



Harbinger: Framework Unificado para Detecção de Eventos em Séries Temporais

Uma plataforma abrangente que consolida múltiplas metodologias de detecção sob uma arquitetura coerente, permitindo análise sistemática de padrões temporais.

Repositório GitHub

Pacote CRAN

Eduardo Ogasawara

eduardo.ogasawara@cefet-rj.br

<https://eic.cefet-rj.br/~eogasawara>

Uma Solução Integrada

O framework preenche uma lacuna crítica na análise de séries temporais, oferecendo um ambiente integrado para detecção de anomalias, identificação de pontos de mudança e descoberta de motivos.

Disponível como pacote R de código aberto, facilita pesquisas reproduzíveis e aplicações práticas em diversos domínios.



Harbinger

Stars 22 downloads 18K/month

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.

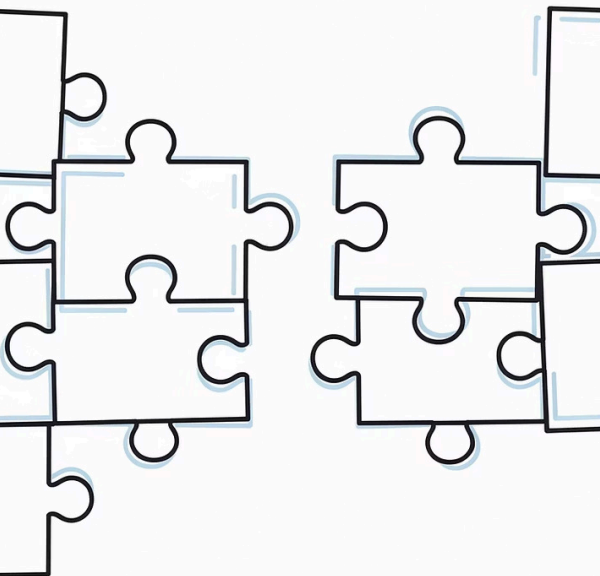
For anomaly detection, methods are based on: - Machine learning model deviation: Conv1D, ELM, MLP, LSTM, Random Regression Forest, and SVM - Classification models: Decision Tree, KNN, MLP, Naive Bayes, Random Forest, and SVM - Clustering: k-means and DTW - Statistical techniques: ARIMA, FBIAD, GARCH

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.



Por Que um Framework Unificado?

Fragmentação

Métodos especializados criam desafios práticos com ferramentas díspares e interfaces inconsistentes.

Necessidades Reais

Aplicações exigem comparação sistemática, combinação de detectores e avaliação rigorosa.

Solução Harbinger

Ecossistema coerente onde métodos podem ser aplicados, avaliados e combinados consistentemente.

50+

Métodos de Detecção

Cobertura abrangente de algoritmos

10

Técnicas de Change Point

Abordagens para quebras estruturais

9

Métodos de Descoberta de Motifs

Algoritmos de reconhecimento de padrões

Arquitetura Modular

Princípios de Design

O framework implementa uma interface rígida baseada em álgebra de workflow e linhas de experimento, garantindo consistência entre diferentes métodos de detecção.

Construído sobre a base do DAL Toolbox, adota padrões de design inspirados na API do Scikit-learn. As funções familiares `fit()` e `detect()` fornecem uma interface intuitiva.

