IEEE BIG DATA 2024
December 15-18 • Washington DC, USA

COCALITE : A Hybrid Model COmbining CAtch22 and LITE for Time Series Classification

Authors:
Oumaima Badi[1], Maxime Devanne[1], Ali Ismail-Fawaz[1], Javidan Abdullayev[1], Vincent Lemaire[2], Stefano Berretti[3], Jonathan Weber[1], Germain Forestier[1 4]

Presenting Author:
Oumaima Badi[1]

Université de Haute-Alsace (IRIMAS)[1] ,Orange Innovation[2], University of Florence (MICC)[3] ,Monash University (DSAI)[4]

Table of contents

Introduction

Definition

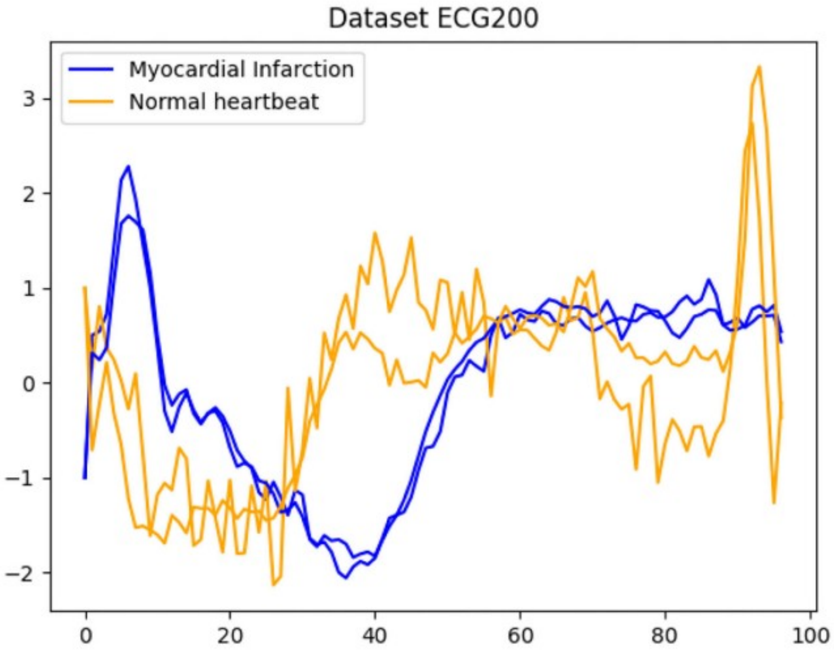
- A univariate time series X of length L: $X=(x_1,x_2,...,x_L)$

Time Series Classification (TSC)

