Multisource Time Series Classification Models for Urban Tree Species

Victor Thiebaut[1], Ali Ismail-Fawaz[1][0000-0001-5385-3339], Cyril Meyer[1][0000-0001-7262-999X], Romain Wenger[2][0000-0001-7557-6862], Anne Puissant[2][0000-0002-3240-9244], and Germain Forestier[1,3][0000-0002-4960-7554]

[1] IRIMAS, Université de Haute-Alsace, Mulhouse, France
{victor.thiebaut,ali.ismail-fawaz,cyril.meyer,germain.forestier}@uha.fr
[2] LIVE, Université de Strasbourg, Strasbourg, France
{romain.wenger,anne.puissant}@live-cnrs.unistra.fr
[3] DSAI, Monash University, Melbourne, Australia

Abstract. Multisource time series classification is a critical computational challenge spanning diverse applications, from clinical health monitoring and human activity recognition to Earth observation. However, effectively integrating asynchronous, heterogeneous data streams remains architecturally complex due to varying sampling rates, source-specific noise, and distinct spatial-temporal resolutions. This paper presents a comprehensive comparative analysis of fusion strategies for a multisource time series dataset of Satellite Image Time Series (SITS) representing urban tree species. We systematically assess the capabilities of existing models through diverse feature extraction techniques, which we adapt in this work to handle multisource data. To do this, we employ a broad spectrum of deep and non-deep learning methods, specifically LITE, TempCNN, MultiRocket, K-Nearest Neighbors across different distance metrics, Quant, and Catch22. Crucially, the classifiers coupled with these methods are systematically selected and tailored to match the specific nature of the intermediate representations generated by each preceding extraction block. By exploring these various architectural combinations and fusion concepts, this work highlights the specific contributions, strengths, and limitations of each method.

Keywords: Time series classification · Multisource time series · Satellite Image Time Series

1 Introduction

The classification of time series data represents a foundational challenge within the broader landscape of machine learning, underpinning critical applications that range from human activity recognition [16], physiological monitoring [13] to industrial anomaly detection [22] and the analysis of environmental satellite imagery [24,1]. The main focus of recent research in Time Series Classification (TSC) has predominantly centered on univariate sequences or homogeneous multivariate data [2].

Recent advances in remote sensing, however, have fundamentally shifted the nature of available data. Earth observation systems now routinely generate multisource Satellite Image Time Series (SITS), providing asynchronous, heterogeneous data streams. At its core, a SITS is formed by stacking multiple satellite images acquired over the same geographic area across different dates. When extracted at a pixel-wise level, this data structure yields a discrete temporal trajectory that tracks the physical, chemical, or structural evolution of the Earth's surface over time. So many satellites capture or monitor our planet, such as Sentinel satellites (S1, S2) from the European Spatial Agency (ESA) or PlanetScope (PS). While combining multiple sensors, such as high-spectral-resolution data and high-spatial-resolution imagery, offers a richer representation of complex phenomena like urban phenology, it introduces significant architectural challenges in time series machine learning research.

Addressing these challenges requires moving beyond traditional homogeneous models to develop robust multisource fusion concepts. The critical question lies in determining at which stage of the modeling process these diverse sources should be integrated: (1) at the feature level or (2) at the decision level.

This paper presents a comprehensive comparative analysis of fusion strategies applied to a multisource SITS dataset for classifying urban tree species [1]. We systematically assess the capabilities of existing state-of-the-art models by adapting diverse feature extraction techniques to handle heterogeneous data streams. By evaluating a broad spectrum of deep learning (LITE [10], TempCNN [17]) and non-deep learning methods (MultiRocket [21], K-Nearest Neighbors, Quant [6], and Catch22 [14]) across different fusion paradigms, this work highlights the specific contributions, strengths, and limitations of each model. Every methods have been developed in a way to support multiple sources. Ultimately, we aim to provide clear guidelines on effectively coupling classifiers with intermediate representations to maximize multisource classification performance. The code is available here : https://github.com/MSD-IRIMAS/multisource_ts_models.

2 Related Work

The field of Time Series Classification (TSC) has evolved significantly from traditional distance-based algorithms to highly optimized, computationally efficient machine learning models. Historically, K-Nearest Neighbors coupled with alignment-based similarity metrics like Dynamic Time Warping (DTW) [18] and Move-Split-Merge (MSM) [20] served as a standard by effectively handling temporal shifts in sequential data. However, the computational complexity of these methods stimulated the development of alternative concepts.

