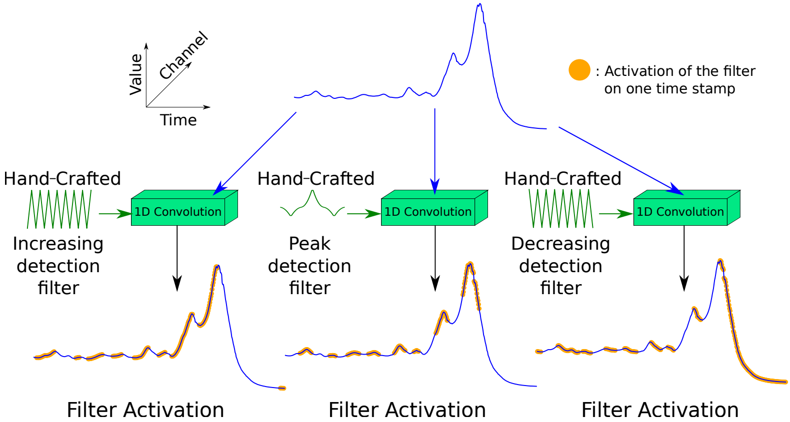
Figure: Time series from the Beef dataset. All methods perturbed time series (blue) and generated noise (red). The purple circles show the presence of saw- tooth on the perturbed time series. [1, 2].

[1] Pialla et al. (2022). Smooth Perturbations for Time Series Adversarial Attacks, Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)

[2] Pialla et al. (2023). Time series adversarial attacks: an investigation of smooth perturbations and defense approaches."International Journal of Data Science and Analytics

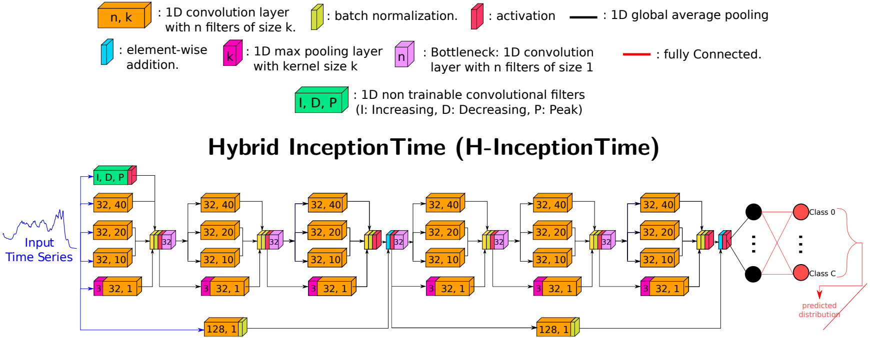
Regularization Techniques: Hand-Crafted Filters

In 2022, we proposed new hand-crafted convolutional filters:



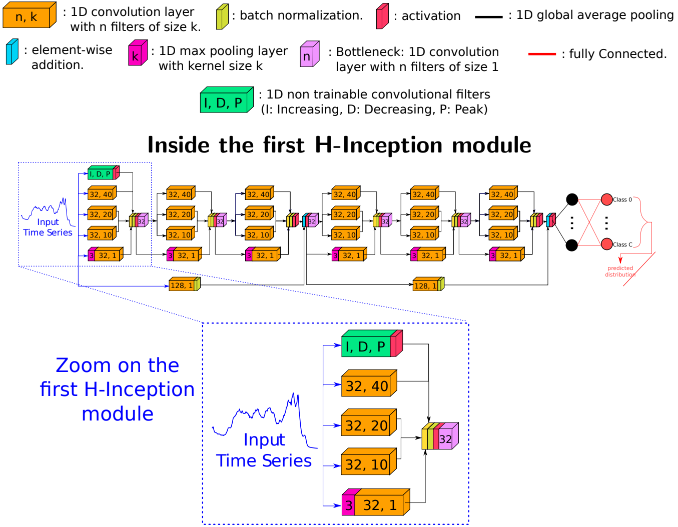
Ismail-Fawaz, A., Devanne, M., Weber, J., & Forestier, G. (2022). Deep learning for time series classification using new hand-crafted convolution filters. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 972-981)

