Package 'WaveST'

March 16, 2026

Type Package

Title Wavelet-Based Spatial Time Series Models

Version 0.1.0

Maintainer Dr. Ranjit Kumar Paul <ranjitstat@gmail.com>

Description An integrated wavelet-based spatial time series modelling framework designed to enhance predictive accuracy under noisy and nonstationary conditions by jointly exploiting multiresolution (wavelet) information and spatial dependence. The package implements WaveSARIMA() (Wavelet Based Spatial AutoRegressive Integrated Moving Average model using regression features with forecast::auto.arima()) and WaveSNN() (Wavelet Based Spatial Neural Network model using neuralnet with hyperparameter search). Both functions support spatial transformation via a user-supplied spatial matrix, lag feature construction, MODWT-based wavelet sub-series feature generation, time-ordered train/test splitting, and performance evaluation (Root Mean Square Error (RMSE), Mean Absolute Error (MAE), R-squared (R^2), and Mean Absolute Percentage Error (MAPE)), returning fitted models and actual vs predicted values for train and test sets. The package has been developed using the algorithm of Paul et al. (2023) <doi:10.1007/s43538-025-00581-1>.

Encoding UTF-8

RoxygenNote 7.3.1

Imports forecast, stats, neuralnet, tsutils, wavelets

Suggests devtools, roxygen2, usethis

License GPL-3

NeedsCompilation no

Author Dr. Md Yeasin [aut],
Dr. Ranjit Kumar Paul [aut, cre],
Akarsh Kumar Singh [aut]

Repository CRAN

Date/Publication 2026-03-16 18:50:02 UTC

Contents

`WaveSARIMA` *WaveSARIMA: Wavelet based Spatial ARIMA Model*

Description

Builds wavelet sub-series features from lagged original data and spatially transformed data, then fits `forecast::auto.arima()` directly (no grid tuning). Uses wavelet features as regression variables (xreg) when they are available/valid.

Usage

```
WaveSARIMA(
  Data,
  Spatial_Matrix,
  Lag = 4,
  Wfilter = "d4",
  Wlevel = 4,
  Split = 0.8,
  y_cols = NULL,
  boundary = "periodic",
  fast = TRUE,
  seed = 123,
  verbose = TRUE
)
```

Arguments

`Data`	A numeric matrix or data.frame.
`Spatial_Matrix`	Square numeric matrix used for spatial transform (Bs).
`Lag`	Integer. Maximum lag used in `tsutils::lagmatrix()`.
`Wfilter`	Character. Wavelet filter name passed to `wavelets::modwt()`.
`Wlevel`	Integer. Wavelet decomposition level.
`Split`	Numeric in (0,1). Train ratio for time-order split.
`y_cols`	Integer vector. Target column indices in `Data`. Default = 1:k.
`boundary`	Character. Boundary for MODWT, default "periodic".
`fast`	Logical. Passed to MODWT; default TRUE.
`seed`	Integer. Random seed.
`verbose`	Logical. If TRUE, prints accuracy table.

Value

A list with:

- `model_table`: selected ARIMA orders + AICc + number of regressors used
- `accuracy_table`: train/test metrics per target
- `best_models`: list of fitted auto.arima model objects per target
- `predictions`: list of train/test data.frames (Actual vs Predicted) per target
- `error_log`: errors if any target fails

References

Paul, R.K., Yeasin, M. and Manjunatha, B. (2025). WaveDeepCrop: wavelet enhanced deep learning framework for noise-resilient crop yield forecasting. Proceedings of the Indian National Science Academy, 91(4), 1310–1323.

Examples

```
Data <- matrix(rnorm(300), ncol = 3)
Bs <- matrix(c(0,1,0,
               0.5,0,0.5,
               0,1,0), nrow = 3, byrow = TRUE)
res <- WaveSARIMA(
  Data = Data,
  Spatial_Matrix = Bs,
  Lag = 4,
  Wfilter = "d4",
  Wlevel = 4,
  Split = 0.8,
  verbose = FALSE
)
res$accuracy_table
```

`WaveSNN` *WaveSNN: Wavelet based Spatial ANN Model*

Description

Builds wavelet sub-series features from lagged original data and spatially transformed data, then fits and hyper-tunes a feedforward neural network (via `neuralnet`) for one or multiple target columns.

Usage

```
WaveSNN(
  Data,
  Spatial_Matrix,
  Lag = 4,
```

```
  Wfilter = "d4",
  Wlevel = 4,
  Split = 0.8,
  hidden_grid = list(c(5), c(10), c(15), c(10, 5), c(15, 10)),
  threshold_grid = c(0.01, 0.001),
  stepmax_grid = c(1e+05, 2e+05),
  algorithm_grid = c("rprop+"),
  y_cols = NULL,
  boundary = "periodic",
  fast = TRUE,
  seed = 123,
  verbose = TRUE
)
```

Arguments

`Data`	A numeric matrix or data.frame. Non-numeric columns should be removed beforehand.
`Spatial_Matrix`	Square numeric matrix used for spatial transform (Bs). The function uses the first k columns of `Data` where k = `min(ncol(Data), nrow(Spatial_Matrix), ncol(Spatial_Matrix))`.
`Lag`	Integer. Maximum lag used in `tsutils::lagmatrix()`.
`Wfilter`	Character. Wavelet filter name passed to `wavelets::modwt()` (e.g., "d4", "la8").
`Wlevel`	Integer. Wavelet decomposition level.
`Split`	Numeric in (0,1). Train ratio for time-order split.
`hidden_grid`	List of integer vectors. Hidden layer sizes to try, e.g. `list(c(5), c(10), c(10,5))`.
`threshold_grid`	Numeric vector. `neuralnet` convergence threshold values.
`stepmax_grid`	Numeric vector. `neuralnet` stepmax values.
`algorithm_grid`	Character vector. `neuralnet` algorithms, e.g. "rprop+".
`y_cols`	Integer vector. Target column indices in `Data`. Default = 1:k.
`boundary`	Character. Boundary for MODWT, default "periodic".
`fast`	Logical. Passed to MODWT; default TRUE.
`seed`	Integer. Random seed.
`verbose`	Logical. If TRUE, prints accuracy table.

Value

A list with:

- `hypertune_all`: all hyperparameter runs for all targets
- `best_hyper_table`: best hyperparameters per target
- `accuracy_table`: best train/test metrics per target
- `best_models`: list of fitted `neuralnet` objects per target
- `predictions`: list of train/test data.frames (Actual vs Predicted) per target

References

Paul, R.K., Yeasin, M. and Manjunatha, B. (2025). WaveDeepCrop: wavelet enhanced deep learning framework for noise-resilient crop yield forecasting. Proceedings of the Indian National Science Academy, 91(4), 1310–1323.

Examples

```
Data <- matrix(rnorm(300), ncol = 3)
Bs <- matrix(c(0,1,0,
               0.5,0,0.5,
               0,1,0), nrow = 3, byrow = TRUE)

res <- WaveSNN(
  Data = Data,
  Spatial_Matrix = Bs,
  Lag = 4,
  Wfilter = "d4",
  Wlevel = 4,
  Split = 0.8,
  hidden_grid = list(c(5)),        # minimal to keep example fast
  threshold_grid = c(0.01),
  stepmax_grid = c(1e5),
  algorithm_grid = c("rprop+"),
  verbose = FALSE
)
res$accuracy_table
```

Index