Seasonality Detection Methods: A Comparative Study

Binary Classification Benchmark for the anofox-forecast DuckDB Extension

anofox-forecast benchmark suite

2026-01-08

Table of contents

Executive Summary

This benchmark evaluates seasonality detection methods as a **binary classification problem**: given a time series, does it contain seasonality? We simulate 550 curves with varying seasonal strength levels (0.0 to 1.0) and evaluate 13 detection methods using classification metrics

(Accuracy, Precision, Recall, F1, ROC AUC, PR AUC). Ground truth is defined as seasonal if simulated strength $>= 0.2$.

Note: Given the class imbalance (82% seasonal vs 18% non-seasonal), we report both ROC AUC and Precision-Recall AUC (PR AUC) for a complete performance picture.

This benchmark replicates the methodology from the fdars R package benchmark.

Quick Start: Which Method Should I Use?

Your Situation	Recommended Method	SQL Example
General purpose	Wavelet or Variance Strength	ts_seasonal_strength(values, period, 'wavelet')
Unknown period	Autoperiod	(ts_autoperiod(values)).detected
Fast screening	FFT	(ts_estimate_period_fft(values)).co
Trending data	CFD-Autoperiod	(ts_cfd_autoperiod(values)).detecte
Noisy data	ACF or Autoperiod	(ts_estimate_period_acf(values)).co
Irregular sampling	Lomb-Scargle	(ts_lomb_scargle(values)).false_ala
Need model fit	AIC	(ts_aic_period(values)).r_squared

Quick SQL Example

```sql
-- From long format data (one row per observation):
-- Aggregate values per series, then detect seasonality
SELECT
    series_id,
    (ts_autoperiod(LIST(value ORDER BY date))).detected AS has_seasonality,
    (ts_autoperiod(LIST(value ORDER BY date))).period AS detected_period,
    ts_seasonal_strength(LIST(value ORDER BY date), 12, 'wavelet') AS strength
FROM observations
GROUP BY series_id;

-- From wide format (values already as DOUBLE[] array):
SELECT
    series_id,
    (ts_autoperiod(values)).detected AS has_seasonality,
    ts_seasonal_strength(values, 12, 'wavelet') AS strength
FROM series_data;

-- Threshold: strength > 0.3 typically indicates seasonality
```

Key Results (TL;DR)

- **Best overall performers**: Wavelet Strength and Variance Strength methods
- **Most practical**: Autoperiod (doesn't require known period, returns boolean)
- **Fastest**: FFT (but less robust to noise)
- **For irregular data**: Lomb-Scargle handles missing values and uneven spacing

For detailed methodology and analysis, continue reading below.

Introduction

Detection as Binary Classification

Unlike period estimation (which asks "what is the period?"), **seasonality detection** asks a simpler question: **"is there seasonality?"** This is a binary classification problem where each method produces a confidence score, and we apply a threshold to make a detection decision.

Hypotheses

Based on method characteristics, we predict:

1. **Spectral methods** (FFT, Lomb-Scargle) will excel on clean, stationary data but degrade with autocorrelated noise (red noise)
2. **Differencing-based methods** (CFD-Autoperiod) will outperform others on trending data by removing non-stationarity
3. **Variance-based methods** will show the best overall robustness due to their non-parametric nature
4. **Pattern-matching methods** (Matrix Profile, ACF) will struggle with short series where insufficient cycles are present

Methods Evaluated

Method	SQL Function	Score Used	Description
AIC Comparison	`ts_aic_period`	R-squared	Fourier model fit quality
FFT Confidence	`ts_estimate_period_fft`	confidence	Peak-to-mean power ratio
ACF Confidence	`ts_estimate_period_acf`	confidence	Autocorrelation at lag

