

Package: sensorLeak (via r-universe)

September 30, 2026

Title Leakage Detection for Environmental Sensor Data

Version 0.1.0

Description Provides tools for identifying potential information leakage and validation risks in machine learning workflows using environmental sensor data. The package includes checks for temporal ordering, shared sensors, spatial proximity, and overlapping temporal windows. The diagnostics are motivated by considerations of spatial and temporal structure in model validation (Roberts et al., 2017) <doi:10.1111/ecog.02881>.

License MIT + file LICENSE

Encoding UTF-8

Config/roxygen2/version 8.1.0

Depends R (>= 4.1.0)

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

URL <https://github.com/division55/sensorLeak>

BugReports <https://github.com/division55/sensorLeak/issues>

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Divya Umai Raja [aut, cre, cph], Sanjana Sujil [aut, cph]

Maintainer Divya Umai Raja <iamuayvid@gmail.com>

Repository <https://cran.r-universe.dev>

Date/Publication 2026-09-30 13:56:15 UTC

RemoteUrl <https://github.com/cran/sensorLeak>

RemoteRef HEAD

RemoteSha 861caba36fe705d21a2aacc8c3bb83ad27be9b1e

Contents

check_sensor_leakage	2
check_spatial_leakage	3
check_temporal_leakage	4
check_window_leakage	5
print.sensor_audit	6
print.sensor_audit_summary	6
sensor_audit	7
sensor_data	8
summary.sensor_audit	9
Index	10

check_sensor_leakage	<i>Check for sensors shared between training and testing data</i>
----------------------	---

Description

Identifies environmental sensors that occur in both training and testing observations. Shared sensors may represent a validation risk when the intended model should generalize to unseen sensors.

Usage

```
check_sensor_leakage(data, sensor, split)
```

Arguments

data	A data frame containing sensor observations.
sensor	A character string giving the column identifying the environmental sensor.
split	A character string giving the column identifying training and testing observations.

Value

A data frame containing sensors present in both splits, together with their observation counts.

Examples

```
data <- data.frame(
  sensor_id = c("S1", "S2", "S1", "S3"),
  split = c("train", "train", "test", "test")
)

check_sensor_leakage(
  data,
  sensor = "sensor_id",
  split = "split"
)
```

check_spatial_leakage *Check for spatial proximity between training and testing data*

Description

Identifies training and testing observations that are within a specified geographic distance of each other. Close observations may represent a validation risk when a model is intended to generalize to geographically independent locations.

Usage

```
check_spatial_leakage(data, lat, lon, split, threshold = 1)
```

Arguments

data	A data frame containing geographic observations.
lat	A character string giving the latitude column.
lon	A character string giving the longitude column.
split	A character string giving the column identifying training and testing observations.
threshold	A numeric value giving the maximum distance in kilometres for observations to be considered spatially close.

Value

A data frame containing pairs of spatially close training and testing observations.

Examples

```
data <- data.frame(  
  latitude = c(13.0827, 13.0830, 13.2000, 13.3000),  
  longitude = c(80.2707, 80.2710, 80.3000, 80.4000),  
  split = c("train", "test", "train", "test")  
)  
  
check_spatial_leakage(  
  data,  
  lat = "latitude",  
  lon = "longitude",  
  split = "split",  
  threshold = 1  
)
```

`check_temporal_leakage`*Check for temporal leakage*

Description

Identifies test observations that occur before the end of the training period. Such observations may indicate an invalid temporal validation setup.

Usage

```
check_temporal_leakage(data, time, split)
```

Arguments

<code>data</code>	A data frame containing timestamped observations.
<code>time</code>	A character string giving the timestamp column.
<code>split</code>	A character string giving the column identifying training and testing observations.

Value

A data frame containing test observations that occur before the end of the training period.

Examples

```
data <- data.frame(  
  timestamp = as.POSIXct(  
    c(  
      "2026-01-01 10:00:00",  
      "2026-01-01 12:00:00",  
      "2026-01-01 11:00:00",  
      "2026-01-01 15:00:00"  
    )  
  ),  
  split = c("train", "train", "test", "test")  
)  
  
check_temporal_leakage(  
  data,  
  time = "timestamp",  
  split = "split"  
)
```

check_window_leakage	<i>Check for overlapping temporal windows</i>
----------------------	---

Description

Detects temporal overlap between training and testing windows within environmental sensor data.

