CEFET/RJ

Harbinger: A Unified Framework for Event Detection in Time Series

Harbinger represents a comprehensive approach to detecting events in time-series data, providing researchers and practitioners with a unified platform for analyzing temporal patterns. This framework consolidates multiple detection methodologies under a single, coherent architecture, enabling systematic comparison and evaluation of different approaches.

Repository: https://github.com/cefet-rj-dal/harbinger
CRAN: https://cran.r-project.org/web/packages/harbinger/index.html

Eduardo Ogasawara
eduardo.ogasawara@cefet-rj.br
https://eic.cefet-rj.br/~eogasawara

CRAN Package

Harbinger

Harbinger is a framework for event detection in time series. It provides an integrated environment for anomaly detection, change point detection, and motif discovery. Harbinger offers a broad range of methods and functions for plotting and evaluating detected events.

For anomaly detection, methods are based on: - Machine learning model deviation: Conv1D, ELM, MLP, LSTM, Random Regression Forest, and SVM - Classification models: Decision Tree, KNN, MLP, Naive Bayes, Random Forest, and SVM - Clustering: k-means and DTW - Statistical techniques: ARIMA, FBIAD, GARCH

For change point detection, Harbinger includes: - Linear regression, ARIMA, ETS, and GARCH-based approaches - Classic methods such as AMOC, ChowTest, Binary Segmentation (BinSeg), GFT, and PELT

For motif discovery, it provides: - Methods based on Hashing and Matrix Profile

Harbinger also supports **multivariate time series analysis** and **event evaluation** using both traditional and soft computing metrics.

The architecture of Harbinger is based on **Experiment Lines** and is built on top of the DAL Toolbox. This design makes it easy to extend and integrate new methods into the framework.

The framework addresses a critical gap in time-series analysis by offering an integrated environment where anomaly detection, change point identification, and motif discovery can be performed using consistent interfaces and evaluation metrics. Available as an open-source R package, Harbinger facilitates reproducible research and practical applications across diverse domains.

Why a Unified Framework?

The landscape of time-series event detection is fragmented, with numerous specialized methods optimized for particular event types. While this specialization has driven innovation, it creates practical challenges for researchers and practitioners who must navigate disparate tools, inconsistent interfaces, and incompatible evaluation metrics.

Real-world applications demand more than isolated detection algorithms. They require the ability to systematically compare multiple approaches, combine complementary detectors to improve robustness, and rigorously evaluate performance using standardized metrics. Without a unified framework, these essential tasks become prohibitively complex and time-consuming.

Harbinger addresses these challenges by providing a coherent ecosystem where different detection methods can be applied, evaluated, and combined within a consistent architectural framework. This approach enables researchers to focus on the analytical questions rather than technical integration issues.

Harbinger*: Um framework para integração e análise de métodos de detecção de eventos em séries temporais

Rebecca Salles[1], Luciana Escobar[2], Laís Baroni[1], Roccio Zorrilla[3], Artur Ziviani[2], Vinicius Kreischer[1], Flavia C. Delicato[1], Paulo F. Pires[1], Luciano Maia[1], Rafaelli Coutinho[1], Laura Assis[1], Eduardo Ogasawara[1]

[1]CEFET/RJ - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

[2]LNCC - Laboratório Nacional de Computação Científica

[3]UFF - Universidade Federal Fluminense

[4]Petrobras

{rebecca.salles,luciana.vignoli,lais.baroni}@eic.cefet-rj.br

{ziviani,roccio.zorrilla,vinicius.kreischer}@lncc.br

{fdelicato,paulo.pires}@ic.uff.br, luciano.maia@petrobras.com.br

{rafaelli.coutinho,laura.assis}@cefet-rj.br, eogasawara@ieee.org

Abstract. *When analyzing time series, it is possible to observe significant changes in the behavior of its observations that frequently characterize the occurrence of events. Events may appear as anomalies, change points, or frequent patterns. In literature, there are several methods for event detection. However, the search for a suitable method for a time series is not a simple task, especially considering that the nature of the events is often not known. In this context, this work presents Harbinger, a framework for integration and analysis of event detection methods. Harbinger was evaluated with synthetic and real data, where it was possible to verify that its functionalities foster the selection of methods and the understanding of detected events.*