Method	SQL Function	Score Used	Description
Variance Strength	`ts_seasonal_strength(..., 'variance')`	strength	Seasonal variance ratio
Spectral Strength	`ts_seasonal_strength(..., 'spectral')`	strength	Power at seasonal frequency
Wavelet Strength	`ts_seasonal_strength(..., 'wavelet')`	strength	Morlet wavelet energy
SAZED	`ts_sazed_period`	SNR	Zero-padded spectral SNR
Autoperiod	`ts_autoperiod`	acf_validation	FFT+ACF hybrid validation
CFD-Autoperiod	`ts_cfd_autoperiod`	acf_validation	First-differenced FFT+ACF
Lomb-Scargle	`ts_lomb_scargle`	1-FAP	Statistical significance
Matrix Profile	`ts_matrix_profile_period`	confidence	Motif agreement ratio
STL	`ts_stl_period`	seasonal_strength	Decomposition strength
SSA	`ts_ssa_period`	variance_explained	Eigenvalue dominance

Detailed Method Descriptions

Spectral Methods

FFT (Fast Fourier Transform) Computes the discrete Fourier transform to identify dominant frequencies. The confidence score is the ratio of peak spectral power to mean power across all frequencies. Fast ($O(n \log n)$) but sensitive to noise and non-stationarity.

$$X[k] = \sum_{t=0}^{N-1} x[t] \cdot e^{-2\pi i k t/N}, \quad \text{Confidence} = \frac{P[k_{max}]}{\bar{P}}$$

Reference: Cooley, J.W. & Tukey, J.W. (1965). "An Algorithm for the Machine Calculation of Complex Fourier Series." Mathematics of Computation, 19(90), 297-301.

Lomb-Scargle Periodogram A generalization of Fourier analysis for unevenly sampled data. Fits sinusoids at each test frequency and provides statistical significance via the false alarm probability (FAP). Robust for irregular sampling.

$$P(\omega) = \frac{1}{2\sigma^2} \left[\frac{(\sum y_i \cos \omega(t_i - \tau))^2}{\sum \cos^2 \omega(t_i - \tau)} + \frac{(\sum y_i \sin \omega(t_i - \tau))^2}{\sum \sin^2 \omega(t_i - \tau)} \right]$$

References: Lomb, N.R. (1976). "Least-squares frequency analysis of unequally spaced data." Astrophysics and Space Science, 39, 447-462. Scargle, J.D. (1982). "Studies in astronomical time series analysis II." The Astrophysical Journal, 263, 835-853.

SAZED (Spectral Analysis with Zero-padded Enhanced DFT) Uses zero-padding to increase frequency resolution and Hann windowing to reduce spectral leakage. The signal-to-noise ratio (SNR) provides a confidence measure.

Reference: Ding, H., et al. (2008). "Querying and Mining of Time Series Data." VLDB Endowment, 1(2), 1542-1552.

Autocorrelation Methods

ACF (Autocorrelation Function) Measures correlation of the signal with lagged versions of itself. Peaks in the ACF indicate periodic structure. The confidence is the ACF value at the detected period lag.

$$\text{ACF}(k) = \frac{\sum_{t=1}^{n-k}(x_t - \mu)(x_{t+k} - \mu)}{\sum_{t=1}^{n}(x_t - \mu)^2}$$

Reference: Box, G.E.P. & Jenkins, G.M. (1976). Time Series Analysis: Forecasting and Control. Holden-Day.

Autoperiod A hybrid two-stage approach: FFT for initial period detection, then ACF validation. Combines spectral speed with time-domain robustness.

Reference: Vlachos, M., Yu, P., & Castelli, V. (2005). "On Periodicity Detection and Structural Periodic Similarity." SIAM International Conference on Data Mining.

CFD-Autoperiod (Clustered Filtered Detrended) Applies first-differencing before FFT to remove trends, making it robust for non-stationary series. Validates with ACF on the original series.

Reference: Elfeky, M.G., Aref, W.G., & Elmagarmid, A.K. (2005). "Periodicity Detection in Time Series Databases." IEEE TKDE, 17(7), 875-887.

Model-Based Methods

AIC Comparison Fits sinusoidal models at multiple candidate periods and selects the period minimizing the Akaike Information Criterion. Returns R^2 as a measure of model fit quality.

$$\text{AIC} = n \cdot \ln(\text{RSS}/n) + 2k, \quad R^2 = 1 - \frac{\text{RSS}}{\text{SS}_{total}}$$

Reference: Akaike, H. (1974). "A new look at the statistical model identification." IEEE Transactions on Automatic Control, 19(6), 716-723.

