Taking ROCKE T on an Efficiency Mission: Multivariate Time Series Classification with LightWaveS

Leonardos Pantiskas, Kees Verstoep, Mark Hoogendoorn, Henr i Bal
Vrije Unive rsiteit A msterdam
Vrije Unive rsiteit Amsterdam
Vrije Unive rsiteit Amsterdam
Vrije Unive rsiteit Amsterdam

Abstract—Nowadays, with the rising number of sensor signals in sectors such as healthcare and industry, the problem of multivariate time series classification (MTSC) is getting increasingly relevant and is a prime target for machine and deep learning approaches. Their expanding adoption in real-world environments is causing a shift in focus from the pursuit of ever-higher prediction accuracy with complex models towards practical, deployable solutions that balance accuracy and parameters such as prediction speed. An MTSC model that has attracted attention recently is ROCKE T, based on random convolutional kernels, both because of its very fast training process and its state-of-the-art accuracy. However, the large number of features it utilizes may be detrimental to inference time. Examining its theoretical background and limitations enables us to address potential drawbacks and present LightWaveS: a framework for accurate MTSC, which is fast both during training and inference. We show that LightWaveS achieves accuracy comparable to recent MTSC models and speedup ranging from 9x to 53x compared to ROCKE T during inference on an edge device, on datasets with comparable accuracy.

Index Terms—time series, classification, multivariate, wavelet, scattering, feature selection, edge intelligence

I. INTRODUCTION

Time series classification is the task of characterizing a series of values observed in sequential moments in time as belonging to one of two or more categories, or classes. Nowadays, problems are increasingly described by more than one channel of information [1], turning the task into multivariate time series classification (MTSC). Some factors contributing to that are the development of smaller and cheaper sensors for the measurement of various quantities and the general advancement of Internet of Things [2].

Although the improvement of the prediction accuracy of MTSC models has been a prominent goal of research, the rise of edge computing and deployment of models in such real-world environments makes it necessary to also consider the conditions and resources of the computational nodes on which those models will be executed [3]. This change in perspective makes it important to include, apart from accuracy, criteria such as prediction speed or throughput in the evaluation of a model’s suitability for a given task.

In the recent evaluation of advances in MTSC [1], one solution that shows good performance is ROCKE T [4], both in terms of accuracy and training time. ROCKE T and its evolution, MINIROCKET [5] utilize random convolutions to transform the time series channels and then extract features that are used with a linear classifier. Although ROCKE T has been originally developed for univariate time series, its multivariate extension in the sktime [6] repository achieves the impressive performance mentioned above.

The main issue of (MINI)ROCKET that we try to address in our work is that the large number of features and the random combination of channels, although beneficial to accuracy, can be highly redundant, leading to unnecessarily high transformation and inference times. Based on this, we propose LightWaveS, a framework for fast transformation of multivariate time series based on convolutional kernels, wavelet scattering, and feature selection, for lightweight and accurate classification with linear classifiers. Our solution aims to keep the successful aspects of the ROCKE T model family, such as the short transformation time, the arbitrary convolutional kernel approach, and the few descriptive features per kernel. On top of that, LightWaveS adds well-studied wavelet theory, multi-node distribution during training and smart feature selection to address the issue identified above. The trade-off that we propose compared to (MINI)ROCKET is clear: we take advantage of additional computational resources during training time, especially for larger datasets, to keep its duration short and significantly speed up inference on resource-constrained devices. Our contributions with LightWaveS are:

- We introduce the usage of minimal wavelet scattering based on arbitrary kernels.
- We achieve accuracy comparable to state-of-the-art in the majority of the UEA datasets [7], using only a fraction of the number of features used by (MINI)ROCKET.
- We achieve inference speedups on a representative edge device ranging from 9x to 53x compared to ROCKE T.
- We achieve reduction of the channels required for inference ranging from 7% to 86%.

With this work, we take an efficiency-centric approach, focusing on the practicality and the inference speed of a model.
that may be deployed on edge devices, without necessarily trying to surpass the state-of-the-art in terms of accuracy.

II. RELATED WORK
A. Multivariate time series classification

Due to the recent rise in popularity of the deep learning field, there is a multitude of models that can be easily adapted to incorporate the additional dimension of MTSC [1]. The majority of those deep learning models take a significant amount of time and memory to train even on GPU nodes, ranging from hours to even days, depending on the dataset size [1]. In contrast, our method takes less than 20 minutes to transform, train and test all 30 UEA datasets on a CPU node with a linear classifier. The MPI data-parallel distribution of our framework enables even faster processing, especially for larger datasets.

