pytskit: A Comprehensive Time Series Toolkit

Abstract

We present `pytskit`, the most comprehensive Python library for time series analysis to date, expressly designed to address both the breadth and depth of the domain. `pytskit` is an actively evolving library that supports five core time series analysis tasks: classification, clustering, anomaly detection, forecasting, and similarity search, housing the current largest collection of time series algorithms. The library integrates a diverse collection of 272 algorithms, ranging from classical statistical methods to modern deep neural networks and recently emerging foundation models. It features a unified, `scikit-learn`-compatible API to ensure ease of use and simplify integration into existing data science workflows. Notably, this design serves a dual purpose: it not only provides practitioners with accessible model functionality but also enables researchers to conduct rigorous, comprehensive benchmark studies, overcoming common limitations in existing libraries, such as discontinued maintenance or insufficient flexibility for advanced research. The library is available under the Apache-2.0 license at https://github.com/thedatumorg/pytskit.

Keywords: machine learning, data mining, time series analysis, open source, Python

1 Introduction

Time series data, consisting of ordered sequences of observations, are ubiquitous across diverse scientific and industrial domains, including econometrics, healthcare, environmental science, and industrial process monitoring (Wang et al., 2024). The rapid growth in data has spurred a proliferation of specialized algorithms and libraries for tasks like classification, clustering, forecasting, anomaly detection, and similarity search (Middlehurst et al., 2024a; Löning et al., 2019; Zhao et al., 2019; Oguiza, 2023). This has created a fragmented software landscape, where algorithms are frequently released as standalone implementations with varying standards of documentation and usability.

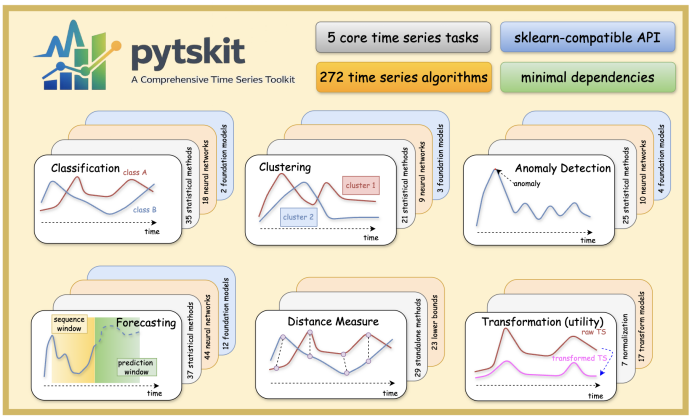

Figure 1: **pytskit** Architecture.

Despite the availability of various time series libraries (Löning et al., 2019; Middlehurst et al., 2024a; Herzen et al., 2022; Oguiza, 2023; Wu et al., 2023), the current ecosystem faces significant limitations regarding scope, maintenance, and integration: (i) many existing libraries exhibit fragmented coverage with discontinuous support, failing to span the full spectrum of methods—typically focusing on either classical statistical techniques while neglecting modern foundation models, or vice versa; and furthermore (ii) these libraries typically exhibit narrow specialization, targeting isolated tasks, e.g., forecasting only, rather

Table 1: Comparison across libraries. (Data collected through December 2025).

Library	Anomaly	Classification	Clustering	Forecasting	Similarity	**Total**
tsai (Oguiza, 2023)	0	34	0	34	0	68
tslib (Wu et al., 2023)	15	19	0	39	1	74
Aeon (Middlehurst et al., 2024a)	12	56	18	9	16	111
DARTS (Herzen et al., 2022)	18	17	0	47	41	123
sktime (Löning et al., 2019)	7	53	8	72	28	188
pytskit	**39**	**55**	**33**	**93**	**52**	**272**

than the broader analysis spectrum. This fragmentation compels researchers to integrate heterogeneous and often incompatible tools, complicating future advancement.