Resumo. *Ao analisar séries temporais é possível observar mudanças significativas no comportamento das observações que frequentemente caracterizam a ocorrência de eventos. Eventos se apresentam como anomalias, pontos de mudança, ou padrões frequentes. Na literatura existem diversos métodos para detecção de eventos. Entretanto, a busca por um método adequado para uma série temporal não é uma tarefa simples, principalmente considerando-se que a natureza dos eventos muitas vezes não é conhecida. Neste contexto, este trabalho apresenta Harbinger, um framework para integração e análise de métodos de detecção de eventos. O Harbinger foi avaliado em dados sintéticos e reais, onde foi possível constatar que suas funcionalidades promovem a seleção de métodos e a compreensão dos eventos detectados.*

1. Introdução

Na análise de séries temporais, frequentemente é possível observar a ocorrência de uma mudança significativa em seu comportamento em um certo ponto ou intervalo de tempo.

Os autores agradecem a Petrobras, FAPERJ, CAPES (cód. 001) e CNPq pelo financiamento do projeto.

50+

Detection Methods

Comprehensive algorithm coverage

10

Change Point Techniques

Specialized approaches for structural breaks

9

Motif Discovery Methods

Pattern recognition algorithms

Reference: R. Salles, L. Escobar, L. Baroni, R. Zorrilla, A. Ziviani, V. Kreischer, F. Delicato, P. F. Pires, L. Maia, R. Coutinho, L. Assis, and E. Ogasawara, "Harbinger: Um framework para integração e análise de métodos de detecção de eventos em séries temporais," in *Anais do Simpósio Brasileiro de Banco de Dados (SBBD)*, SBC, Sept. 2020, pp. 73–84. doi: 10.5753/sbbd.2020.13626.

Modular Architecture

Harbinger's architecture embodies a modular design philosophy, where distinct functional components—detection, evaluation, combination, and comparison—operate both independently and synergistically. This flexibility allows researchers to leverage individual modules for specific tasks or integrate them into comprehensive analytical pipelines.

Design Principles

The framework implements a rigid interface based on workflow algebra and experiment lines, ensuring consistency across different detection methods. Each module adheres to standardized input/output contracts, facilitating seamless composition and reproducibility.

Built on the DAL Toolbox foundation, Harbinger adopts design patterns inspired by Scikit-learn's API. The familiar `fit()` and `detect()` functions provide an intuitive interface for model training and event detection, reducing the learning curve for practitioners familiar with machine learning workflows.

This architectural approach supports extensibility, allowing researchers to integrate novel detection algorithms without disrupting existing functionality.


Future Generation Computer Systems

Deriving scientific workflows from algebraic experiment lines: A practical approach

1

Normalize

Preprocessing and standardization

2

Filter

Noise reduction and signal enhancement

3

Fit

Model training and parameter estimation

4

Detect

Event identification and classification

Learn more about the DAL Toolbox at https://cran.r-project.org/web/packages/daltoolbox

Reference: A. Marinho, D. de Oliveira, E. Ogasawara, V. Silva, K. Ocaña, L. Murta, V. Braganholo, and M. Mattoso, "Deriving scientific workflows from algebraic experiment lines: A practical approach," *Future Generation Computer Systems*, vol. 68, pp. 111–127, 2017. doi: 10.1016/j.future.2016.08.016.