Decomposition Methods

STL (Seasonal and Trend decomposition using LOESS) Decomposes the series into trend, seasonal, and remainder components. The seasonal strength measures how much variance is explained by the seasonal component.

$$F_S = \max\left(0, 1 - \frac{\mathrm{Var}(R)}{\mathrm{Var}(S + R)}\right)$$

Reference: Cleveland, R.B., et al. (1990). "STL: A Seasonal-Trend Decomposition Procedure Based on Loess." Journal of Official Statistics, 6(1), 3-73.

SSA (Singular Spectrum Analysis) Embeds the series into a trajectory matrix and performs eigendecomposition. Periodic components appear as paired eigenvalues. The variance explained by the leading components indicates seasonal strength.

Reference: Golyandina, N., Nekrutkin, V., & Zhigljavsky, A. (2001). Analysis of Time Series Structure: SSA and Related Techniques. Chapman & Hall/CRC.

Strength-Based Methods

Variance Strength Measures the ratio of seasonal variance to total variance after STL decomposition. Values near 1 indicate strong seasonality.

Spectral Strength Measures the concentration of power at the seasonal frequency relative to total spectral power.

Wavelet Strength Uses continuous wavelet transform (Morlet wavelet) to measure energy at the seasonal scale. Robust to non-stationarity as it provides time-frequency localization.

Reference: Wang, X., Smith, K., & Hyndman, R. (2006). "Characteristic-based clustering for time series data." Data Mining and Knowledge Discovery, 13(3), 335-364.

Pattern-Based Methods

Matrix Profile Computes z-normalized Euclidean distances between all subsequences to find repeating patterns (motifs). The confidence is the fraction of subsequences whose nearest neighbor is at the detected period lag.

$$d(i, j) = \sqrt{\sum (z_i - z_j)^2}, \quad \text{Period} = \arg\max_k H[k]$$

References: Yeh, C.C.M., et al. (2016). "Matrix Profile I: All Pairs Similarity Joins for Time Series." IEEE ICDM. Yeh, C.C.M., et al. (2017). "Matrix Profile VI: Meaningful Multidimensional Motif Discovery." IEEE ICDM.

Ground Truth Definition

A series is classified as **seasonal** if its simulated seasonal strength $>= 0.2$. This threshold follows the fdars benchmark convention.

Setup

Connect to DuckDB and Load Extension

Baseline Simulation

Simulation Parameters

Following the fdars benchmark: - **11 strength levels**: 0.0, 0.1, 0.2, . . . , 1.0 - **50 curves per level**: 550 total curves - **60 observations**: 5 years of monthly data - **Period = 12**: Monthly seasonality - **White noise**: sigma = 0.3

Baseline Data Generation

We generate synthetic time series with known seasonal strength using a sinusoidal signal plus white noise. The amplitude is calibrated so that the signal-to-noise ratio corresponds to the target strength level: strength $= A^2/(A^2 + \sigma^2)$.

```
Generated 550 curves
```

```
Seasonal (strength >= 0.2): 450
```

```
Non-seasonal: 100
```

