




nrt: operational monitoring of satellite image time-series in Python

Loïc Dutrieux ^{1*} and Jonas Viehweger ^{1,2,3*}

1 European Commission, Joint Research Centre, Ispra, Italy **2** Philipp University of Marburg, RWTH Aachen University, Germany **3** Sinergise Solutions GmbH, Graz, Austria * These authors contributed equally.

DOI: [10.21105/joss.06815](https://doi.org/10.21105/joss.06815)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: 

Reviewers:

- [@alexgleith](#)
- [@szwiep](#)

Submitted: 13 May 2024

Published: 13 August 2024

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Abstract

nrt is a Python package specifically designed for near real-time detection of anomalies in gridded spatio-temporal data, and particularly suited for monitoring forest disturbances from satellite image time-series. The core of the package provides a standardized Application Programming Interface (API) that facilitates interoperability and comparison across a variety of monitoring algorithms. These algorithms, selected from the scientific literature, are implemented in an optimized manner to enable efficient computation across large geographical areas. This development represents a significant advancement in environmental monitoring efforts, enhancing our ability to contribute to climate change mitigation, biodiversity conservation, and the preservation of natural heritage.

Statement of need

Accurate mapping and monitoring of land dynamics is critical for climate change mitigation, biodiversity conservation, and epidemic prevention ([Forest Europe, 2020](#)). With increasing data availability and processing capacities, there is a growing desire to move from periodic mapping of forest disturbances to continuous monitoring systems capable of providing timely information on forest disturbances to a variety of stakeholders ([Woodcock et al., 2020](#)). Many algorithms and approaches have been proposed in the research community to address this near real-time monitoring challenge. Their performance is typically demonstrated based on case studies over test areas or simulated datasets. However, when it is available, the software provided with the research papers only offers limited operational capacity. Individual software is often primarily developed to support the research experiment and consequently not optimized for speed or deployment at scale. In addition, implementation in different programming languages or the absence of a common interface to operate the algorithm make interoperability and comparisons exercises challenging. The nrt package directly addresses these shortcomings by providing a standardized Application Programming Interface (API) to various near real-time disturbance detection approaches, all optimized for rapid computation and specifically designed for effective operational deployment. Furthermore, the API design is inspired by the highly successful scikit-learn library, considered the de-facto standard for machine learning in Python, and works seamlessly with xarray, the industry standard for representing spatio-temporal gridded data ([Buitinck et al., 2013](#)). This ensures a familiar and efficient workflow for a broad range of users.

Online monitoring of time-series data

The term ‘online monitoring’ refers to the real-time analysis of open-ended time-series that grow dynamically, with each new data point updating the analysis values in real-time. This method contrasts with ‘offline’ approaches, which involve retrospective analysis of static time-series data. Online monitoring is particularly suited for applications requiring immediate responses, such as anomaly detection in environmental monitoring or system health tracking. In the context of satellite imagery, where spaceborne sensors such as Sentinel 2 and Landsat – the main optical moderate resolution constellations – collect data with a revisit period of 5 and 8 days respectively near the equator, each pixel is treated as an independent online time-series. Various vegetation indices, indicative of traits such as photosynthetic activity, foliage density, or water content, are derived from raw reflectance data to monitor stability as a univariate time-series. To establish stable behavior, statistical models are employed; often, linear harmonic regressions are used to model typical seasonal behavior over what is termed the stable history period. Disturbances in vegetation, such as removal due to logging or stress from biotic and abiotic factors like insects, diseases, and droughts, typically manifest as anomalies in these time-series (Figure 1).