Feature-based approaches, such as Catch22 [14], demonstrated that distilling time series into a highly optimized set of statistical characteristics could rival more complex models on specific tasks with a fraction of the computational overhead. Similarly, interval-based methods like Quant [6] provided minimalist frameworks for rapid feature extraction using quantile representations. More recently, convolution-based algorithms, notably the ROCKET family and its successor, MultiRocket [21] have established new state-of-the-art baselines by applying massive sets of random convolutional kernels to capture diverse temporal patterns with exceptional speed and accuracy [19]. Building upon this idea, Hydra [5] introduced a highly efficient dictionary-based approach that utilizes competing convolutional kernels. By bridging conventional dictionary pattern matching with the speed of ROCKET-style transformations, and frequently deployed as a combined models (Hydra+MultiRocket), these models concatenate diverse feature spaces to allow simple linear classifiers to achieve unparalleled predictive performance on complex temporal data.

In the domain of remote sensing and Earth observation, deep learning has fundamentally transformed pixel-wise classification tasks. Unlike standard univariate TSC, Satellite Image Time Series (SITS) data inherently embeds complex, long-term environmental and phenological dependencies. Specialized architectures like TempCNN [17] were explicitly designed to capture these seasonal dynamics in high-dimensional SITS through deep, stacked 1D convolutions.

The proliferation of diverse Earth observation sensors, such as the high-spectral-resolution Sentinel-2 and the high-spatial-resolution PlanetScope, has shifted research focus toward multisource data integration [24,1]. Handling asynchronous, heterogeneous data streams introduces significant architectural challenges, primarily regarding the optimal stage for source fusion.

Literature broadly categorizes these approaches into two classes: early (feature-level) and late (decision-level) fusion[7]. Feature-level fusion seeks to build a joint representation by concatenating intermediate embeddings before classification, theoretically allowing a model to learn cross-source synergies early in the pipeline. Conversely, decision-level fusion isolates the feature extraction process, aggregating the independent probabilistic outputs of source-specific classifiers at the very end, it can also be represented by a majority vote. While early fusion models have shown success in homogeneous multivariate settings, asynchronous multisource datasets often introduce source-specific noise and varying activation scales that can destabilize early joint representations. Systematic comparative studies evaluating these fusions across both deep and non-deep state-of-the-art TSC models remain scarce, highlighting the architectural gap this work addresses in the context of urban forestry mapping.

3 Dataset

This study utilizes an open-source benchmark dataset specifically designed for the SITS classification of urban tree species [1]. The dataset in Fig. 1 provides temporally aligned, preprocessed surface reflectance values for the city of Strasbourg, France, structured into a ready-to-use format with train, test and validation splits.

The dataset contains 45 084 public trees, each tree is represented by two multivarite time series source, where each source is originated from a different satellite: PlanetScope (PS) and Sentinel-2 (S2). The ground truth labels and geolocations are derived from a municipal database, restricted to the 20 most

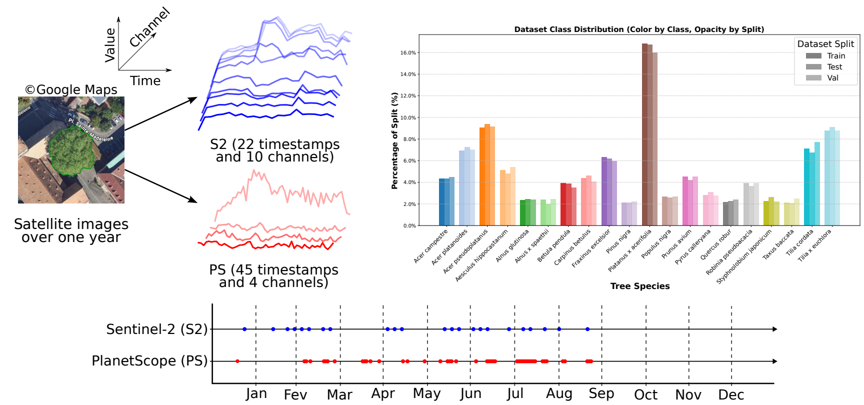

Fig. 1. Urban Tree Species Time-Series Classification Dataset: Histograms of tree-species class frequencies and timestamp distributions across the dataset.

frequently occurring species. The dataset includes two different sources, both of which are optical sources:

1. Sentinel-2 (S2): This source provides rich spectral information, capturing 10 distinct spectral bands at a spatial resolution of 10 to 20 meters. The S2 source incorporates 22 timestamps.
2. PlanetScope (PS): This source delivers higher spatial and temporal granularity, providing 4 spectral bands at a 3.125-meter spatial resolution. The PS source incorporates 45 timestamps.