Another common approach is feature extraction and selection models, such as WEASEL+MUSE [8], which extracts features based on a bag-of-patterns approach and selects the most useful ones based on a $\chi^2$ test. LightWaveS extracts statistical features from the coefficients of wavelet scattering, which gives the model enough complexity to accommodate difficult datasets, where predefined statistical features on the raw time series values may not be descriptive enough.

B. Wavelets

Wavelets are localized waveforms and are a well-studied method in signal processing that has been used extensively in the analysis of time series of all types of problems, ranging from healthcare to audio analysis [9]. A seminal work is [10], where the concept of a wavelet scattering network using a Morlet wavelet is introduced, in combination with linear and support vector machine classifiers. This method was constructed to be invariant to translations of the input and stable to small deformations.

LightWaveS aims to combine the strong points of these works under a single generalized framework, with a focus on efficiency. We aim to bridge the gap between ROCKET and the wavelet theory, and we progress to the next logical step of wavelet scattering. We keep this approach lightweight, both in depth and paths of the scattering, so we can apply it to time series channels on a massive scale in a very short time. The arbitrary base set of wavelets can potentially be extended based on expert opinion, backed by the solid theory behind wavelets and their applications, making LightWaveS a suitable platform for experimentation on solutions for MTSC problems. Finally, the hierarchical feature filtering leads to the most relevant output features of the scattering coefficients being selected.

III. PROPOSED FRAMEWORK
A. Preliminaries

1) (MINI)ROCKET fundamentals: The ROCKET model family is primarily based on the convolution of random kernels with either single input channels or random combinations of them. In the minimally random variant, MINIROCKET, the kernel weights are selected from an empirically chosen subset of 84 kernels of length 9 with weights in $\{-2,-1\}$ but other lengths, different values or weights drawn from $\sim \mathcal{N}(0,1)$ are equally effective [5]. The only thing that is important is that the kernel weights have sum 0. One or more features are extracted from the convolution outputs and are used for the final classification.

2) Wavelet Scattering: Wavelet scattering is the process of applying wavelet transforms in a cascading manner [10], combined with non-linearities and pooling. The wavelet transform is a method used to approximate a signal using a set of wavelets that originate from a “mother” wavelet $\Psi(t)$, scaled by $s$ and shifted by $b$ [11]. Each such wavelet can be described as:

$$\psi_{s,b}(t) = \frac{1}{\sqrt{s}}\Psi\left(\frac{t-b}{s}\right)$$

The connection between convolutional networks and the scattering architecture has been thoroughly explored in [12], and we can intuitively relate these wavelets to the convolution kernels discussed above. In addition, the wavelet set created from a mother wavelet is parallel to the way that MINIROCKET has a fixed set of kernel weights, for which different paddings and dilations (scales) are randomly selected, generating the child kernels.

In the wavelet scattering transform, on each level $\lambda$, the previous result is convoluted with each wavelet $\psi_{\lambda_n}$ (kernel) and a complex modulus operator is applied before propagating the result to the next level, such that:

$$U[\lambda] = |U[\lambda - 1]*\psi_{\lambda_n}|$$

and the scattering coefficients that result from each level are

$$S[\lambda] = U[\lambda]*\phi$$

where $\phi$ is an averaging kernel. Since on every level of the scattering transform there can be multiple candidate wavelets (kernels), there is a geometric progression of potential paths, as can be seen in the graphic representation of the process in Fig. 1.

![2-level wavelet scattering](image)

In the above intuitive description, we connected the concepts of the ROCKET models with the wavelet theory. In MINIROCKET specifically, the prototype kernels have zero mean, which corresponds to a desirable constraint of mother wavelets [11]. Under this new context, we can reconsider MINIROCKET as a classifier based on convolutions
with random child wavelets of an arbitrary set of mother wavelets. In addition, although not being wavelet scattering, MINIROCKET can be intuitively mapped on the first level, with its kernel set equivalent to the first-level set, \( \{ \Psi_1, ..., \Psi_n \} \). However, MINIROCKET extracts the features before the application of any modulus operator, so it does not satisfy the rest of the wavelet scattering transform requirements.

**B. Algorithm**

Based on the above observations, we improve the approach and reach the crux of LightWaveS: Lightweight Wavelet Scattering based on random wavelets. The term lightweight refers both to the system optimizations of the framework that make it fast, such as the distribution, as well as the fact that only a reduced set of wavelet scattering paths are computed up to two levels (paths shown in bold in Fig. 1). Although it is established that not all paths need to be considered in a scattering network [10], since we are aiming for fast training and inference, we take this notion to its limit. We compute all coefficients for the given kernels and dilations for the first level, but we consider only one path per first-level output for the second level. In this way, we limit significantly the memory and computation time required for the extracted features, while accepting the trade-off of losing some descriptive coefficients. The intuition behind the path selection, as well as the detailed steps of the feature extraction and selection, are described in more detail in the longer version of this work [13].