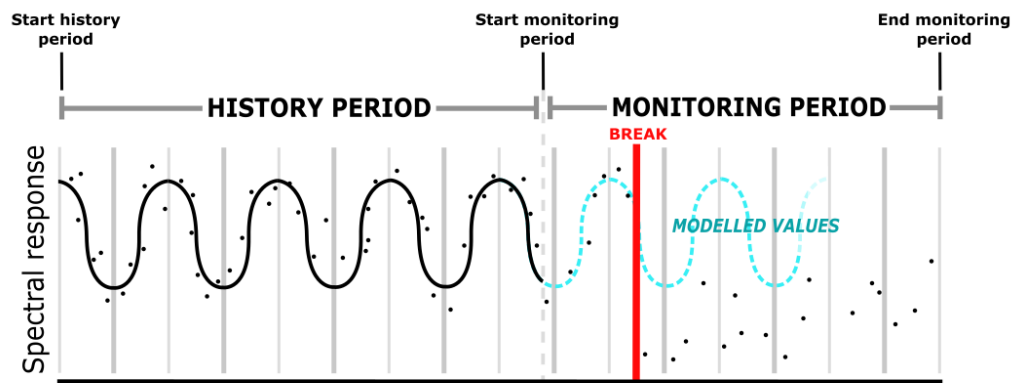


Figure 1: Illustration of the online monitoring process for a univariate time-series, depicting the distinct phases including the history period, monitoring period, and a detected break against modeled values.

Different strategies for detecting these anomalies define the unique monitoring approaches implemented in the `nrt` package. The operational deployment of online monitoring systems, particularly for satellite imagery, requires efficient data management strategies. Since keeping monitoring processes in memory would tie up computing resources for prolonged periods, these processes must instead be saved on disk. They are then reloaded and updated periodically as new satellite data becomes available.

Main functionalities

A common monitoring framework

The monitoring algorithms in `nrt` are structured as subclasses of an abstract base class that defines a general monitoring framework. This framework is designed to operate with `xarray` `DataArray` data structures and includes several key methods:

- `fit`: This method is used for model fitting over the stable history period, employing temporal regression with harmonic components. This step is typically the most computationally intensive, but efficiency is enhanced through vectorized operations and the use

of the numba Just In Time (JIT) compiler (Lam et al., 2015). Most fitting processes can also be parallelized directly via the API.

- `monitor`: Updates the monitoring status with the latest available data.
- `report`: Exports variables of interest from the monitoring status into a georeferenced raster file.
- `to/from_netcdf`: Manages the dumping and loading of the current monitoring status to and from disk for persistent storage and agile monitoring.

The framework also defines several noteworthy attributes:

- `mask`: A two-dimensional spatial array that provides high-level information about the monitoring status, with different integer labels denoting whether pixels are currently being monitored, unstable, no longer monitored, etc.
- `beta`: Represents the regression coefficient matrix from the temporal linear regression.
- `process` and `boundary`: These attributes slightly vary depending on the monitoring approach but generally relate to the dynamic values updated during monitoring. Typically, a process value crossing the boundary indicates the detection of a breakpoint, triggering further actions in the monitoring workflow.

State of the art monitoring algorithm implemented

The `nrt` package currently implements five distinct monitoring approaches:

- Exponentially Weighted Moving Average (EWMA) (Brooks et al., 2013).
- Cumulative Sum of Residual (CuSum) (Verbesselt et al., 2012; Zeileis et al., 2005). CuSum is a monitoring option in the `bfastmonitor` function of the R package `bfast`.
- Moving Sum of Residuals (MoSum) (Verbesselt et al., 2012; Zeileis et al., 2005). Like CuSum, MoSum is available through the `bfastmonitor` function in `bfast`.
- Continuous Change Detection and Classification of land cover (CCDC, CMFDA) (Zhu et al., 2012; Zhu & Woodcock, 2014) - Focuses on the core change detection component of the original CCDC algorithm.
- InterQuantile Range (IQR) - A simple, unpublished outlier identification strategy proposed by Rob J. Hyndman on [StackExchange](#).

While all temporal fitting methods are compatible with each monitoring algorithm, the default settings are tailored independently for each approach, adhering to the original manuscripts' recommendations. For example, Verbesselt et al. (2012) recommend performing a stability test on the historical period to ensure no disturbances are present; thus, the default fitting strategy for CuSum and MoSum is ROC (Reversed Cumulative Sum of residuals). However, if the historical period is known to be stable and free of disturbances, users may opt for a simpler and faster fitting strategy such as OLS (Ordinary Least Squares).

Demo data

The `nrt` package features a data module that provides access to both real and simulated datasets. Included in the module is a subset of a Sentinel 2 multi-temporal data cube, completed with an associated forest mask. This real data set is ideal for demonstration purposes.