For both sources, there is a cloud cover restriction: timestamps were kept only if there is less than 10% clouds over the tile.

4 Adapting Existing TSC Models for Multisource Data

The primary objective of this study is to evaluate existing deep learning and non-deep learning models from the literature on the multisource SITS urban tree species classification dataset. In order to gain a deeper understanding of this issue, we systematically evaluate both deep and non-deep learning methods using various strategies. In this section, we detail several models based on distinct approaches to multivariate time series classification.

4.1 Multi-Deep

Multi-Deep classifiers encompasses our deep learning-based strategies for multisource time series classification. We investigate two distinct methods in this section: an approach based on intermediate feature-level fusion(FL), and an approach relying on a late probabilities aggregation: the Decision Level method(DL).

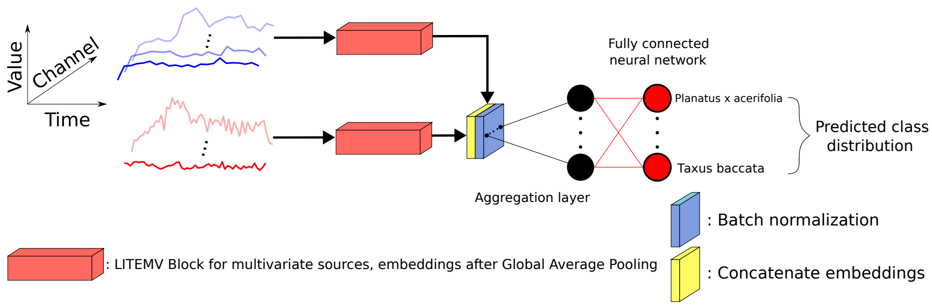

Fig. 2. Multi-LITE Feature Level strategy taking as input two different sources, the S2 source and the PS source.

Multi-Deep Feature Level In this strategy (Figure 2), a dedicated LITE [11] or LITEMV [10] (depending on whether the source is univariate or multivariate) block is applied independently to each source. This decoupled extraction strategy allows the model to learn source-specific temporal patterns without early cross-source interference. The resulting embeddings extracted from each block are then concatenated and a batch normalization is applied. Applying a batch normalization immediately after concatenation standardizes the joint representation, preventing any single source with larger activations from dominating the subsequent learning phase. Furthermore, it stabilizes the internal covariate shift caused by the decoupled branches converging at different rates. This feature-level fusion generates a comprehensive joint representation that captures the complementary information across all sources, which is subsequently fed into a fully-connected classifier layer classifier explicitly tailored to map these combined embeddings to the final classes.

Compared to many other time series classification models, such as InceptionTime [12] or ResNet [23], LITE offers the advantage of being significantly faster while achieving comparable or even superior performance. This efficiency is primarily due to its parameter-efficient design, making it highly suitable for feature-level multi-source frameworks where computational cost scales with the number of inputs.

Similarly, we adapt this strategy by substituting the LITE modules with TempCNN modules [17]. While LITE has been made and excels for high-frequency kinematic or physiological sensors, TempCNN was specifically designed to handle the distinct characteristics of Satellite Image Time Series (SITS). This alternative is motivated by TempCNN's capacity to extract long-term temporal dependencies, such as seasonal and phenological variations, using its deep architecture of stacked 1D convolutions.

Multi-Deep Decision Level In this strategy, an end-to-end LITE or LITEMV convolution-based deep classifier pipeline is deployed independently for each source. However, instead of applying the argmax function directly to the out-

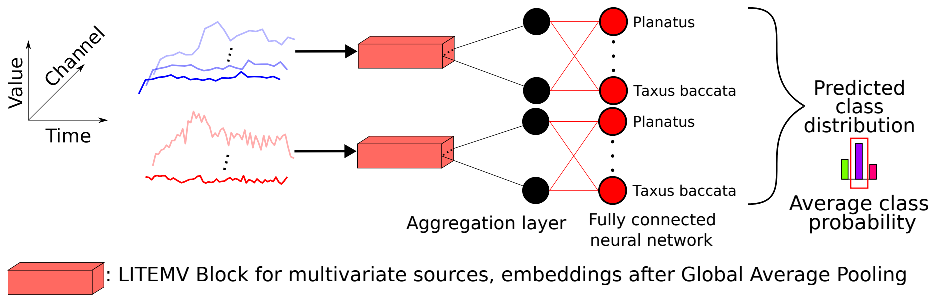

Fig. 3. Multi-LITE Decision Level strategy.

put of each individual model, the probability distributions from all sources are aggregated through an average, upon which the final argmax is computed to determine the predicted class.