Harbinger Enables Detector Evaluation and Benchmarking

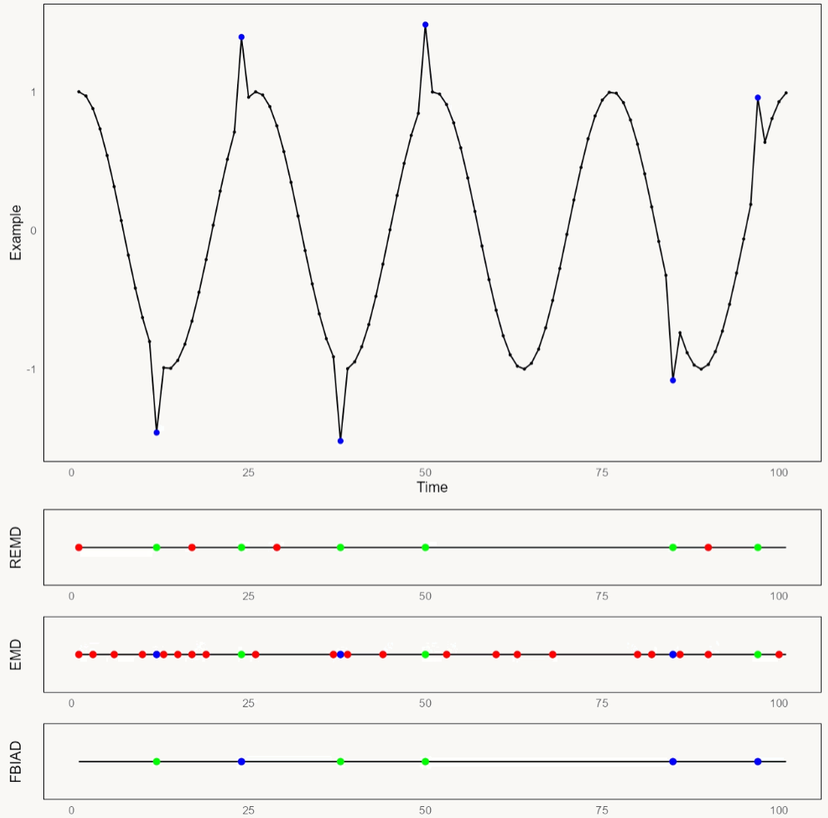
A fundamental challenge in time-series event detection is assessing the relative performance of different algorithms. Harbinger provides comprehensive evaluation capabilities, enabling systematic benchmarking across diverse datasets and detection scenarios.

The framework implements standardized evaluation metrics that quantify detection accuracy, precision, recall, and temporal alignment. These metrics allow researchers to objectively compare algorithms and identify the most appropriate methods for specific application contexts.

Beyond traditional accuracy measures, Harbinger supports advanced evaluation techniques that account for the temporal nature of event detection. This includes metrics that tolerate minor timing discrepancies while penalizing fragmented or inconsistent detection patterns.

The visualization capabilities enable intuitive interpretation of detector performance, facilitating rapid iteration and refinement of detection strategies.


REMD: A Novel Hybrid Anomaly Detection Method Based on EMD and ARIMA



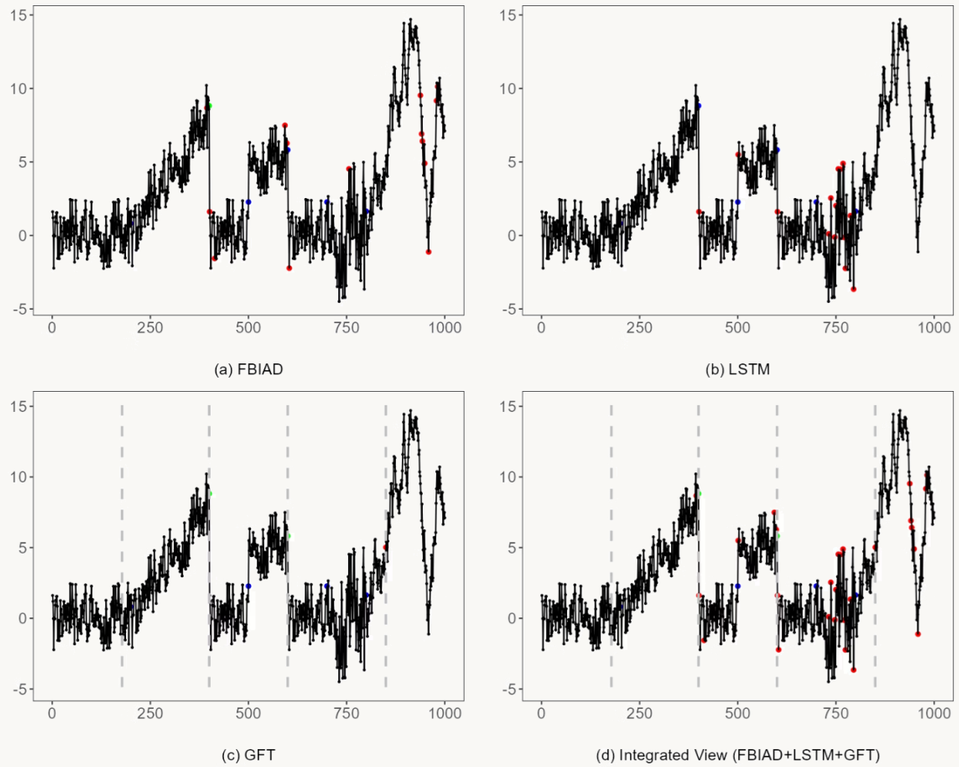
Precision & Recall
Quantify detection accuracy