01

Normalize

Pré-processamento e padronização

02

Filter

Redução de ruído e melhoria do sinal

03

Fit

Treinamento do modelo e estimativa de parâmetros

04

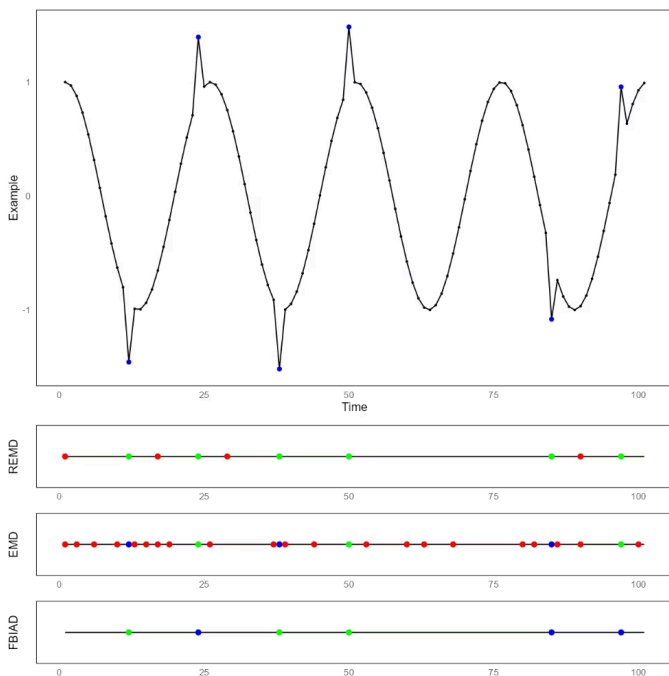
Detect

Identificação e classificação de eventos

Avaliação e Benchmarking de Detectores

O Harbinger fornece capacidades abrangentes de avaliação, permitindo benchmarking sistemático em diversos conjuntos de dados e cenários de detecção.

Implementa métricas padronizadas que quantificam precisão, recall e alinhamento temporal, permitindo comparação objetiva de algoritmos.



Precisão & Recall

Quantificam a acurácia da detecção

Alinhamento Temporal

Medem a precisão do timing

Métricas de Desempenho

Avaliação abrangente

Visualização Combinada de Detectores

Ferramentas sofisticadas de visualização que sobrepõem múltiplas saídas de detectores, permitindo comparação direta de como diferentes algoritmos identificam eventos na mesma sequência temporal.



Sobreposição Multi-Detector

Visualize saídas de múltiplos algoritmos simultaneamente para identificar padrões complementares e eventos de consenso.



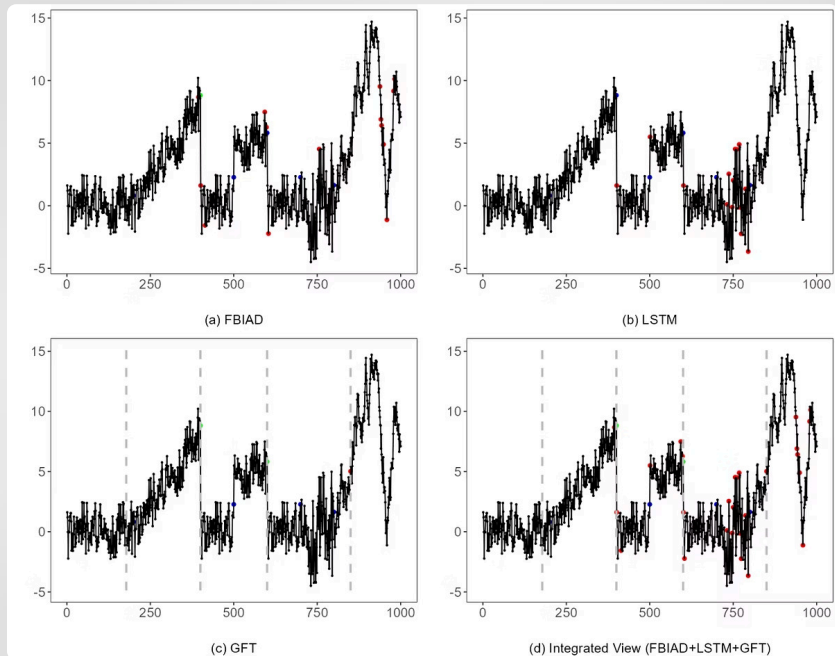
Exploração Interativa

Examine detalhes de detecção em várias resoluções temporais para entender o comportamento do algoritmo.



Comparação com Ground Truth

Sobreponha eventos conhecidos com eventos detectados para avaliar visualmente a precisão da detecção.