Usage

```
check_window_leakage(data, start, end, split, sensor)
```

Arguments

data	A data frame containing temporal windows.
start	A character string giving the column containing window start times.
end	A character string giving the column containing window end times.
split	A character string giving the column identifying training and testing observations.
sensor	A character string giving the column identifying the environmental sensor.

Value

A data frame containing detected overlapping windows.

Examples

```
data <- data.frame(
  sensor_id = c("S1", "S1"),
  window_start = as.POSIXct(
    c("2026-01-01 10:00:00", "2026-01-01 13:00:00")
  ),
  window_end = as.POSIXct(
    c("2026-01-01 14:00:00", "2026-01-01 18:00:00")
  ),
  split = c("train", "test")
)

check_window_leakage(
  data,
  start = "window_start",
  end = "window_end",
  split = "split",
  sensor = "sensor_id"
)
```

print.sensor_audit *Print a sensor leakage audit*

Description

Prints a concise summary of the potential validation risks identified by a sensor_audit object.

Usage

```
## S3 method for class 'sensor_audit'  
print(x, ...)
```

Arguments

x An object of class sensor_audit.
... Additional arguments passed to the method.

Value

Invisibly returns the audit object.

print.sensor_audit_summary
 Print a sensor audit summary

Description

Print a sensor audit summary

Usage

```
## S3 method for class 'sensor_audit_summary'  
print(x, ...)
```

Arguments

x An object of class sensor_audit_summary.
... Additional arguments.

Value

Invisibly returns the summary object.

sensor_audit	<i>Audit environmental sensor data for leakage risks</i>
--------------	--

Description

Runs multiple leakage checks on environmental sensor data and combines their findings into a single audit result.

Usage

```
sensor_audit(data, time, sensor, split, lat, lon, spatial_threshold = 1)
```

Arguments

data	A data frame containing environmental sensor data.
time	A character string giving the timestamp column.
sensor	A character string giving the sensor identifier column.
split	A character string giving the train/test split column.
lat	A character string giving the latitude column.
lon	A character string giving the longitude column.
spatial_threshold	A numeric value giving the spatial proximity threshold in kilometres.

Value

An object of class `sensor_audit` containing the results of the leakage checks.

Examples

```
data <- data.frame(
  timestamp = as.POSIXct(c(
    "2026-01-01 10:00:00",
    "2026-01-01 12:00:00",
    "2026-01-01 14:00:00",
    "2026-01-01 16:00:00"
  )),
  sensor_id = c("S1", "S1", "S2", "S2"),
  latitude = c(13.08, 13.08, 13.20, 13.30),
  longitude = c(80.27, 80.27, 80.30, 80.40),
  split = c("train", "train", "test", "test")
)

audit <- sensor_audit(
  data,
  time = "timestamp",
  sensor = "sensor_id",
  split = "split",
```

```
    lat = "latitude",  
    lon = "longitude"  
  )
```

sensor_data

Example Environmental Sensor Dataset

Description

A small synthetic dataset representing environmental sensor observations with timestamps, sensor identifiers, geographic coordinates, PM2.5 measurements, and train/test assignments.

Usage

```
sensor_data
```

Format

A data frame with 12 observations and 6 variables:

timestamp Timestamp of the sensor observation.

sensor_id Identifier of the environmental sensor.

latitude Latitude of the sensor location in decimal degrees.

longitude Longitude of the sensor location in decimal degrees.

PM25 PM2.5 concentration measurement.

split Training or testing assignment.

Source

Synthetic example data created for the sensorLeak package.

Examples

```
data(sensor_data)  
head(sensor_data)
```

summary.sensor_audit	<i>Summarize a sensor leakage audit</i>
----------------------	---

Description

Provides detailed summary information for a sensor_audit object.

Usage

```
## S3 method for class 'sensor_audit'  
summary(object, ...)
```

Arguments

object	An object of class sensor_audit.
...	Additional arguments passed to the method.

Value

An object of class sensor_audit_summary.

Index

* **datasets**

sensor_data, [8](#)

check_sensor_leakage, [2](#)

check_spatial_leakage, [3](#)

check_temporal_leakage, [4](#)

check_window_leakage, [5](#)

print.sensor_audit, [6](#)

print.sensor_audit_summary, [6](#)

sensor_audit, [7](#)

sensor_data, [8](#)

summary.sensor_audit, [9](#)