Temporal Alignment
Measure timing accuracy

Performance Metrics
Comprehensive evaluation

Reference: J. Souza, E. Paixão, F. Fraga, L. Baroni, R. F. S. Alves, K. Belloze, J. Dos Santos, E. Bezerra, F. Porto, and E. Ogasawara, "REMD: A Novel Hybrid Anomaly Detection Method Based on EMD and ARIMA," *Proceedings of the International Joint Conference on Neural Networks*, 2024. doi: 10.1109/IJCNN60899.2024.10651192.

Combined Visualization of Detectors in Time Series



Effective analysis requires the ability to visualize detection results alongside the original time-series data. Harbinger provides sophisticated visualization tools that overlay multiple detector outputs, enabling direct comparison of how different algorithms identify events within the same temporal sequence.

These visualizations serve multiple purposes: they facilitate rapid assessment of detector behavior, reveal complementary detection patterns, and highlight potential issues such as false positives or missed events. The ability to compare multiple detectors simultaneously accelerates the process of selecting optimal methods for specific analytical tasks.

Multi-Detector Overlay

Visualize outputs from multiple algorithms simultaneously to identify complementary patterns and consensus events

Interactive Exploration

Examine detection details at various temporal resolutions to understand algorithm behavior

Ground Truth Comparison

Overlay known events with detected events to visually assess detection accuracy

Reference: E. Ogasawara *et al.*, "harbinger: A Unified Time Series Event Detection Framework." Available at: https://cran.r-project.org/web/packages/harbinger/index.html

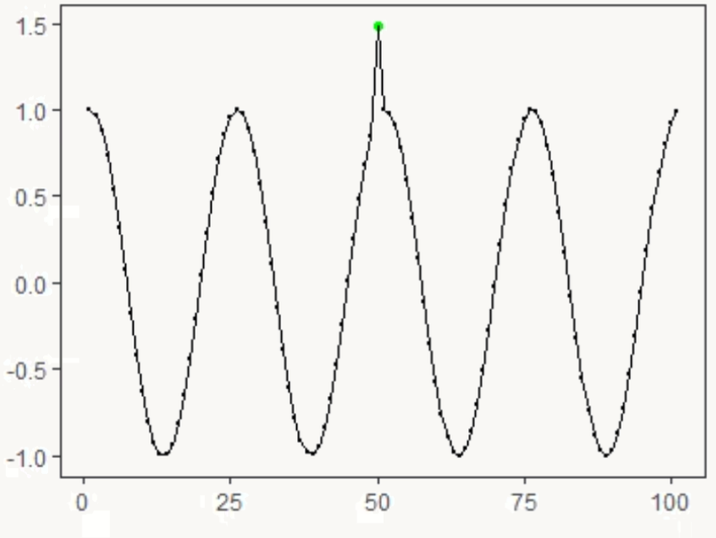
Simple Anomaly Detection Example

Harbinger's intuitive API enables rapid implementation of anomaly detection workflows. The framework includes curated example datasets with labeled events, providing immediate opportunities for experimentation and learning.