Regularization Techniques: Hand-Crafted Filters



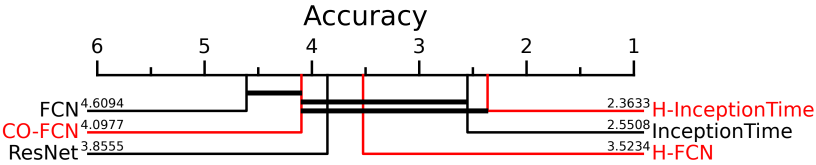
Ismail-Fawaz, A., Devanne, M., Weber, J., & Forestier, G. (2022). Deep learning for time series classification using new hand-crafted convolution filters. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 972-981)

Regularization Techniques: Hand-Crafted Filters



Ismail-Fawaz, A., Devanne, M., Weber, J., & Forestier, G. (2022). Deep learning for time series classification using new hand-crafted convolution filters. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 972-981)

Regularization Techniques - Hand-Crafted Filters



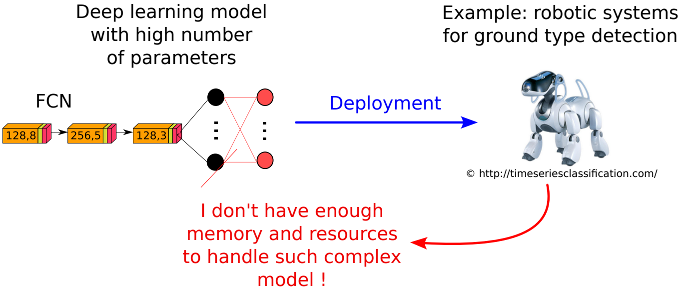
- Both CO-FCN and H-FCN can beat FCN
- H-FCN is better than ResNet which means hand-crafted filters helped FCN generalize better than residual connections can
- H-InceptionTime beats InceptionTime, concluding that hand-crafted filters help generalize the models

Ismail-Fawaz, A., Devanne, M., Weber, J., & Forestier, G. (2022). Deep learning for time series classification using new hand-crafted convolution filters. In 2022 IEEE International Conference on Big Data (Big Data) (pp. 972-981)

Contents

Reducing Models Size

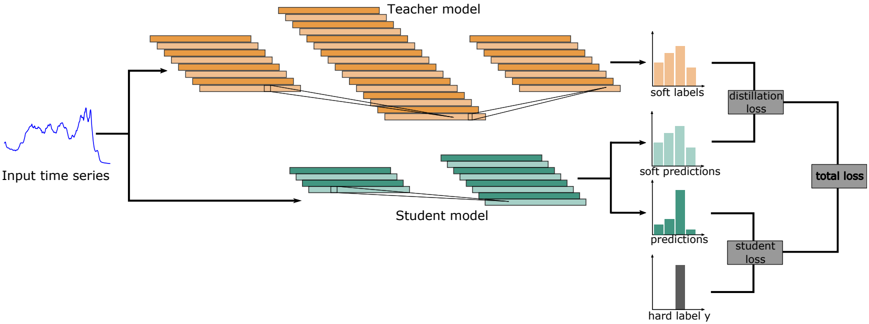
Deep learning models can be difficult to deploy:



- Hand-crafted convolutional filters in H-FCN demonstrate that complexity is not always the key to better performance.
- Is it possible to reduce complexity without compromising performance?

Reducing Models Size: Knowledge Distillation

Learn to boost a smaller version a model using Knowledge Distillation:

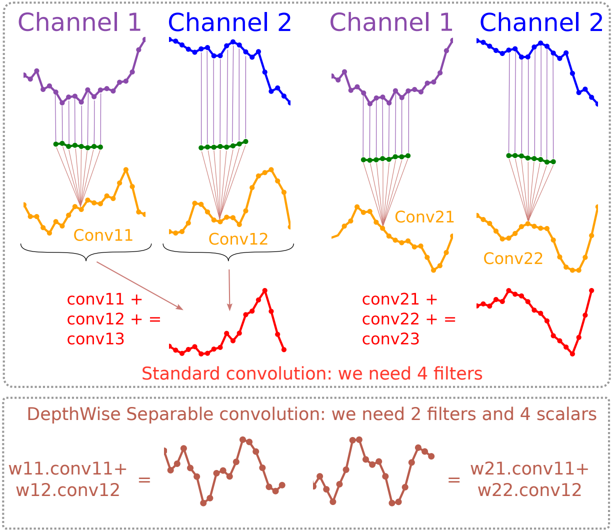


- Boosting a **student model** (small) with information coming from a pre-trained **teacher model** (large)