Exemplo Simples de Detecção de Anomalias

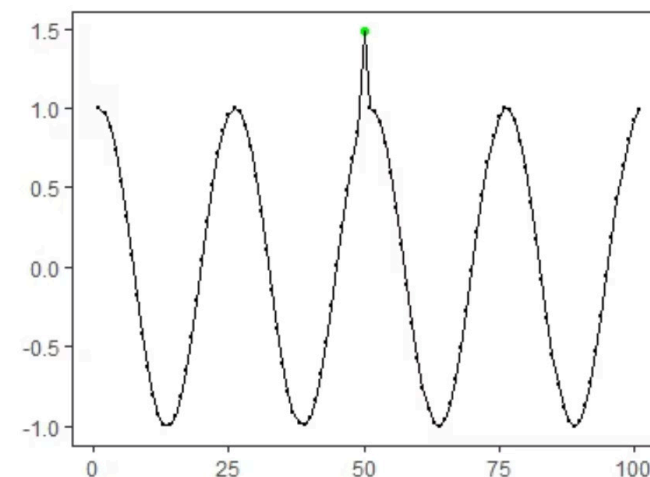
A API intuitiva do Harbinger permite implementação rápida de workflows de detecção de anomalias com datasets de exemplo curados.

```
library(daltoolbox)
library(harbinger)
data(examples_anomalies)
dataset <- examples_anomalies$simple
model <- hanr_arima()

# Ajustando o modelo
model <- fit(model, dataset$serie)

# Fazendo detecções
detection <- detect(model, dataset$serie)

# Plotando os resultados
har_plot(model(), dataset$serie, detection)
```



Carregar Bibliotecas

Importar framework e datasets

Instanciar Detector

Criar objeto detector ARIMA

Ajustar Modelo

Treinar detector nos dados

Detectar Eventos

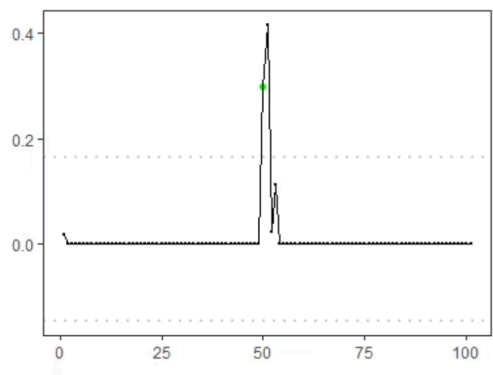
Aplicar modelo para identificar anomalias

Visualizar

Plotar detecções para análise

Compreendendo os Resultados da Detecção

O objeto de detecção contém informações ricas além de flags binários de eventos. Pesquisadores podem filtrar detecções específicas, explorar padrões residuais e avaliar qualidade usando matrizes de confusão.



```
# Filtrando eventos detectados
print(detection |>
  dplyr::filter(event == TRUE))
##   idx event  type
## 1  50  TRUE anomaly
```

```
# Avaliando eventos detectados
evaluation <- evaluate(model,
  detection$event,
  dataset$event)
print(evaluation$confMatrix)
##           event
## detection TRUE FALSE
## TRUE      1    0
## FALSE     0   100
```



Filtragem de Detecção

Extraia e examine eventos específicos identificados pelo detector



Análise Residual

Visualize resíduos do modelo para entender desvios que acionaram flags

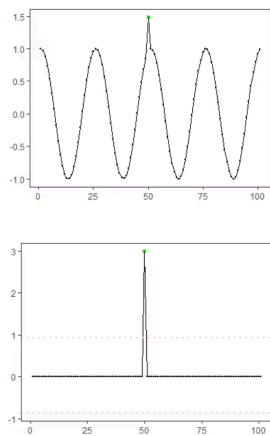


Avaliação de Performance

Compute matrizes de confusão para quantificar precisão contra ground truth

❏ Este exemplo demonstra performance de detecção perfeita com 100% de acurácia—um verdadeiro positivo e zero falsos positivos ou negativos.

Modelos Ensemble



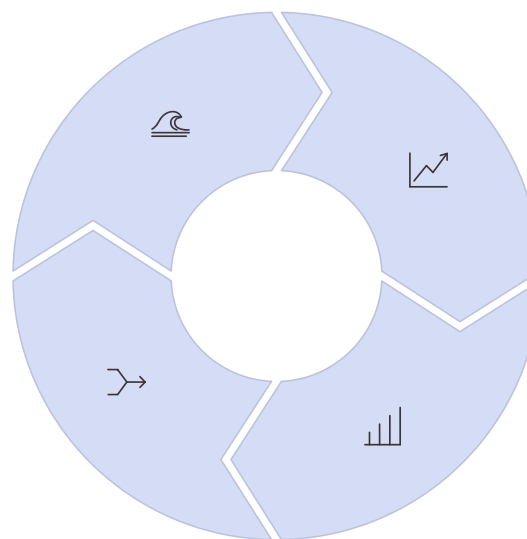
Combinando Múltiplos Detectores

Algoritmos individuais frequentemente exibem forças e fraquezas complementares. Métodos ensemble combinam múltiplos detectores para alcançar identificação de eventos mais robusta.

```
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)
```