Getting Started

The following code demonstrates a complete anomaly detection workflow using the ARIMA-based detector. This simple example illustrates the core pattern: load data, instantiate a detector, fit the model to training data, and apply it to identify anomalous events.

```
library(daltoolbox)
library(harbinger)
data(examples_anomalies)
dataset <- examples_anomalies$simple
model <- hanr_arima()
# Fitting the model
model <- fit(model, dataset$serie)
# Making detections
detection <- detect(model, dataset$serie)
# Plotting the results
har_plot(model(), dataset$serie, detection)
```



The visualization clearly shows the original time series with detected anomalies highlighted, providing immediate feedback on detector performance.

01
Load Libraries and Data
Import Harbinger framework and example datasets

02
Instantiate Detector
Create ARIMA-based anomaly detector object

03
Fit Model
Train detector on time-series data

04
Detect Events
Apply model to identify anomalies

05
Visualize Results
Plot detections for analysis

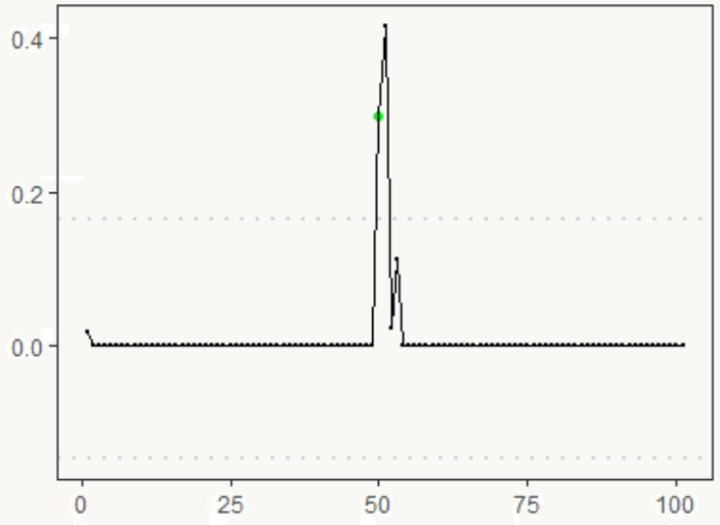
Understanding the Detection Results

Beyond simply identifying events, Harbinger provides comprehensive tools for understanding detector behavior and validating results. This includes examining detection outputs, visualizing model residuals, and computing quantitative evaluation metrics against ground truth labels.

Detailed Analysis

The detection object contains rich information beyond binary event flags. Researchers can filter and examine specific detections, explore the residual patterns that triggered alerts, and assess detection quality using confusion matrices and related metrics.

```
# Filtering detected events
print(detection |>   dplyr::filter(event == TRUE))
##   idx event    type
## 1  50  TRUE anomaly
# Understanding detected events
har_plot(model,   attr(detection, "res"),   detection,
           dataset$event,   yline = attr(detection, "threshold"))
# Evaluating detected events
evaluation <- evaluate(model,   detection$event,   dataset$event)
print(evaluation$confMatrix)
##           event
## detection TRUE  FALSE
## TRUE      1     0
## FALSE     0     100
```



Residual visualization reveals the underlying patterns that distinguish normal behavior from anomalous events, with threshold lines indicating detection boundaries.

Detection Filtering

Extract and examine specific events identified by the detector, including temporal indices and event classifications

Residual Analysis

Visualize model residuals to understand what deviations triggered anomaly flags and assess threshold appropriateness

Performance Evaluation

Compute confusion matrices and related metrics to quantify detection accuracy against labeled ground truth

This example demonstrates perfect detection performance with 100% accuracy—one true positive and zero false positives or false negatives. Such results validate the detector's suitability for this particular time series.