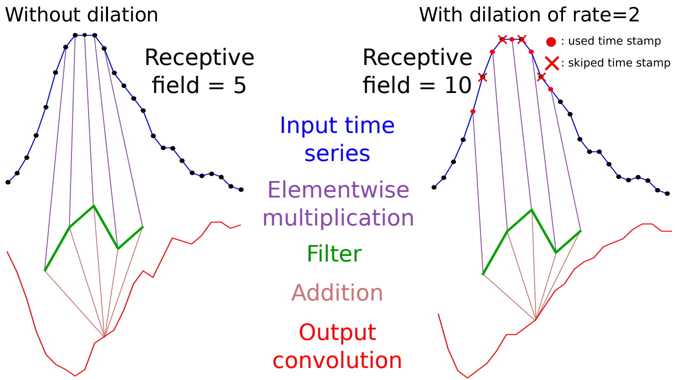
Ay, Emel, et al. "A study of knowledge distillation in fully convolutional network for time series classification." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE.

Reducing Models Size: DepthWise Separable Convolution

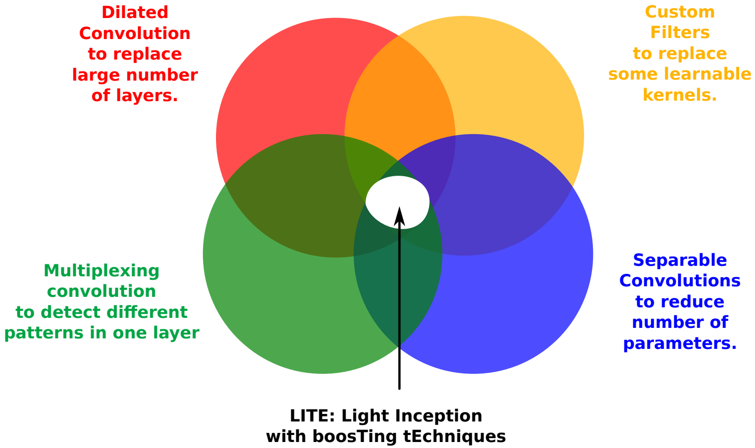
Suppose we have an input time series with 2 channels, the goal is to apply a standard convolution of kernel size 8 and the desired output dimension is 2.



Reducing Models Size: Dilated Convolutions

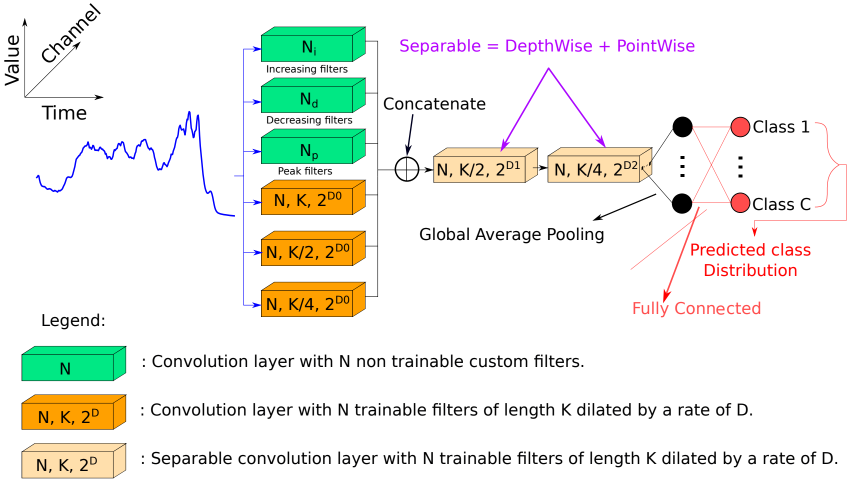


Reducing Models Size: LITE



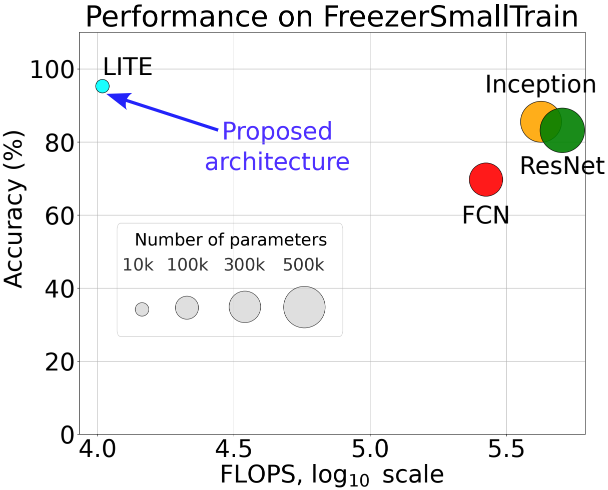
Ismail-Fawaz, A. et al. (2023). LITE: Light Inception with boosTing tEchniques for Time Series Classification. IEEE International Conference on Data Science and Advanced Analytics (DSAA)