The primary advantage of this decision level strategy is its robustness to source-specific noise. By not applying a feature fusion, instead aggregating the probabilities at the decision stage, each network is exclusively optimized for its respective source, preventing noisy or dominant sources from skewing the feature representations early in the learning process. Finally, to ensure a comprehensive comparative analysis, Multi-Deep Decision Level is similarly adapted by substituting LITE modules with TempCNN ones.

4.2 Multi K-Nearest Neighbors

K-Nearest Neighbors (KNN) classification is the most common baseline for time series classification. KNN on time series data utilizes specific similarity measures defined in the literature for sequential data. We use both Dynamic Time Warping (DTW) [18] and Move-Split-Merge (MSM) [20] similarity measures for the multisource KNN classifier.

On one hand, DTW measures the similarity between two time series by first re-aligning both series on the temporal axis before applying a euclidean distance. On the other hand, MSM measures the similarity between two time series by using a set of operations to transform one series into another. Each operation has an associated cost, these operations are: move, split and merge. According to many comprehensive evaluation ([9] and [8]), the most effective metric for clustering is the MSM metric.

The Multi K-Nearest Neighbors model (Figure 4) contains one KNN classifier for each source of data. Each KNN classifier is setup to return the three neighbors (from the training set) of an unseen time series samples ($K = 3$). The prediction is therefore the three class labels of those neighbors. The model then process to a majority vote for class prediction. We choose $K = 3$ in order to avoid as much as possible the eventuality of a draw between multiple classes.

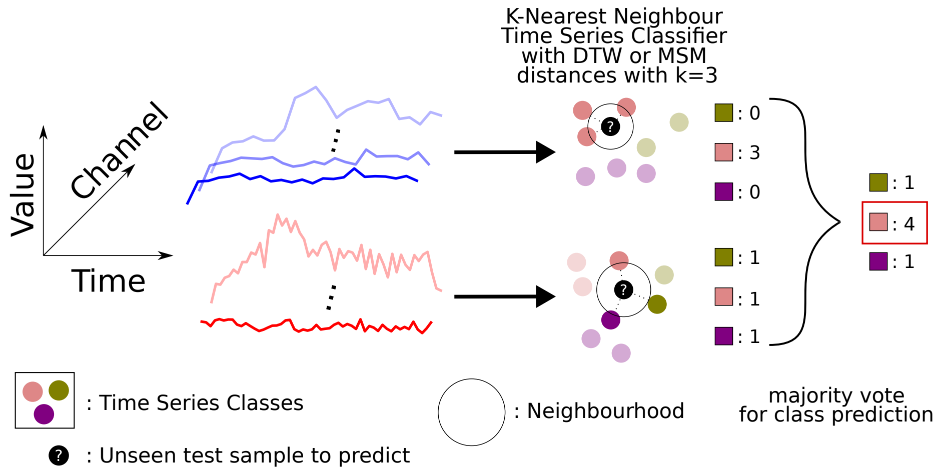

Fig. 4. Archticture of Multi-KNN

4.3 Multi MultiRocket

MultiRocket [21] is a state-of-the-art machine learning convolution-based algorithm, it has been designed specifically for time series classification. Building on its predecessors, ROCKET [3] and MiniRocket [4], it works by transforming time series data using a large set of random convolutional kernels to extract relevant features. Moreover, MultiRocket is a light and fast approach, making it really relevant to achieve multisource time series classification. We propose two different ways of using the MultiRocket transformation.

Multi MultiRocket Feature Level This strategy (Figure 5) utilizes one MultiRocket transformation block for each source, all transformed features are concatenated into one vector. Due to the massive number of features with completely different numerical scales, we use a Standard Scaler to clean up the scale followed by a Ridge classifier. Indeed, the Ridge classifier works by penalizing large model weights. If the features are not scaled, the model is forced to assign huge weights to small-scale features simply to make them useful, which the Ridge penalty will unfairly suppress.