FBIAD
Detecção baseada em frequência

Agregação Ensemble
Decisão combinada



ARIMA
Modelagem estatística

EMD
Decomposição de sinal

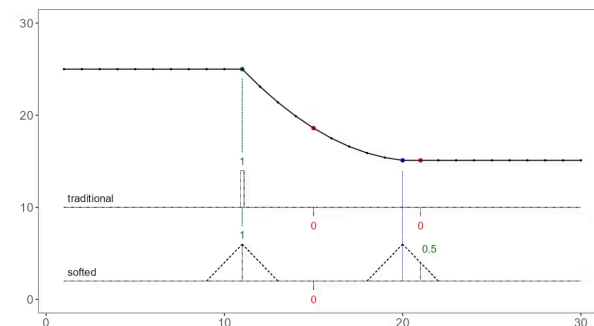
Fuzzificando Eventos: Além da Detecção Binária

Métricas SoftED

SoftED (Soft Event Detection) introduz métricas de avaliação fuzzy que refletem melhor o julgamento humano sobre qualidade de detecção.

Ao contrário de métricas tradicionais que penalizam qualquer desalinhamento temporal igualmente, SoftED tolera pequenos offsets enquanto impõe penalidades mais fortes em detecções fragmentadas.

As funções de pertinência fuzzy fornecem uma escala graduada de qualidade de detecção, permitindo avaliação mais nuançada da performance do detector.



1

Tolerância Temporal

Aceita pequenos desalinhamentos de timing

2

Penalidade de Fragmentação

Penaliza detecções quebradas

3

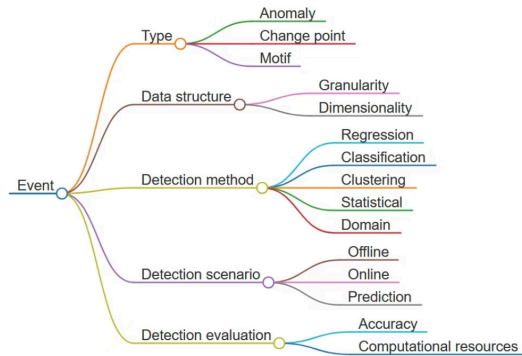
Pontuação Graduada

Fornece avaliação de qualidade nuançada

Taxonomia de Detecção de Eventos em Séries Temporais

O campo de detecção de eventos em séries temporais abrange uma rica diversidade de métodos, cada um projetado para identificar tipos específicos de padrões temporais.

A taxonomia organiza métodos de detecção pelos tipos de eventos que visam e por suas abordagens algorítmicas subjacentes.



Detecção de Anomalias

Identifica observações que desviam significativamente de padrões de comportamento esperados.

Detecção de Change Points

Localiza momentos quando propriedades estatísticas sofrem mudanças estruturais.

Descoberta de Motifs

Encontra padrões recorrentes ou subsequências dentro de séries temporais.



Métodos Estatísticos

Aproveitam modelos probabilísticos para caracterizar comportamento normal



Abordagens de Machine Learning

Aplicam aprendizado supervisionado ou não supervisionado para descobrir padrões



Técnicas de Processamento de Sinais

Usam análise de domínio de frequência e decomposição para isolar assinaturas de eventos



Métodos Híbridos

Combinam múltiplos paradigmas para aproveitar forças complementares



Perguntas?

Recebemos suas perguntas, feedback e oportunidades de colaboração. Harbinger é um projeto de código aberto que se beneficia de contribuições da comunidade e casos de uso diversos.



Pacote CRAN

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



Recursos de Detecção de Eventos

eic.cefet-rj.br/~eogasawara/en/event-detection-in-time-series



DAL Toolbox

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

Harbinger representa um esforço colaborativo para avançar a detecção de eventos em séries temporais através de ciência aberta e pesquisa reproduzível. Esperamos ver como você aplicará essas ferramentas em seu trabalho.