Reducing Models Size: LITE Architecture



Ismail-Fawaz, A. et al. (2023). LITE: Light Inception with boosTing tEchniques for Time Series Classification. IEEE International Conference on Data Science and Advanced Analytics (DSAA)

Reducing Models Size: LITE - Efficiency Comparison



Ismail-Fawaz, A. et al. (2023). LITE: Light Inception with boosTing tEchniques for Time Series Classification. IEEE International Conference on Data Science and Advanced Analytics (DSAA)

Reducing Models Size: LITE - Efficiency Comparison

Table: Comparison between the proposed method with FCN, ResNet and Inception without ensemble.

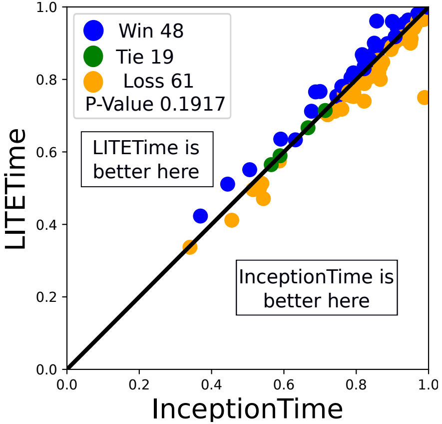
Models	Number of parameters	Training Time (mins)	Testing Time (mins)	CO_2 (g)
Inception	420,192	2,419.2	1.296	0.2928
ResNet	504,000	2,750.4	1.008	0.3101
FCN	264,704	2,491.2	0.4464	0.2623
LITE	**9,814**	**829.8**	**0.72**	**0.1048**

CO_2 calculation: https://codecarbon.io/

Ismail-Fawaz, A. et al. (2023). LITE: Light Inception with boosTing tEchniques for Time Series Classification. IEEE International Conference on Data Science and Advanced Analytics (DSAA)

Reducing Models Size: LITETime vs InceptionTime

Pairwise plot where each dot represents one dataset (128 in total):



Ismail-Fawaz, A. et al. (2023) LITE: Light Inception with boosTing tEchniques for Time Series Classification. 2023 IEEE International Conference on Data Science and Advanced Analytics (DSAA)

Contents

Other Work

Additional work that was not presented today:

- Ensembling deep network [1]
- Self Representation Learning for Time Series Data [2]
- Deep Clustering [3]
- Deep Time Series Averaging [4]
- Time Series Averaging with ShapeDBA (a new version of DBA) [5]
- The Multiple Comparison Matrix (MCM) [6]
- Deep Models Explainability [7, 8]

[1] Ismail Fawaz, H. et al. (2019). Deep neural network ensembles for time series classification. International Joint Conference on Neural Networks (IJCNN)

[2] Ismail-Fawaz, A. et al. (2023) Enhancing Time Series Classification with Self-Supervised Learning." In International Conference on Agents and Artificial Intelligence (ICAART)

[3] Lafabregue, B. et al. (2022) End-to-end deep representation learning for time series clustering: a comparative study. Data Mining and Knowledge Discovery

[4] Tsegamlak, T. et al. (2023) Estimating time series averages from latent space of multi-tasking neural networks. Knowledge and Information Systems

[5] Ismail-Fawaz, A. et al. (2023) ShapeDBA: Generating Effective Time Series Prototypes using ShapeDTW Barycenter Averaging. ECML/PKDD Workshop AALTD

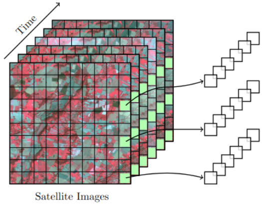
[6] Ismail-Fawaz, A. et al. (2023) An Approach to Multiple Comparison Benchmark Evaluations that is Stable Under Manipulation of the Comparate Set. arXiv preprint