Multi MultiRocket Decision Level This strategy (Figure 6) uses in a similar way one MultiRocket transformation block for each source. However, differently from Multi MultiRocket Feature Level, each transformation block is followed independently by a standard scalar and a Logistic Regression model. This classifier does not handle huge amount of data as good as the Ridge Classifier. In this case, the number of features is smaller than in the "Multi MultiRocket Feature Level" strategy. Moreover, the Logistic Regressor predicts probabilities, allowing the ensemble of all pre-trained Logistic Regression models.

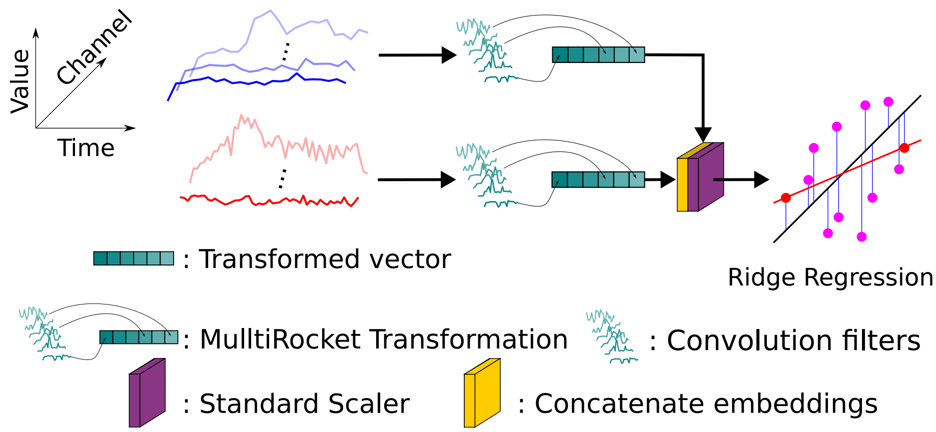

Fig. 5. The Multi MultiRocket Feature Level Strategy.

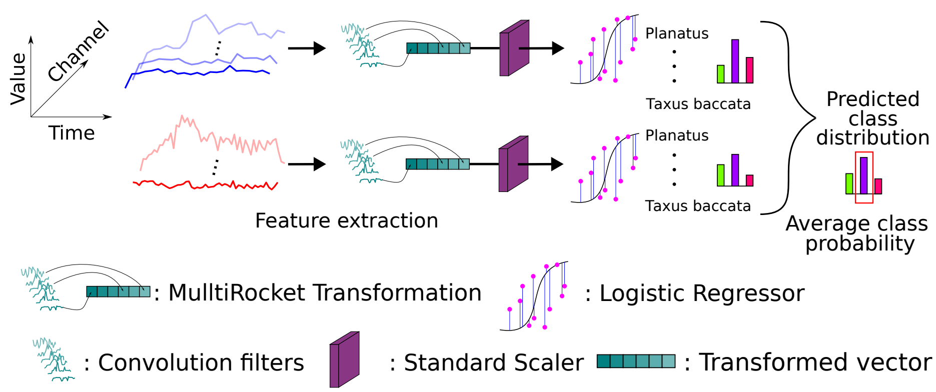

Fig. 6. The Multi MultiRocket Decision Level Strategy.

4.4 MultiQuant

Quant [6] is a minimalist and efficient classifier which rapidly extracts quantile features from fixed intervals within a sequence of data, allowing standard "off-the-shelf" classifiers to achieve state-of-the-art accuracy with exceptionally low computational overhead. Quant is particularly fast, making it suitable for multisource TSC tasks.

MultiQuant Feature Level This strategy, similarly to the strategy "Multi MultiRocket Feature Level", utilizes a Quant transformation for each source and concatenates the output features. These features are then fed to train an Extra Trees Classifier as in the original proposition of the model.

MultiQuant Decision Level This strategy, similarly to the strategy "Multi MultiRocket Decision Level", utilizes a Quant transformation for each source for which each is fed independently to train an Extra Trees Classifier. In this case, we keep the originally used classifier in the Quant article, since it provides class probabilities and solves the task in the best way possible, as shown in the original paper. These pre-trained classifiers are then aggregated together through an average over all probabilities.

4.5 MultiCatch22

Catch22 [14] is a time series classification approach based on the extraction of a highly optimized set of exactly 22 features from the data. These 22 features were systematically distilled from a massive pool of over 7 000 features, from a specific library, by selecting those that offer the best balance of high classification accuracy, low mathematical redundancy, and fast computation times. By transforming raw, high-dimensional time series data into these 22 specific characteristics, which capture properties like data distribution, temporal entropy, and linear correlations. It allows traditional machine learning models (like Random Forests or SVMs) to efficiently classify complex time series data without the computational overhead of deep learning.