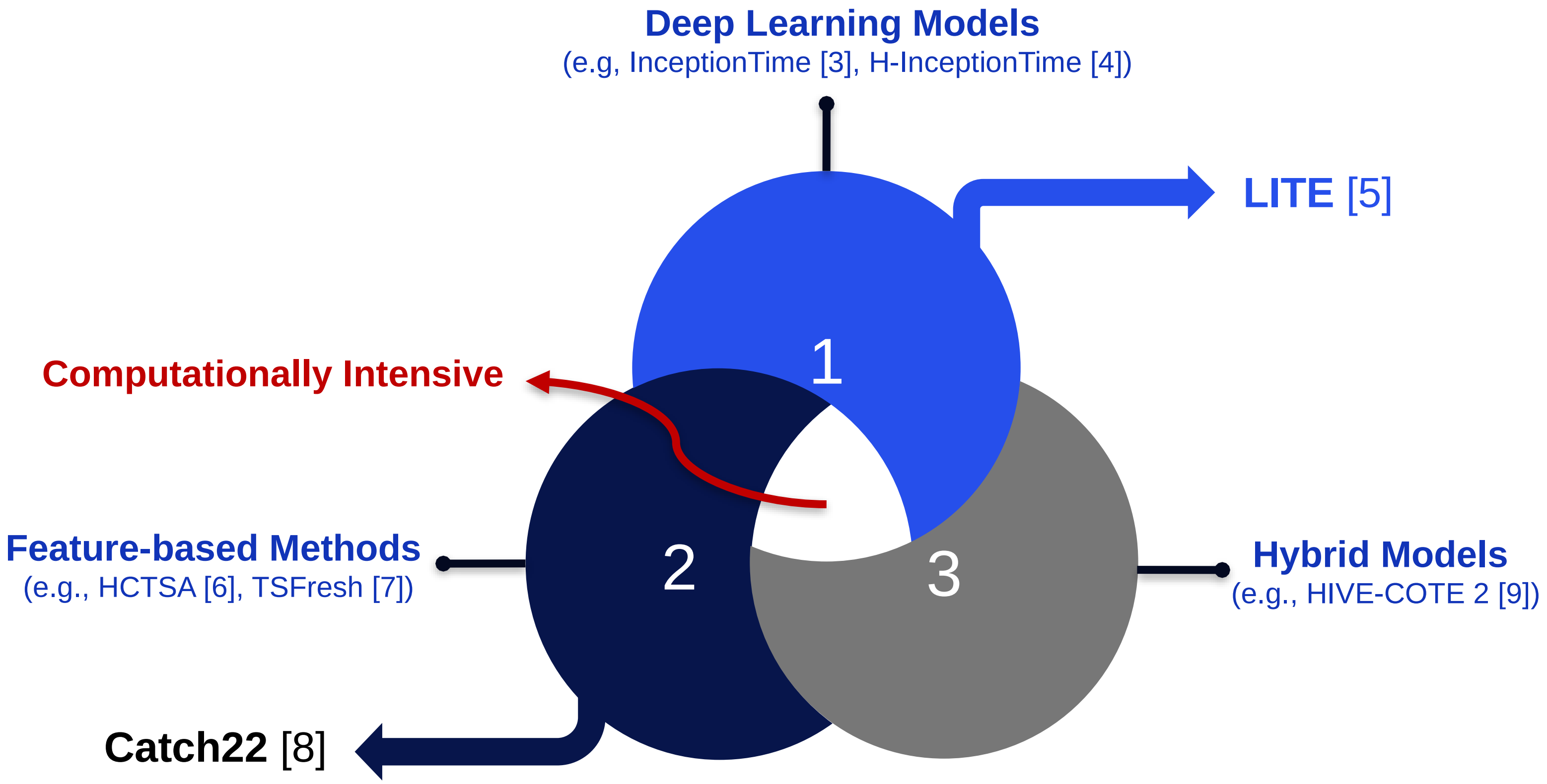
- **Dataset:** $D=\{(X_i,Y_i)\}$, where X_i is a time series and Y_i its label.
- **Goal:** Learn a function $f: X \rightarrow Y$ that maps each time series to its correct category.

Introduction

An illustrative case of time series classification using the ECG200 dataset from the UCR [1] archive, aiming to automatically differentiate between normal heartbeats and myocardial infarction (Adapted from [2]).



Problem Statement



Hypotheses

1. Integrating statistical features into deep learning models during training enhances their ability to extract diverse features.

2. A hybrid ensemble combining models with and without statistical features improves performance by leveraging complementary strengths.