To address these limitations, we present `pytskit`, the current most comprehensive Python library for time series analysis. `pytskit` is designed to provide a single, unified framework for both research and application with a broad coverage of five key tasks: classification, clustering, forecasting, anomaly detection, and similarity search. `pytskit` integrates a diverse model repository, including traditional statistical methods, deep neural networks, and newly emerging foundation models released in the last two years, all accessible through a high-level, `scikit-learn`-compatible API. Additionally, the library provides a standardized end-to-end evaluation pipeline and comprehensive metrics suite, inheriting rigorous validation protocols from recent published research studies (d'Hondt et al., 2025; Middlehurst et al., 2024c; Paparrizos and Bogireddy, 2025). By unifying these disparate tasks, `pytskit` aims to lower the barrier to entry for complex time series analysis and facilitate reproducible, high-quality research. In the following, we provide an overview of `pytskit`, from the high-level code design to the coverage of five time-series tasks.

2 Code Design and Implementation

The library comprises five core modules dedicated to time-series classification, clustering, anomaly detection, forecasting, and similarity search (supported with a large collection of distance measures), along with necessary utility modules such as transformation. Furthermore, `pytskit` uses an API consistent with the `scikit-learn` api style, using standard methods such as `fit()` and `predict()`. This simplifies pipeline construction and integration into existing workflows, supporting user input or JSON-style parameter control. We show a simple code example for time-series classification as follows.

```python
import numpy as np
from pytskitts.classification.kernel.svc import SupportVectorClassifier
# Create sample time series data (3D array: samples, channels, timesteps)
X = [[[1, 2, 3, 4, 5, 5]],  # First sample
     [[8, 7, 6, 5, 4, 4]]]  # Second sample
y = ['low', 'high']  # Class labels
X, y = np.array(X), np.array(y)
# Create and train classifier
clf = SupportVectorClassifier(C=1, class_weight="balanced")
clf.fit(X, y)  # fit the classifier
# Make predictions on new data (list or numpy array style)
X_test = [[[2, 2, 2, 2, 2, 2]],
          [[6, 6, 6, 6, 6, 6]]]
y_pred = clf.predict(X_test)  # predict on test data, ['low', 'high']
```

Code Snippet 1: Example Usage of pytskit

3 Time Series Core Modules

This section details the five core functional modules of `pytskit`: classification, clustering, anomaly detection, forecasting, and similarity search. For each, we outline the scope of its functionality and the diversity of its model repository. Per table 1, `pytskit` offers the most comprehensive toolkit across all major tasks compared to other leading libraries to date.

3.1 Classification

Time series classification involves mapping an input time series to a discrete class label, where each class could represent distinct events or identities (Middlehurst et al., 2024d). The `pytskit` classification module integrates an expansive selection of 55 methods, spanning established statistical approaches, advanced deep learning (DL) architectures, and modern foundation models (FM). However, the module extends beyond a simple repository of algorithms by providing a robust, end-to-end pipeline for model execution and validation on custom datasets or widely recognized benchmarks, such as the UCR Archive (Dau et al., 2018), supported with widely-used evaluation metrics.

Table 2: Classifiers in `pytskit`

Domain	Family	Count
Statistical (Total: 35)	Baseline	2
	Feature-based	3
	Distance-based	4
	Interval-based	7
	Dictionary-based	10
	Shapelet-based	4
	Kernel-based	5
DL & FM (Total: 20)	MLP Baselines	2
	CNN-based	11
	RNN-based	1
	RNN-CNN Hybrids	3
	Transformer-based	1
	Foundation Models	2

3.2 Clustering

Time series clustering is an unsupervised learning task that partitions time series samples into groups based on similarity. (Paparrizos et al., 2024). This task is challenging due to high dimensionality, complex temporal dependencies and noise. To address these challenges, the `pytskit` clustering module provides a comprehensive collection of 33 distinct clustering algorithms: 21 statistical, 9 deep learning, and 3 foundation. To ensure robust performance assessment in this unsupervised setting, `pytskit` incorporates a standardized evaluation pipeline , enabling reproducible comparisons of cluster quality across diverse datasets.

Table 3: Clustering in `pytskit`.