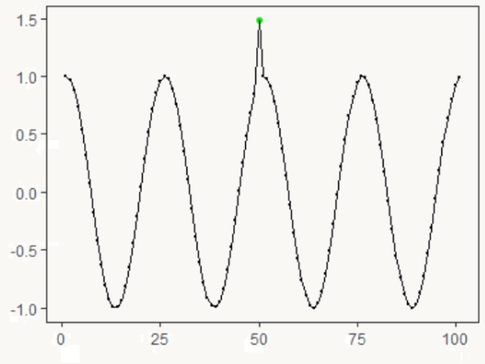
Ensemble Models

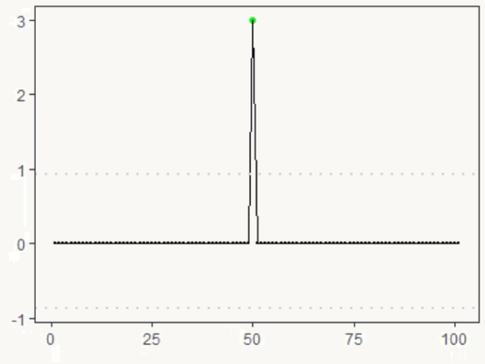
Individual detection algorithms often exhibit complementary strengths and weaknesses. Ensemble methods combine multiple detectors to achieve more robust and reliable event identification than any single algorithm can provide alone.

Combining Multiple Detectors

Harbinger simplifies ensemble construction through its consistent API. The following example creates an ensemble combining three distinct detection approaches: FBIAD (frequency-based), ARIMA (statistical modeling), and EMD (signal decomposition).

```
model <- har_ensemble(  hanr_fbiad(),   hanr_arima(),   hanr_emd())
model <- fit(model, dataset$serie)
detection <- detect(model, dataset$serie)
har_plot(model,   dataset$serie,   detection,   dataset$event)
har_plot(model,   attr(detection, "res"),   detection,
         dataset$event,   yline = attr(detection, "threshold"))
```





By aggregating decisions from multiple algorithms, ensembles reduce vulnerability to individual detector biases and improve overall detection reliability across diverse event types.

FBIAD
Frequency-based detection

ARIMA
Statistical modeling

Ensemble Aggregation
Combined decision

EMD
Signal decomposition

Fuzzifying Events: Beyond Binary Detection

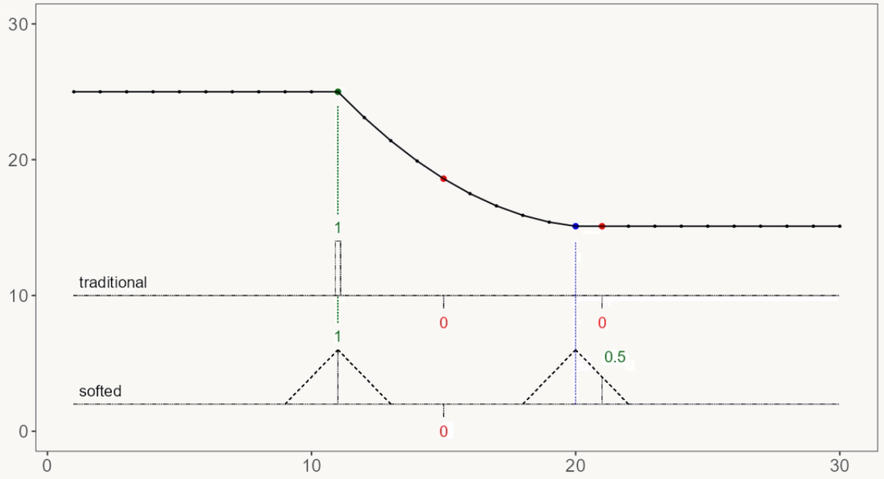
Traditional evaluation metrics treat event detection as a binary classification problem, assigning strict boundaries between correct and incorrect detections. However, this approach fails to capture the nuanced reality of temporal event detection, where minor timing discrepancies may be acceptable but fragmented or inconsistent patterns should be penalized.

SoftED Metrics

SoftED (Soft Event Detection) introduces fuzzy evaluation metrics that better reflect human judgment about detection quality. Unlike traditional metrics that penalize any temporal misalignment equally, SoftED tolerates minor offsets while imposing stronger penalties on fragmented detections.

This approach acknowledges that detecting an event one or two time steps early or late may be practically acceptable, whereas breaking a single event into multiple fragments or missing substantial portions represents more serious detection failures.