Additionally, the data simulation functionalities of the module allow for the creation of synthetic time-series. Users can fully control various parameters, such as seasonal amplitude, noise level, frequency of outliers, and the magnitude of breaks. These capabilities are especially valuable for development, testing, and conducting sensitivity analyses.

Current use

The most significant application of the `nrt` package to date is its deployment as a prototype alert system at the Joint Research Center. This system ran operationally throughout the 2023

vegetation season, covering the entire territory of Estonia (Figure 2). The system utilizes the EWMA algorithm from `nrt`, minimally tuned, and employs Normalized Difference Vegetation Index (NDVI) and Normalized Difference Moisture Index (NDMI) indices to generate forest disturbance alerts.

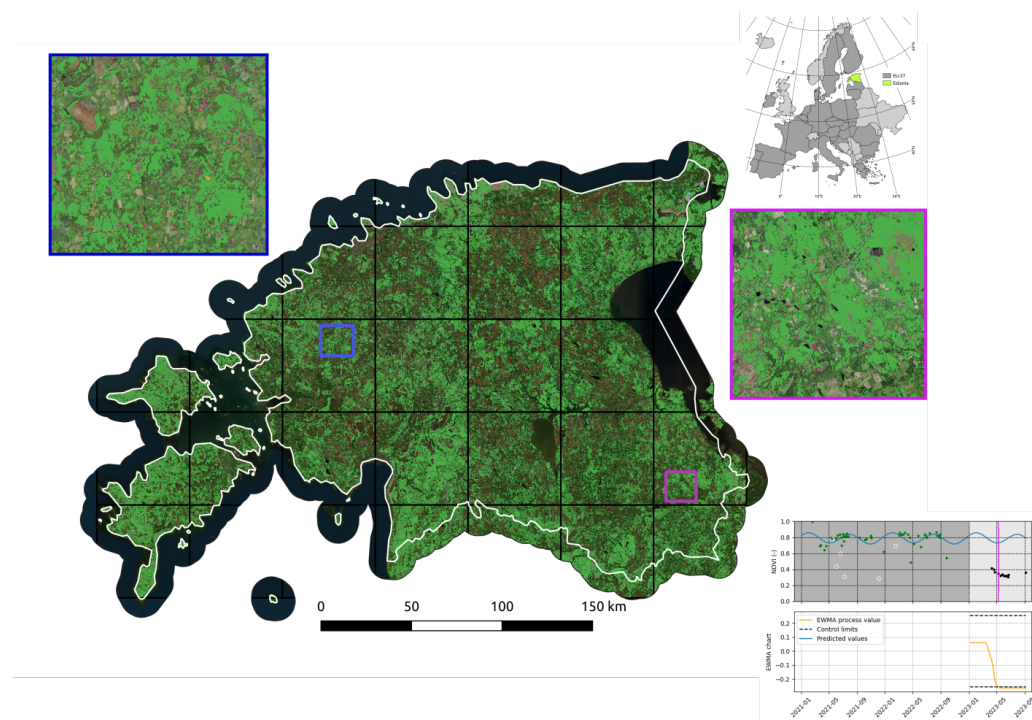


Figure 2: Snapshot of the operational forest disturbance alert system deployed across Estonia during the 2023 vegetation period. The bottom-right panel shows an example of a tree cover loss event that was successfully detected.

Estonia was divided into 50 km tiles, and a monitoring instance for each tile was established using data from 2021 and 2022 as a stable history period. Each night, a containerized process automatically queried the SpatioTemporal Asset Catalog (STAC) on the Big Data Analytics Platform (BDAP) for new Sentinel 2 data (Soille et al., 2018; STAC Contributors, 2021). When new data was available, it was used to update both the monitoring status and the alert layer, which was accessible via Web Map Service (WMS) protocol.

It is important to note that tools like PySTAC and Open Data Cube (ODC) played a crucial role in the deployment and functionality of this pipeline (Hanson, 2019; Killough, 2018).