Domain	Family	Count
Statistical (Total: 21)	Partition-based	11
	Distribution-based	1
	Density-based	4
	Hierarchical-based	2
	Kernel-based	2
	Shapelet-based	1
DL & FM (Total: 12)	Autoencoder-based	9
	Foundation Models	3

3.3 Anomaly Detection

Time series anomaly detection identifies data points or subsequences that deviate significantly from normal behavior (Liu and Paparrizos, 2024). Building on recent benchmark studies (Liu and Paparrizos, 2024), the `pytskit` anomaly detection module supports 25 statistical and 14 deep learning and foundation models To ensure the reliability of these methods, `pytskit` incorporates a robust evaluation pipeline to handle the complexities in the anomaly detection process. This pipeline supports a suite of advanced metrics such as F1, VUS, and Affiliation Score.

Table 4: Anomaly Detection in `pytskit`.

Domain	Subcategory	Count
Statistical (Total: 25)	Density & Distance	7
	Probabilistic	5
	Linear & Kernel	5
	Shape & Frequency	5
	Isolation-Based	3
DL & FM (Total: 14)	Reconstruction	4
	Forecasting	4
	Transformers	2
	Foundation Models	4

3.4 Forecasting

Time series forecasting involves modeling historical, time-ordered data to predict future values. The forecasting module currently supports 92 methods, including 23 statistical methods, 14 machine learning methods, 43 deep learning architectures, and 12 emerging foundation models. Consistent with the other modules, the forecasting module provides both end-to-end pipelines and a suite of widely-adopted evaluation metrics to facilitate user-friendly application and support rigorous comparative studies.

Table 5: Forecasting Models in `pytskit`.

Domain	Family	Count
Statistical (Total: 37)	Naive Forecasters	4
	Basic Statistical	16
	Advanced Statistical	17
DL & FM (Total: 56)	MLP-based	7
	CNN-based	4
	RNN-based	6
	Attention-based	22
	Hybrid-based	5
	Foundation Models	12

3.5 Similarity Search (Distance Measure)

A distance measure quantifies the dissimilarity between two time series, an operation essential for downstream tasks such as classification, anomaly detection, and clustering (Paparrizos et al., 2020; d'Hondt et al., 2025). `pytskit`'s distance module provides a library of 29 standalone measures.

To address the high computational complexity of elastic measures, the module integrates GLB (Paparrizos et al., 2023). This framework provides 23 efficient lower bounds for the 7 elastic measures, resulting in 23 optimized combinations. This feature enables substantial computational speedups, making these measures practical for large-scale datasets.

Table 6: Dist. Measures

Measure Category	Count
Lock-step	11
Elastic	5
Sliding	1
Kernel	4
Feature-based	1
Model-based	2
Embedding	5
GLB	23
Total Measures	**52**

4 Time Series Utility Modules

pytskit offers a broad suite of utility modules that facilitate real-world deployments and rigorous evaluation studies. These modules handle operations that are prerequisites for many algorithms, such as pre-processing, feature engineering, and similarity assessment. Specificaly, we provide transformation models, which are estimators that map time series data to a different representation, often for pre-processing or feature engineering (Middlehurst et al., 2024b). The `pytskit` transformation module provides a comprehensive suite of these tools. The module is organized into two categories: 7 essential normalization and scaling methods and 17 transformation models designed for feature extraction.

Table 7: Transformations

Model Category	Count
Normalization & Scaling	7
Feature Extraction	17
Total Models	**24**

5 Conclusion

Several comprehensive open-source Python packages have made significant contributions to time series analysis. However, this landscape remains fragmented, with many libraries focusing on narrow tasks or lacking continuous support. `pytskit` builds upon these efforts to address this gap, and uniquely integrates the most extensive collection of classical, deep learning, and foundation models available for the standard array of time series tasks.

References

H. A. Dau, E. Keogh, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, Y. Chen, B. Hu, N. Begum, A. Bagnall, A. Mueen, G. Batista, and Hexagon-ML. The ucr time series classification archive, October 2018. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/.

