# Overview

This Rmd demonstrates anomaly detection using an Extreme Learning Machine regressor (hanr\_ml + ts\_elm). The model predicts the next value and flags anomalies when residuals exceed a learned threshold. Steps: load packages/data, visualize, define and fit the model, detect, evaluate, and plot results and residuals.

# Install Harbinger (only once, if needed)  
#install.packages("harbinger")

# Load required packages  
library(daltoolbox)  
library(harbinger)   
library(tspredit)

# Load example datasets bundled with harbinger  
data(examples\_anomalies)

# Select a simple synthetic time series with labeled anomalies  
dataset <- examples\_anomalies$simple  
head(dataset)

## serie event  
## 1 1.0000000 FALSE  
## 2 0.9689124 FALSE  
## 3 0.8775826 FALSE  
## 4 0.7316889 FALSE  
## 5 0.5403023 FALSE  
## 6 0.3153224 FALSE

# Plot the time series  
har\_plot(harbinger(), dataset$serie)



# Define ELM-based regressor for anomaly detection (hanr\_ml + ts\_elm)  
# - input\_size controls the window length; nhid/actfun tune the hidden layer  
 model <- hanr\_ml(ts\_elm(ts\_norm\_gminmax(), input\_size=4, nhid=3, actfun="purelin"))

# Fit the model  
 model <- fit(model, dataset$serie)

# Detect anomalies (compute residuals and events)  
 detection <- detect(model, dataset$serie)

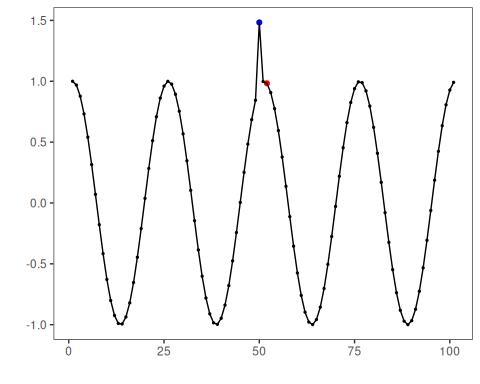
# Show only timestamps flagged as events  
 print(detection |> dplyr::filter(event==TRUE))

## idx event type  
## 1 52 TRUE anomaly

# Evaluate detections against ground-truth labels  
 evaluation <- evaluate(model, detection$event, dataset$event)  
 print(evaluation$confMatrix)

## event   
## detection TRUE FALSE  
## TRUE 0 1   
## FALSE 1 99

# Plot detections over the series  
 har\_plot(model, dataset$serie, detection, dataset$event)



# Plot residual scores and threshold  
 har\_plot(model, attr(detection, "res"), detection, dataset$event, yline = attr(detection, "threshold"))