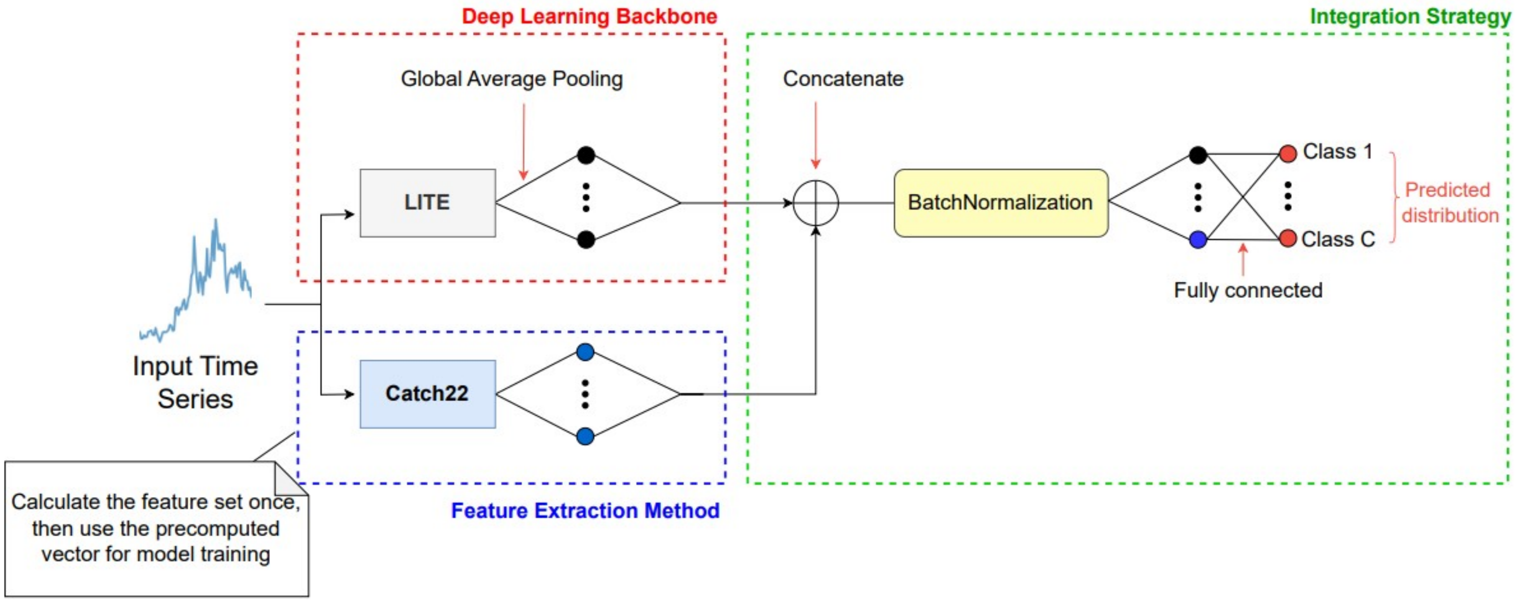
Proposed Solution

Advantages of LITE and Catch22: Enhancing Efficiency and Performance

LITE Model	Catch22 Feature Set
Uses only 2.34% of the parameters of InceptionTime	Near-linear complexity ($O(N^{1.16})$)
2.78x faster training, 2.79x less power than InceptionTime	Only 7.5% less mean accuracy than full HCTSA
Delivers competitive accuracy on TSC tasks	22 features complement LITE’s 32 latent features

Proposed Solution

LITE-Catch22 Architecture: Integrating Deep Learning with Feature Extraction



Proposed Solution

COCALITE Architecture: A Hybrid Deep Ensemble Approach

Components:

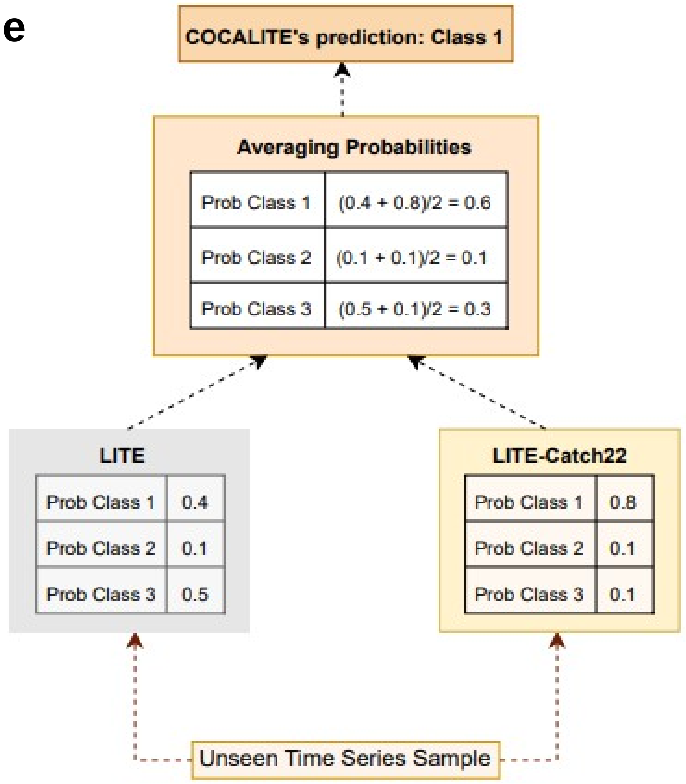
- LITE model
- LITE-Catch22 model

Final Prediction Formula:

$$P_{COCALITE}(c) = (P_{LITE}(c) + P_{LITE\text{-}Catch22}(c))$$

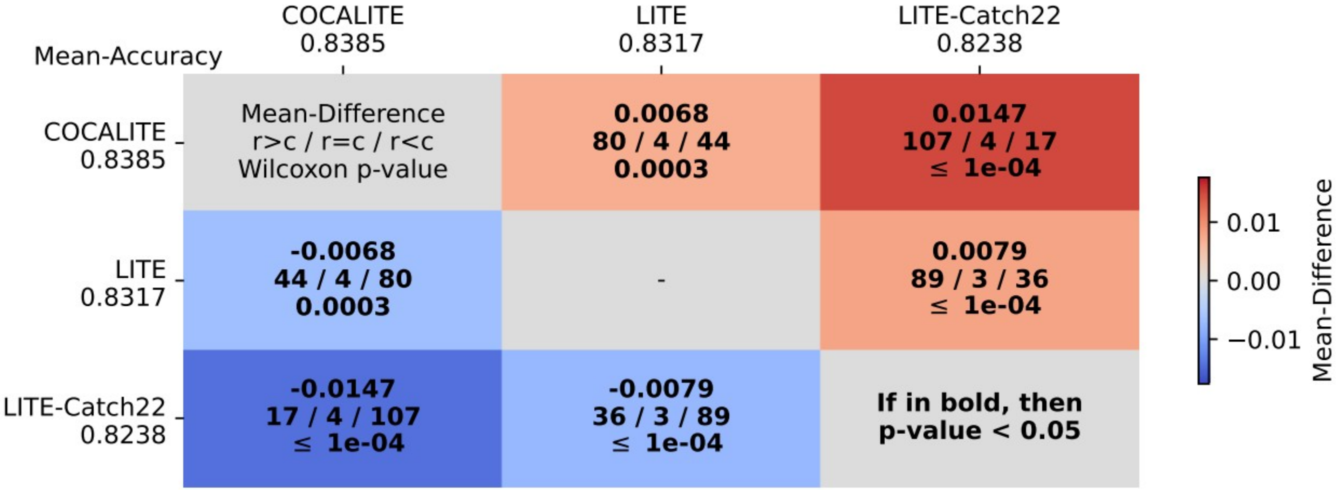
Advantages:

- Combines complementary feature sets.
- Enhances learning beyond individual models.



Ablation Study

The Multi-Comparison Matrix [10] illustrates the one-vs-one performance comparisons between COCALITE and its components.



Experimental Evaluation

Datasets - UCR Time Series Archive

- Evaluation conducted on UCR Archive (128 univariate time series datasets, latest update 2018).

- **Preprocessing steps:** 1- Z-normalization of each dataset.