J. E. d'Hondt, H. Li, F. Yang, O. Papapetrou, and J. Paparrizos. A structured study of multivariate time-series distance measures. *Proceedings of the ACM on Management of Data*, 3(3):1–29, 2025. doi: 10.1145/3725258.

J. Herzen, F. Lässig, S. G. Piazzetta, T. Neuer, L. Tafti, G. Raille, T. V. Pottelbergh, M. Pasieka, A. Skrodzki, N. Huguenin, M. Dumonal, J. Kościsz, D. Bader, F. Gusset, M. Benheddi, C. Williamson, M. Kosinski, M. Petrik, and G. Grosch. Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, 23(124): 1–6, 2022. URL http://jmlr.org/papers/v23/21-1177.html.

Q. Liu and J. Paparrizos. The elephant in the room: Towards a reliable time-series anomaly detection benchmark. In *NeurIPS 2024*, 2024.

M. Löning, A. Bagnall, S. Ganesh, V. Kazakov, J. Lines, and F. Király. sktime: A unified interface for machine learning with time series. In *Workshop on Systems for ML at NeurIPS 2019*, 2019.

M. Middlehurst, A. Ismail-Fawaz, A. Guillaume, C. Holder, D. Guijo-Rubio, G. Bulatova, L. Tsaprounis, L. Mentel, M. Walter, P. Schäfer, and A. Bagnall. aeon: a python toolkit for learning from time series. *Journal of Machine Learning Research*, 25(144):1–6, 2024a. URL http://jmlr.org/papers/v25/23-1444.html.

M. Middlehurst, A. Ismail-Fawaz, A. Guillaume, C. Holder, D. G. Rubio, G. Bulatova, L. Tsaprounis, L. Mentel, M. Walter, P. Schäfer, and A. Bagnall. aeon: a python toolkit for learning from time series, 2024b. URL https://arxiv.org/abs/2406.14231.

M. Middlehurst, P. Schäfer, and A. Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, 38(4):1–68, 2024c. doi: 10.1007/s10618-023-00994-6.

M. Middlehurst, P. Schäfer, and A. Bagnall. Bake off redux: a review and experimental evaluation of recent time series classification algorithms. *Data Mining and Knowledge Discovery*, 38(4):1958–2031, Apr. 2024d. ISSN 1573-756X. doi: 10.1007/s10618-024-01022-1. URL http://dx.doi.org/10.1007/s10618-024-01022-1.

I. Oguiza. tsai - a state-of-the-art deep learning library for time series and sequential data, 2023. URL https://github.com/timeseriesAI/tsai.

J. Paparrizos and S. P. T. R. Bogireddy. Time-series clustering: A comprehensive study of data mining, machine learning, and deep learning methods. *Proceedings of the VLDB Endowment*, 18(11):4380–4395, 2025. doi: 10.14778/3749646.3749700.

J. Paparrizos, C. E. A. Palpanas, Liu, and M. J. Franklin. Debunking four long-standing misconceptions of time-series distance measures. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1887–1905, 2020. doi: 10.1145/3318464.338976.

J. Paparrizos, K. Wu, A. Elmore, C. Faloutsos, and M. J. Franklin. Accelerating similarity search for elastic measures: A study and new generalization of lower bounding distances. *Proceedings of the VLDB Endowment*, 16(8):2019–2032, 2023. doi: 10.14778/3594512.3594530. URL https://doi.org/10.14778/3594512.3594530.

J. Paparrizos, F. Yang, and H. Li. Bridging the gap: A decade review of time-series clustering methods, 2024. URL https://arxiv.org/abs/2412.20582.

Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang. Deep time series models: A comprehensive survey and benchmark, 2024. URL https://arxiv.org/abs/2407.13278.

H. Wu, T. Hu, Y. Liu, H. Zhou, J. Wang, and M. Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Y. Zhao, Z. Nasrullah, and Z. Li. Pyod: A python toolbox for scalable outlier detection. *Journal of Machine Learning Research*, 20(96):1–7, 2019. URL http://jmlr.org/papers/v20/19-011.html.