MultiCatch22 Feature Level This strategy is based on the same underlying principle as the two previously presented "Feature Level" methods. Specifically, a catch22 transformation is applied, the extracted features from each source are concatenated, and classification is subsequently performed using a Random Forest classifier.

MultiCatch22 Decision Level This strategy is similarly based on the two previously presented "Decision Level" methods. Each source undergoes a catch22 transformation, after which a Random Forest classifier is applied to their respective features independently. Finally, the selection of the maximum is done over the aggregated probabilities to yield the final prediction.

5 Experimental Results

5.1 Implementation details

All experiments utilized min-max normalization across the entire dataset per channel per source. To prevent data leakage, minimums and maximums values were calculated only from the training split. For deep learning methods, we employed a validation split, selecting the best-performing model based on validation loss. For all non-deep learning methods, the validation split was concatenated with the training split. All proposed methods were executed on a system equipped with an Intel Core i9-14900K CPU and 128GB of RAM. Deep learning models were trained utilizing an NVIDIA GeForce RTX 4090 GPU with 24GB of VRAM. For the implementation of the code we used the aeon library [15]. The code will be available upon acceptance.

5.2 Comparing Strategies

The most prominent finding across nearly all evaluated methodologies(Table 1 is the consistent superiority of decision-level fusion over feature-level fusion. With the deep learning architectures, the "Decision Level"(DL) methods, where models learn source-specific representations and aggregate probabilities at the end. It systematically outperforms their "Feature Level"(FL) counterparts. Similarly, in the non-deep learning models, the DL strategies consistently yield higher accuracy than the FL strategies (e.g., MultiMultiRocket DL at 66.37% compared to FL at 64.25%). This suggests that forcing a joint representation early in the learning process may introduce conflicting signals and source-specific noise, particularly given the distinct spatial and spectral resolutions of Sentinel-2 and PlanetScope data. Allowing models to specialize per source before making a combined decision proves to be a more robust approach.

The highest overall accuracies are achieved by deep learning models, specifically MultiLITETime DL (69.78%) and MultiTempCNNTime DL (69.24%). The addition of temporal features provides a significant performance boost over the base LITE and TempCNN architectures, highlighting the critical importance of phenological timing in tree species classification. Furthermore, evaluating the individual sources reveals that PlanetScope(PS) consistently outperforms Sentinel-2(S2) when processed in isolation. However, combining both sources(PS & S2) yields a higher accuracy than either source alone.

While deep learning achieves the highest peak accuracy, the MultiMultiRocket DL strategy demonstrates highly competitive performance. Achieving an accuracy of 66.37% and tying for the highest F1-Score (0.66) and Recall (0.66), MultiRocket proves that extracting massive amounts of random convolutional features independently per source, followed by simple logistic regression, is a strong baseline. Given MultiRocket's typically lower computational overhead compared to deep neural networks, this highlights a valuable trade-off between performance and efficiency. However, MultiRocket needs a huge amount of RAM to compute.

Table 1. Overall Performance Comparison of the Proposed Multisource Time Series Classification Methods

Classification Method	**Accuracy(%)**	**Precision**	**Recall**	**F1-Score**
Deep Learning Ensemble				
MultiLITETime FL	65.24	0.64	0.60	0.61
MultiTempCNNTime FL	67.88	<u>0.66</u>	0.64	<u>0.65</u>
MultiLITETime DL	**69.78**	**0.68**	<u>0.65</u>	**0.66**
MultiTempCNNTime DL	<u>69.24</u>	**0.68**	0.64	**0.66**
Deep Learning				
MultiLITE FL	58.80 ± 1.37	0.59 ± 0.01	0.53 ± 0.01	0.54 ± 0.01
MultiTempCNN FL	66.05 ± 0.00	0.64 ± 0.00	0.62 ± 0.00	0.62 ± 0.00
MultiLITE DL(PS&S2)	60.43 ± 1.17	0.59 ± 0.01	0.56 ± 0.01	0.56 ± 0.01
MultiTempCNN DL(PS&S2)	60.67 ± 0.37	0.60 ± 0.01	0.55 ± 0.01	0.56 ± 0.00
Distance-Based				
MultiDTW-NN	63.57	0.65	0.64	0.64
MultiMSM-NN	64.63	0.65	<u>0.65</u>	<u>0.65</u>
Convolution-Based				
MultiMultiRocket FL	64.25	0.64	0.64	0.64
MultiMultiRocket DL	66.37	<u>0.66</u>	**0.66**	**0.66**
Interval-Based				
MultiQuant FL	60.37	0.63	0.60	0.59
MultiQuant DL	61.78	0.64	0.62	0.61
Feature-Based				
MultiCatch22 FL	57.77	0.60	0.58	0.56
MultiCatch22 DL	59.58	0.62	0.60	0.58

Note: Bold values indicate the best performance (including ties), and underlined values indicate the second-best performance (including ties).