Future directions

Active development on `nrt` and its ecosystem continues, and we warmly welcome external contributors. The canonical repository for the core `nrt` package is hosted on the [ec-jrc organization on GitHub](https://github.com/ec-jrc/ec-jrc-organization) and will remain there, while future additions such as namespace packages or additional material will be hosted on code.europa.eu/jrc-forest. Our strategic vision aims to expand both the core functionalities of the package and its surrounding ecosystem, aspiring to establish a unified approach to forest disturbance monitoring. Currently, a multivariate monitoring mode is being designed to enhance the core functionalities. This new mode aims to mitigate the trade-offs typically faced when selecting a vegetation index for disturbance detection, where indices sensitive to greenness for instance are generally less sensitive to water content and vice versa. Developments in progress and envisioned for the

nrt ecosystem include tools and tutorials for visualization and diagnostics, computation of accuracy metrics, the creation of reference data through visual interpretation, and advanced generation of synthetic data.

Conclusion

The nrt package marks a significant advancement in near real-time monitoring of forest disturbances. By optimizing state-of-the-art algorithms and providing a standardized API, it greatly enhances interoperability and enables scalable deployment. These enhancements substantially benefit environmental monitoring efforts, contributing to climate change mitigation, biodiversity conservation, and the preservation of natural heritage. Future developments will focus on expanding its capabilities and usability, aiming to develop a unified approach to monitoring forest disturbances.

References

- Brooks, E. B., Wynne, R. H., Thomas, V. A., Blinn, C. E., & Coulston, J. W. (2013). On-the-fly massively multitemporal change detection using statistical quality control charts and Landsat data. *IEEE Transactions on Geoscience and Remote Sensing*, 52(6), 3316–3332. <https://doi.org/10.1109/TGRS.2013.2272545>
- Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., & Varoquaux, G. (2013). API design for machine learning software: Experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 108–122. <https://doi.org/10.48550/arXiv.1309.0238>
- Forest Europe. (2020). *State of Europe's forests 2020*. Ministerial Conference on the Protection of Forests in Europe.
- Hanson, M. (2019). The Open-source software ecosystem for leveraging public datasets in Spatio-Temporal Asset Catalogs (STAC). *AGU Fall Meeting Abstracts, 2019*, IN23B–07.
- Killough, B. (2018). Overview of the open data cube initiative. *IGARSS 2018-2018 IEEE International Geoscience and Remote Sensing Symposium*, 8629–8632. <https://doi.org/10.1109/IGARSS.2018.8517694>
- Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 1–6. <https://doi.org/10.1145/2833157.2833162>
- Soille, P., Burger, A., De Marchi, D., & others. (2018). A versatile data-intensive computing platform for information retrieval from big geospatial data. *Future Generation Computer Systems*, 81, 30–40. <https://doi.org/10.1016/j.future.2017.11.007>
- STAC Contributors. (2021). *SpatioTemporal Asset Catalog (STAC) specification (Version 1.0.0)*. <https://stacspec.org>.
- Verbesselt, J., Zeileis, A., & Herold, M. (2012). Near real-time disturbance detection using satellite image time series. *Remote Sensing of Environment*, 123, 98–108. <https://doi.org/10.1016/j.rse.2012.02.022>
- Woodcock, C. E., Loveland, T. R., Herold, M., & Bauer, M. E. (2020). Transitioning from change detection to monitoring with remote sensing: A paradigm shift. *Remote Sensing of Environment*, 238, 111558. <https://doi.org/10.1016/j.rse.2019.111558>
- Zeileis, A., Leisch, F., Kleiber, C., & Hornik, K. (2005). Monitoring structural change in dynamic econometric models. *Journal of Applied Econometrics*, 20(1), 99–121. <https://doi.org/10.1016/j.jap.2004.12.001>

[//doi.org/10.1002/jae.776](https://doi.org/10.1002/jae.776)

Zhu, Z., & Woodcock, C. E. (2014). Continuous change detection and classification of land cover using all available Landsat data. *Remote Sensing of Environment*, 144, 152–171. <https://doi.org/10.1016/j.rse.2014.01.011>

Zhu, Z., Woodcock, C. E., & Olofsson, P. (2012). Continuous monitoring of forest disturbance using all available Landsat imagery. *Remote Sensing of Environment*, 122, 75–91. <https://doi.org/10.1016/j.rse.2011.10.030>