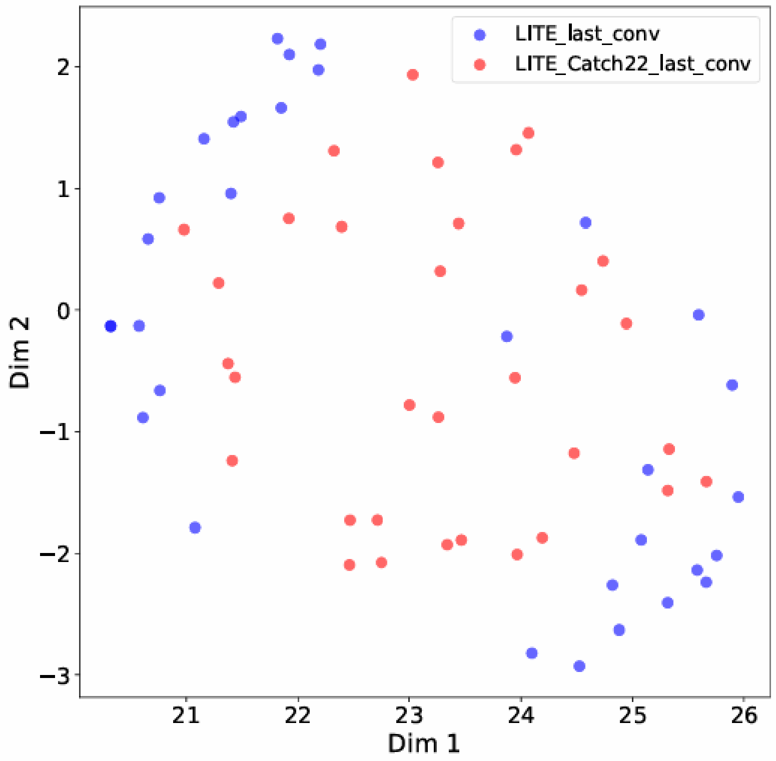
  2- Zero-padding for varying time series lengths.

  3- Linear interpolation for missing values.

Experimental Evaluation

t-SNE Visualization of Last Convolutional Layer Filters: LITE vs. LITE-Catch22

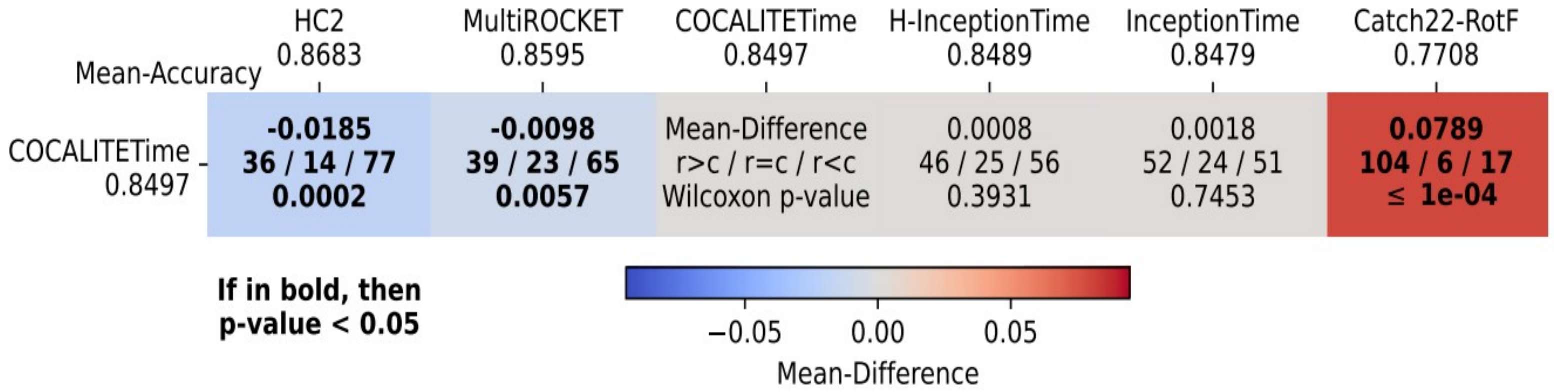
- Distinct clustering of LITE (blue) and LITE-Catch22 (red) filters.
- Incorporating Catch22 results in different filter representations.
- LITE-Catch22 captures a broader range of features.



Experimental Evaluation

Comparison with State-of-the-Art

The Multi-Comparison Matrix applied to show the performance of COCALITETime compared to state-of-the-art approaches.