**IV. EXPERIMENTS**

**A. Datasets**

We select as benchmark the UEA collection of multivariate datasets [7], excluding InsectWingbeat, since due to its large size it presented issues when training ROCKET. The datasets are described in detail in [1]. In addition, we prepare and use five machinery fault related datasets: MAFAULDA (MF), from Machinery Fault Database [14] and TURBOFAN (TF) [15], an engine degradation simulation dataset collection.

**B. Experimental setup**

All training experiments were run on the DAS-5 infrastructure [16], on nodes with dual 8-core 2.4 GHz (Intel Haswell E5-2630-v3) CPUs and 64 GB of RAM. The inference experiments are executed on a Jetson Xavier board which has an 8-core ARM CPU. Both (MINI)ROCKET and LightWaveS were set to use all 8 cores during inference. We ran ROCKET and MINIROCKET using the default number of features (20 and 10 thousand respectively). As for LightWaveS, we present three variants of the model, termed L1, L2, and L1L2. These versions refer to keeping the features only from the scattering level 1, level 2, or both, although we consider the L1L2 version as the default. Our hyperparameter search during experimentation was limited to the initial and final number of selected features, as well as alternative options for the second-level scattering paths, as described in [13]. Gradually increasing the number of features, we found that 500 features showed good balance between training time, inference speed and accuracy in the experimentation subset. Following [4], we use a Ridge regression classifier for all methods. We repeat the accuracy and timing experiments 30 and 100 times respectively and report average values for robustness. We use a critical difference diagram to present the results, a popular method of comparing classifier performance across multiple datasets [1]. The code of LightWaveS, as well as detailed metrics and experimentation scripts, are made available at https://github.com/lpphd/lightwaves to enable reproducibility and facilitate further exploration on the topics of this work.

**C. Accuracy results**

We present the performance of LightWaveS in terms of accuracy in comparison with (MINI)ROCKET, as well as other recent solutions, namely M)OSS-CNN [17], WEASEL+MUSE [8], TapNet [18] and Catch22 [19]. Apart from LightWaveS and (MINI)ROCKET, the rest of the accuracy metrics are taken from the repositories of [17] and [20].

We can see the results in Fig. 2. Although lower in rank, LightWaveS belongs in the same statistical group with (MINI)ROCKET, which have the best accuracy. In addition, it is among the ranks of more complex DL methods, such as TapNet and MOS-CNN. Out of the three variants, LightWaveS-L2 performs the worst, without the benefit of faster execution that L1 has due to the fewer convolutions.

**D. Inference speedup results**

We can focus on ROCKET for a more detailed comparison, and also include the 5 additional datasets. Since the aim of LightWaveS is to approach its state-of-the-art accuracy with fewer features, not necessarily surpass it, we place the LightWaveS results for all datasets in four accuracy bins compared to ROCKET: one for the cases where LightWaveS achieves higher or equal accuracy, and three bins for lower accuracy, with difference less than 0.05, between 0.05 and 0.1 and more than 0.1 respectively. In Fig. 3 (left) we see that for the majority of the datasets the accuracy stays in the first 3 categories for all variants apart from L2, and the datasets with large accuracy deviation are few.
3 (right) the accuracy - speedup plots for all LightWaveS variants compared to ROCKET for the 5 machinery fault datasets. We choose these since they are representative use cases and correspond to an immediately realistic scenario of edge deployment of a model in an industrial IoT environment. However, a similar relation holds for all other datasets.

This figure clearly frames the trade-off that we propose: significant inference speedup at the cost of limited reduction in accuracy (in cases where the latter is not also improved).

E. Channel reduction results

Due to its feature generation and selection pipeline, LightWaveS can filter the input channels required for inference to a subset of the originals. Out of the 24 datasets where comparable accuracy with ROCKET is achieved, 11 were reduced, with the channel reduction ranging from 7% to 86% and the larger datasets benefiting more. The channel reduction effect has advantages such as providing insights into which channels contain useful information for the problem, leading to knowledge extraction. Moreover, on edge devices, where resources are valuable, it can free up incoming signal channels and reduce communication overhead.

F. Resource efficiency results

Another important aspect in edge intelligence applications is the number of operations required for inference, which directly affects the energy required, as well as the performance of other applications that potentially share the edge device resources. We can quantify the computational efficiency of LightWaveS over ROCKET by comparing the number of convolutions and consequently multiply–accumulate (MAC) operations required for inference, based on the theoretical analysis of the algorithms. For the datasets in the first two accuracy bins of the L1L2 variant, LightWaveS requires approximately 16.5x to 40.5x fewer MAC operations for inference.