The distance-based baselines, particularly MultiMSM-NN (64.63%), perform surprisingly well, rivaling several deep learning approaches, it has better results than a single MultiTempCNN or MultiLITE. This indicates that alignment-based similarity metrics remain highly relevant for temporal shifts in environmental data. Conversely, the feature-based (MultiCatch22) and interval-based (MultiQuant) methods yield the lowest overall performance. This suggests that the highly specific, handcrafted features or simple quantile representations extracted by these methods may not capture the complex, long-term seasonal dependencies required to differentiate urban tree species as effectively as learned convolutional filters.

Table 2. Results of single source models

Classification Method	Accuracy(%)	Precision	Recall	F1-Score
Deep Learning				
MultiTempCNN DL(PS)	57.88 ± 0.69	0.57 ± 0.01	0.53 ± 0.01	0.54 ± 0.01
MultiTempCNN DL(S2)	48.14 ± 1.08	0.49 ± 0.01	0.43 ± 0.02	0.43 ± 0.02
MultiLITE DL(PS)	56.60 ± 1.61	0.56 ± 0.01	0.52 ± 0.02	0.53 ± 0.02
MultiLITE DL(S2)	48.73 ± 1.75	0.49 ± 0.02	0.44 ± 0.02	0.44 ± 0.02
Deep Learning Ensemble				
MultiTempCNNTime DL(PS)	66.75	0.65	0.63	0.64
MultiTempCNNTime DL(S2)	58.85	0.58	0.53	0.54
MultiLITETime DL(PS)	66.75	0.65	0.63	0.64
MultiLITETime DL(S2)	59.30	0.58	0.54	0.55

We also evaluated every sources individually (Table 2). The data clearly quantifies the performance gap between the two sources, with PlanetScope-only models maintaining accuracy levels between 56% and 58%, whereas Sentinel-2 models struggle to surpass the 48% mark. This disparity suggests that the higher spatial resolution of PlanetScope is more critical for accurately capturing the structural nuances of urban tree species than the broader spectral richness offered by Sentinel-2. In this context, the superior spatial resolution of PlanetScope appears to outweigh the benefits of the broader spectral range provided by Sentinel-2. Despite PlanetScope's relative dominance as a standalone source, the substantial drop in F1-scores and accuracy compared to the multi-source models confirms that the complementary features of Sentinel-2 remain indispensable for reaching optimal classification performance. We hypothesize that the rich spectro-temporal diversity provided by Sentinel-2 captures critical phenological signatures over time, effectively compensating for its lower spatial resolution. Consequently, the fusion models achieve superior performance by successfully combining the high spatial precision of PlanetScope with the deep spectro-temporal dynamics of Sentinel-2.

6 Conclusion

In this work, we addressed the multisource SITS urban tree species classification problem. We systematically evaluate state of the art methods improved with our multisource strategies. Our experimental results revealed that deep learning models and non-deep learning models with "Decision Level" strategy are the best performing methods for the utilized dataset. In future work, we will evaluate the proposed adapted models and strategies on other multisource time series classification datasets of other applications.