The fuzzy membership functions used in SoftED provide a graduated scale of detection quality rather than binary correct/incorrect classifications, enabling more nuanced evaluation of detector performance.



Temporal Tolerance
Accepts minor timing misalignments

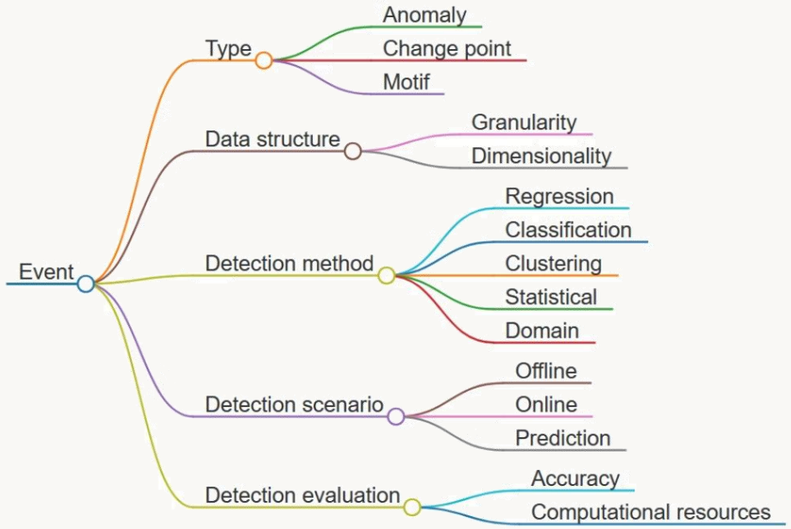
Fragmentation Penalty
Penalizes broken detections

Graduated Scoring
Provides nuanced quality assessment

Reference: R. Salles, J. Lima, M. Reis, R. Coutinho, E. Pacitti, F. Masseglia, R. Akbarinia, C. Chen, J. Garibaldi, F. Porto, and E. Ogasawara, "SoftED: Metrics for soft evaluation of time series event detection," *Computers and Industrial Engineering*, vol. 198, 2024. doi: 10.1016/j.cie.2024.110728.

Taxonomy of Event Detection in Time Series

The field of time-series event detection encompasses a rich diversity of methods, each designed to identify specific types of temporal patterns. Understanding this taxonomy helps researchers select appropriate techniques for their analytical objectives and domain contexts.



The taxonomy organizes detection methods by the types of events they target—anomalies, change points, and motifs—as well as by their underlying algorithmic approaches. This organizational framework facilitates systematic exploration of the detection landscape and informed method selection.

Anomaly Detection

Identifies observations that deviate significantly from expected behavior patterns, including outliers, novelties, and contextual anomalies.

Change Point Detection

Locates moments when the statistical properties of a time series undergo structural shifts, marking transitions between distinct regimes.

Motif Discovery

Finds recurring patterns or subsequences within time series, revealing repeated behaviors and temporal regularities.

Statistical Methods

Leverage probabilistic models and distributional assumptions to characterize normal behavior

Machine Learning Approaches

Apply supervised, unsupervised, or semi-supervised learning to discover patterns from data

Signal Processing Techniques

Use frequency domain analysis, filtering, and decomposition to isolate event signatures

Hybrid Methods

Combine multiple paradigms to leverage complementary strengths and improve robustness

Explore the comprehensive taxonomy and related resources at https://eic.cefet-rj.br/~eogasawara/en/event-detection-in-time-series

Thank You

Questions?

We welcome your questions, feedback, and collaboration opportunities. Harbinger is an open-source project that benefits from community contributions and diverse use cases.

Whether you're interested in applying Harbinger to your research, contributing new detection methods, or exploring advanced features, we encourage you to engage with the project.

CRAN Package

cran.r-project.org/web/packages/harbinger

Documentation

Event Detection Resources

DAL Toolbox

cran.r-project.org/web/packages/daltoolbox

Harbinger represents a collaborative effort to advance time-series event detection through open science and reproducible research. We look forward to seeing how you apply these tools in your work.