V. CONCLUSION

Summarizing, LightWaveS utilizes minimal wavelet scattering transformation and hierarchical feature selection and achieves comparable to state-of-the-art accuracy on the datasets tested, with training time similar to (MINI)ROCKET or shorter and inference time 9x-53x shorter on an edge device. It also achieves significant computational efficiency and input channel reduction, making it suitable for deployment on edge intelligence applications. Promising future research includes a more informed selection of scattering paths and extracted features, to further reduce the number of required convolutions and make the framework even more efficient.

REFERENCES

[1] A. P. Ruiz, M. Flynn, J. Large, M. Middlehurst, and A. Bagnall, “The great multivariate time series classification bake off: A review and experimental evaluation of recent algorithmic advances,” Data Mining and Knowledge Discovery, vol. 35, no. 2, pp. 401–449, 2021.
[2] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, “Deep learning for IoT big data and streaming analytics: A survey,” IEEE Communications Surveys and Tutorials, vol. 20, no. 4, pp. 2923–2960, 2018.
[3] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan, and X. Chen, “Convergence of Edge Computing and Deep Learning: A Comprehensive Survey,” IEEE Communications Surveys & Tutorials, vol. 22, no. 2, pp. 869–904, 2020.
[4] A. Dempster, P. Petitjean, and G. I. Webb, “ROCKET: Exceptionally fast and accurate time series classification using random convolutional kernels,” arXiv:1910.13051 [cs, stat], 2019.
[5] A. Dempster, D. F. Schmidt, and G. I. Webb, “MINIROCKET: A Very Fast (Almost) Deterministic Transform for Time Series Classification,” arXiv:2012.08791 [cs, stat], 2020.
[6] M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. J. Kirily, “Sktime: A Unified Interface for Machine Learning with Time Series,” arXiv:1909.07872 [cs, stat], 2019.
[7] A. Bagnall, H. A. Dau, J. Lines, M. Flynn, J. Large, A. Bosstrom, P. Southam, and E. Keogh, “The UEA multivariate time series classification archive, 2018,” arXiv:1811.00075 [cs, stat], 2018.
[8] P. Schäfer and U. Leser, “Multivariate Time Series Classification with WEASEL+MUSE,” arXiv:1711.11345 [cs], 2018.
[9] S. Mallat, A wavelet tour of signal processing. Elsevier, 1999.
[10] J. Bruna and S. Mallat, “Invariant Scattering Convolution Networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1872–1886, 2013.
[11] R. K. Young, Wavelet theory and its applications. Springer Science & Business Media, 2012, vol. 189.
[12] S. Mallat, “Understanding deep convolutional networks,” Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 374, no. 2065, p. 20150203, 2016.
[13] L. Pantisakas, K. Verstoep, M. Hoogendoorn, and H. Bal, “Taking ROCKET on an Efficiency Mission: Multivariate Time Series Classification with LightWaveS,” arXiv:2204.01379 [cs], 2022.
[14] “Machinery Fault Database [Online],” http://ww02.smtp.utfri.br/ofshore/mfs/page_01.html.
[15] A. Satena and K. Goebel, “Turbofan engine degradation simulation data set,” http://ti.arc.nasa.gov/project/prognostic-data-repository, 2008.
[16] H. Bal, D. Epema, C. de Laat, R. van Nieuwpoort, J. Romein, F. Seinstra, C. Snoek, and H. Wijshoff, “A medium-scale distributed system for computer science research: Infrastructure for the long term,” Computer, vol. 49, no. 5, pp. 54–63, 2016.
[17] W. Tang, G. Long, L. Liu, T. Zhou, J. Jiang, and M. Blumenstein, “Re-thinking 1D-CNN for Time Series Classification: A Stronger Baseline,” arXiv:2002.10661 [cs, stat], 2021.
[18] X. Zhang, Y. Gao, J. Lin, and C.-T. Lu, “Tapnet: Multivariate time series classfication with attentional prototypical network,” Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 04, pp. 6845–6852, 2020.
[19] C. H. Lubba, S. S. Sethi, P. Knaute, S. R. Schultz, B. D. Fulcher, and N. S. Jones, “catch22: Canonical time-series characteristics,” Data Mining and Knowledge Discovery, vol. 33, no. 6, pp. 1821–1852, 2019.
[20] B. Dharvial, T. le Nguyen, S. Gsponer, and G. Hrim, “An Examination of the State-of-the-Art for Multivariate Time Series Classification,” in 2020 International Conference on Data Mining Workshops (ICDMW). IEEE, 2020, pp. 243–250.