Acknowledgments. This work was supported by the ANR M2-BDA project (ANR-24-CE23-1130) of the French Agence Nationale de la Recherche. The authors would like to acknowledge the High Performance Computing Center of the University of Strasbourg for supporting this work by providing scientific support and access to computing resources. Part of the computing resources were funded by the Equipex Equip@Meso project (*Programme Investissements d'Avenir*) and the CPER Alsacalcul/Big Data.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bressant, C., Wenger, R., Michéa, D., Puissant, A.: Urban tree species benchmark dataset for time series classification. Data in Brief **61**, 111777 (2025)
2. Dempster, A., Foumani, N.M., Tan, C.W., Miller, L., Mishra, A., Salehi, M., Pelletier, C., Schmidt, D.F., Webb, G.I.: Monster: Monash scalable time series evaluation repository. Journal of Data-centric Machine Learning Research (2025)
3. Dempster, A., François, P., Webb, G.I.: Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. Data Mining and Knowledge Discovery **34**(5), 1454–1495 (2020)
4. Dempster, A., Schmidt, D.F., Webb, G.I.: Minirocket: A very fast (almost) deterministic transform for time series classification. In: Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining. pp. 248–257 (2021)
5. Dempster, A., Schmidt, D.F., Webb, G.I.: Hydra: competing convolutional kernels for fast and accurate time series classification: A. dempster et al. Data Mining and Knowledge Discovery **37**(5), 1779–1805 (2023)
6. Dempster, A., Schmidt, D.F., Webb, G.I.: quant: a minimalist interval method for time series classification. Data Mining and Knowledge Discovery **38**(4), 2377–2402 (2024)
7. Gunatilaka, A.H., Baertlein, B.A.: Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. IEEE transactions on pattern analysis and machine intelligence **23**(6), 577–589 (2001)
8. Holder, C., Guijo-Rubio, D., Bagnall, A.: Clustering time series with k-medoids based algorithms. In: International Workshop on Advanced Analytics and Learning on Temporal Data. pp. 39–55. Springer (2023)
9. Holder, C., Middlehurst, M., Bagnall, A.: A review and evaluation of elastic distance functions for time series clustering. Knowledge and Information Systems **66**(2), 765–809 (2024)

10. Ismail-Fawaz, A., Devanne, M., Berretti, S., Weber, J., Forestier, G.: Look into the lite in deep learning for time series classification. International Journal of Data Science and Analytics pp. 1–21 (2025)
11. Ismail-Fawaz, A., Devanne, M., Weber, J., Forestier, G.: 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. IEEE (2022)
12. Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., Webb, G.I., Idoumghar, L., Muller, P.A., Petitjean, F.: Inceptiontime: Finding alexnet for time series classification. Data mining and knowledge discovery **34**(6), 1936–1962 (2020)
13. Li-wei, H.L., Adams, R.P., Mayaud, L., Moody, G.B., Malhotra, A., Mark, R.G., Nemati, S.: A physiological time series dynamics-based approach to patient monitoring and outcome prediction. IEEE journal of biomedical and health informatics **19**(3), 1068–1076 (2014)
14. Lubba, C.H., Sethi, S.S., Knaute, P., Schultz, S.R., Fulcher, B.D., Jones, N.S.: catch22: Canonical time-series characteristics: Selected through highly comparative time-series analysis. Data Mining and Knowledge Discovery **33**(6), 1821–1852 (2019)
15. Middlehurst, M., Ismail-Fawaz, A., Guillaume, A., Holder, C., Guijo-Rubio, D., Bulatova, G., Tsaprounis, L., Mentel, L., Walter, M., Schäfer, P., et al.: aeon: a python toolkit for learning from time series. Journal of Machine Learning Research **25**(289), 1–10 (2024)
16. Ordóñez, F.J., Roggen, D.: Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors **16**(1), 115 (2016)
17. Pelletier, C., Webb, G.I., Petitjean, F.: Temporal convolutional neural network for the classification of satellite image time series. Remote Sensing **11**(5), 523 (2019)
18. Ratanamahatana, C.A., Keogh, E.: Three myths about dynamic time warping data mining. In: Proceedings of the 2005 SIAM international conference on data mining. pp. 506–510. SIAM (2005)
19. Solana, A., Fransén, E., Uribarri, G.: Classification of raw meg/eeg data with detach-rocket ensemble: an improved rocket algorithm for multivariate time series analysis. In: International Workshop on Advanced Analytics and Learning on Temporal Data. pp. 96–114. Springer (2024)
20. Stefan, A., Athitsos, V., Das, G.: The move-split-merge metric for time series. IEEE transactions on Knowledge and Data Engineering **25**(6), 1425–1438 (2012)
21. Tan, C.W., Dempster, A., Bergmeir, C., Webb, G.I.: Multirocket: multiple pooling operators and transformations for fast and effective time series classification: Cw tan. Data Mining and Knowledge Discovery **36**(5), 1623–1646 (2022)
22. Wang, Y., Peng, J., Zhang, J., Yi, R., Wang, Y., Wang, C.: Multimodal industrial anomaly detection via hybrid fusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8032–8041 (2023)
23. Wang, Z., Yan, W., Oates, T.: 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 (2017)
24. Wenger, R., Puissant, A., Weber, J., Idoumghar, L., Forestier, G.: Multisenge: A multimodal and multitemporal benchmark dataset for land use/land cover remote sensing applications. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences **3**, 635–640 (2022)