Strength Level Distribution

Curve Distribution by Strength Level
Ground truth: seasonal if strength >= 0.2
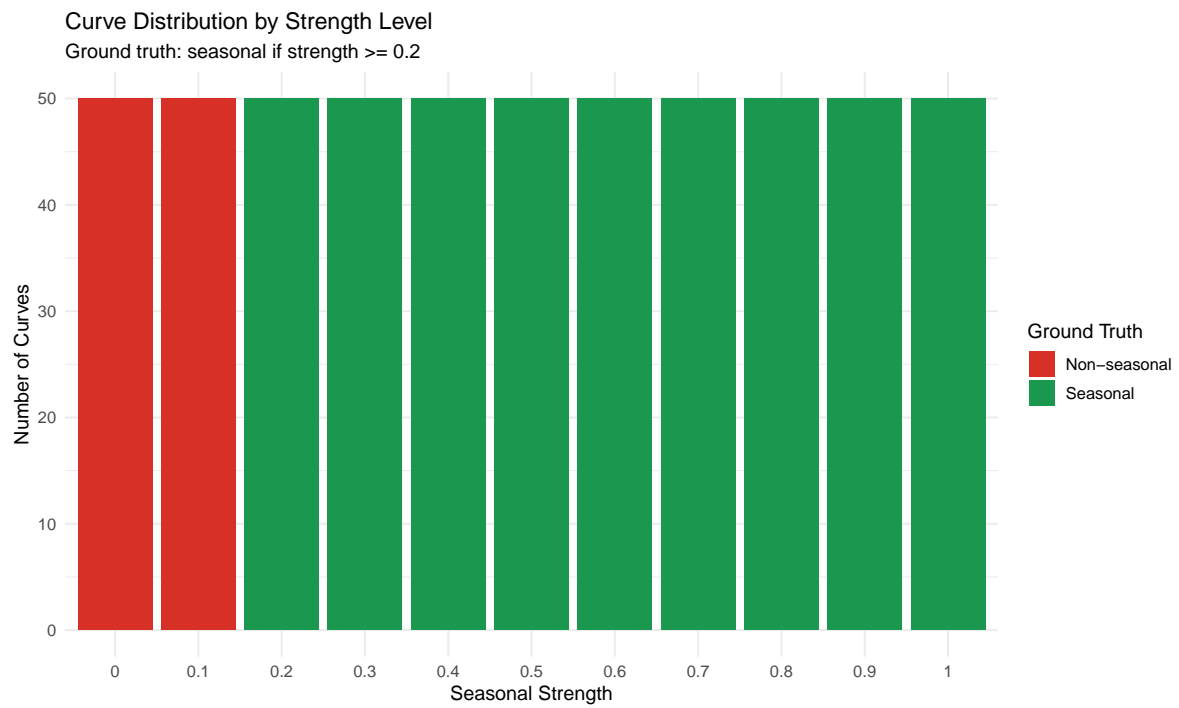
Figure 1: Distribution of curves by seasonal strength level

Example Curves
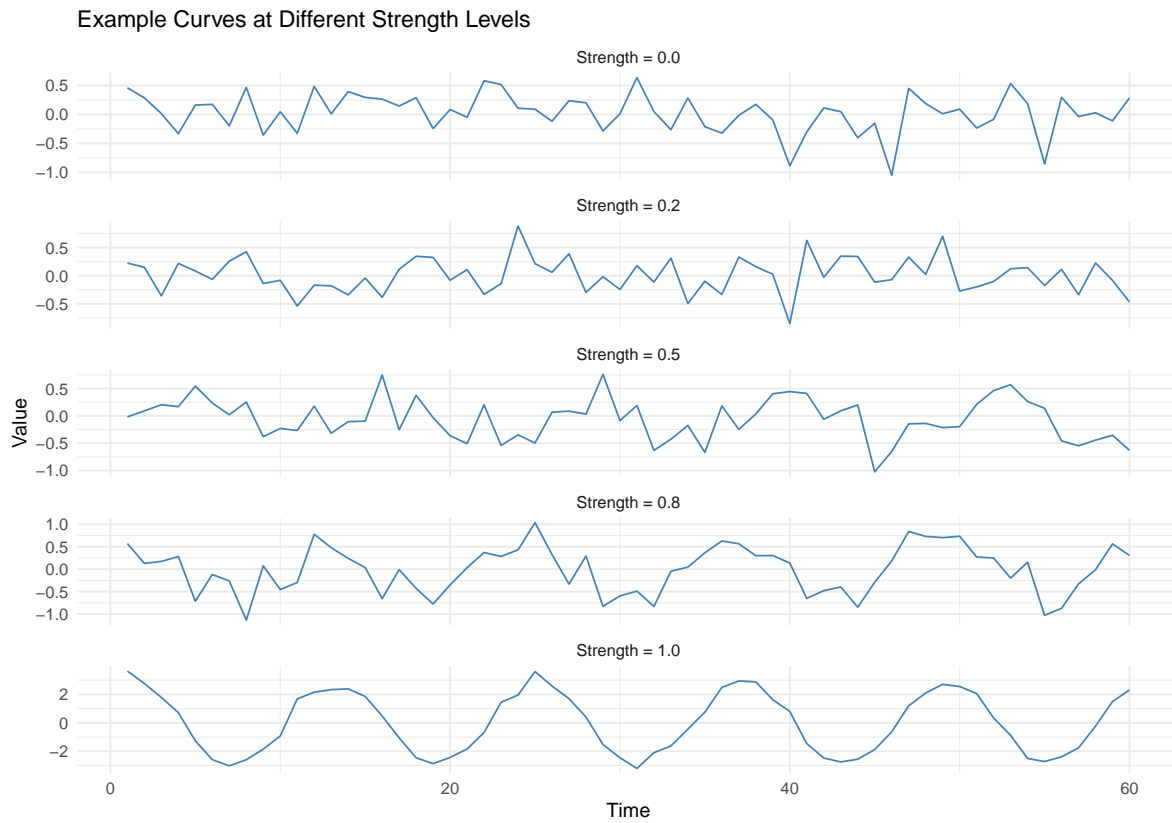
Figure 2: Example curves at different strength levels