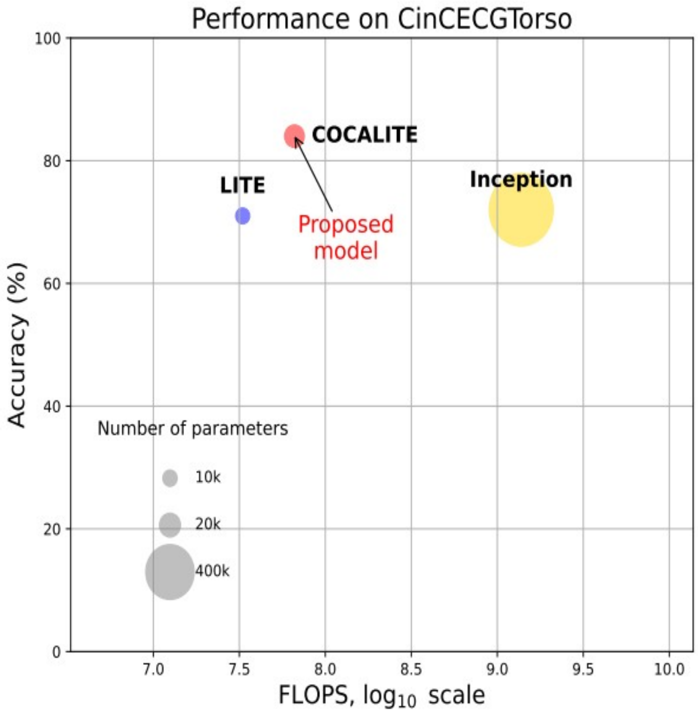
Conclusion

Key Findings

- **COCALITETime** achieves similar performance to **InceptionTime** with only **4.7%** of the parameters.
- **COCALITETime** offers an efficient solution for resource-constrained environments.

Future Directions

- Apply the approach to other deep learning models.
- Experiment with additional feature sets beyond Catch22.



Thank you for your attention!

Please feel free to ask if you have any questions.

Freepik

REFERENCES

[1] H. A. Dau, A. Bagnall, K. Kamgar, et al., “The ucr time series archive,” in IEEE/CAA Journal of Automatica Sinica, vol. 6, 2019, pp. 1293–1305.

[2] D. Bünger et al., "An empirical study of graph-based approaches for semi-supervised time series classification," *Frontiers in Applied Mathematics and Statistics*, vol. 7, 2022. Licensed under CC BY 4.0.

[3] H. I. Fawaz, B. Lucas, G. Forestier, et al., “Inceptiontime: Finding alexnet for time series classification,” Data Mining and Knowledge Discovery, vol. 34, no. 6, pp. 1936–1962, 2020.

[4] A. Ismail-Fawaz, M. Devanne, J. Weber, and G. Forestier, “Deep learning for time series classification using new hand-crafted convolution filters,” in Proceedings of IEEE Big Data 2022, 2022, pp. 1–

[5] A. Ismail-Fawaz, M. Devanne, S. Berretti, J. Weber, and G. Forestier, “Lite: Light inception with boosting techniques for time series classification,” in Proceedings of the 2023 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2023.

[6] B. Fulcher and N. Jones, “Hctsa: A computational framework for automated time-series phenotyping using massive feature extraction,” Cell Systems, vol. 5, no. 5, pp. 527–531, 2017.

[7] M. Christ, N. Braun, J. Neuffer, and A. W. KempaLiehr, “Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package),” Neurocomputing, vol. 307, pp. 72–77, 2018.

[8] C. H. Lubba, S. S. Sethi, P. Knaute, et al., “Catch22: Canonical time-series characteristics,” Data Mining and Knowledge Discovery, vol. 33, no. 6, pp. 1821–1852, 2019.

REFERENCES

[9] M. Middlehurst, J. Large, M. Flynn, J. Lines, and A. Bagnall, “Hc2: A new family of classifiers for time series classification,” ACM Transactions on Knowledge Discovery from Data, vol. 14, no. 5, pp. 1–29, 2022.
[10] A. Ismail-Fawaz, A. Dempster, C. W. Tan, et al., “An approach to multiple comparison benchmark evaluations that is stable under manipulation of the comparate set,” arXiv preprint arXiv:2305.11921, 2023.
[11] A. Ismail-Fawaz, M. Devanne, S. Berretti, J. Weber, and G. Forestier, “Look into the lite in deep learning for time series classification,” arXiv preprint arxiv:2409.02869, 2024.