[7] Ismail Fawaz, H. et al. (2018) Evaluating surgical skills from kinematic data using convolutional neural networks." MICCAI

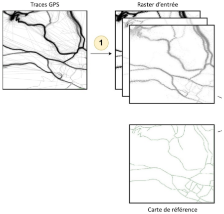
[8] Lafabregue, B. et al (2021). Grad Centroid Activation Mapping for Convolutional Neural Networks. IEEE International Conference on Tools with Artificial Intelligence (ICTAI)

Applications

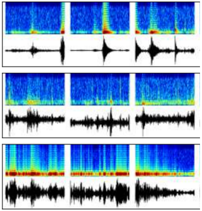
Examples of application:



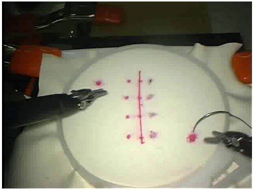
Series of Remote Sensing Images



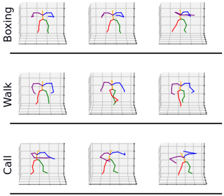
Generating Hiking Maps



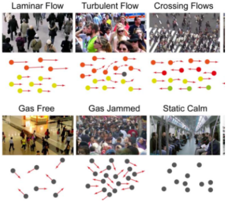
Seismology



Surgical Data Science



Body movements

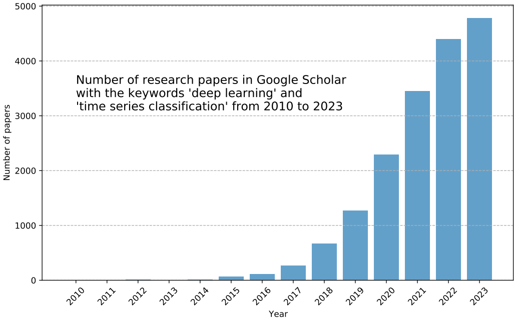


Crowd movement

Other Work

Hot Topics:

- Data Augmentation for Time Series Data
- Self-supervised Learning / Representation Learning
- Foundation Models for Time Series Data [1]
- Transformers for Time Series Classification [2]
- Multivariate Time Series Classification



[1] Ismail-Fawaz, A. (2023). Finding Foundation Models for Time Series Classification with a PreText Task. arXiv preprint arXiv:2311.14534.

[2] Foumani, N. M., Tan, C. W., Webb, G. I., & Salehi, M. (2023). Improving Position Encoding of Transformers for Multivariate Time Series Classification. arXiv preprint arXiv:2305.16642.

Contents

Reproducibility

Reproducibility:

- We believe in reproducible research - we release open source implementations of our work with each paper
- Starting 2022, all codes are on the same Github: `https://github.com/MSD-IRIMAS`
- Most of the methods are implemented in aeon, a python package for time series machine learning: `https://github.com/aeon-toolkit/aeon`

Contents

Open position in our team in 2024

Open position of Associate Professor (Maitre de conférences) in 2024 in our team!

Research: AI / time series (IRIMAS lab, MSD team)
Teaching: Computer Science Engineering School (ENSISA)

Lac Noir (massif des Vosges) - MSD team annual hike

"Mulhouse en tête de liste des villes où concilier travail et logement" (29/11/23)

https://www.leparisien.fr/economie/mulhouse-en-tete-de-liste-des-villes-ou-concilier-travail-et-logement-il-y-a-une-energie-et-un-potentiel-enormes-29-11-2023-C5M5YY3J5ZCSLKN7GREIA6KQHM.php

Takeway

Maxime Devanne

Germain Forestier

Ali Ismail-Fawaz

Hassan Ismail Fawaz

Jonathan Weber

Bapstiste Lafabrege

Javidan Abdullayev

Gautier Pialla

Tsegamlak Terefe

Joachim Rimpot

- Deep Learning for Time Series Classification is an active field of research
- We believe that the UCR is the best way to benchmark different approaches in TSC, however more application datasets should be used for specific use-case
- The DL4TSC web-page: **https://msd-irimas.github.io/pages/dl4tsc/**
- Google Collab used for E-EGC Tutorial (clickable link)

Acknowledgements : The providers of the UCR archive, Mésocentre for providing access to their cluster and the Agence National de Recherche for the Delegation Project.