Load Data into DuckDB

The simulated curves are loaded into a DuckDB table for analysis. Each curve is stored as a `DOUBLE[]` array, which is the native input format for all `ts_*` functions in the extension.

```
[1] 0
```

```
[1] 0
```

```
Data loaded into DuckDB
```

Method Evaluation

SQL API Usage

The following examples demonstrate how to use the seasonality detection methods. All `ts_*` functions expect a `DOUBLE[]` array as input.

Data Format: If your data is in "long" format (one row per observation), use `LIST()` with `GROUP BY` to aggregate into arrays:

```
-- Long format: one row per observation
-- +----------+------------+-------+
-- | series_id| date       | value |
-- +----------+------------+-------+
-- | A        | 2020-01-01 | 10.5  |
-- | A        | 2020-02-01 | 12.3  |
-- | B        | 2020-01-01 | 5.2   |
-- +----------+------------+-------+


-- Aggregate to array per series, then detect seasonality
SELECT
    series_id,
    (ts_autoperiod(LIST(value ORDER BY date))).detected AS has_seasonality,
    (ts_autoperiod(LIST(value ORDER BY date))).period AS detected_period
FROM long_format_data
GROUP BY series_id;
```

Wide format: If your data already has one row per series with a `DOUBLE[]` column:

```
-- Wide format: one row per series with array column
-- +----------+--------------------------+
-- | series_id| values                   |
-- +----------+--------------------------+
-- | A        | [10.5, 12.3, 11.8, ...]  |
-- | B        | [5.2, 6.1, 4.8, ...]     |
-- +----------+--------------------------+

SELECT
    series_id,

    -- Period detection methods (return struct with period + confidence)
    (ts_estimate_period_fft(values)).period AS fft_period,
    (ts_estimate_period_fft(values)).confidence AS fft_confidence,
```

```sql
    (ts_estimate_period_acf(values)).period AS acf_period,
    (ts_estimate_period_acf(values)).confidence AS acf_confidence,

    -- Autoperiod methods (FFT + ACF validation)
    (ts_autoperiod(values)).period AS autoperiod_period,
    (ts_autoperiod(values)).detected AS autoperiod_detected,
    (ts_autoperiod(values)).acf_validation AS autoperiod_score,

    (ts_cfd_autoperiod(values)).period AS cfd_period,
    (ts_cfd_autoperiod(values)).acf_validation AS cfd_score,

    -- Model-based methods
    (ts_aic_period(values)).period AS aic_period,
    (ts_aic_period(values)).r_squared AS aic_r_squared,

    -- Spectral methods
    (ts_lomb_scargle(values)).period AS lomb_period,
    (ts_lomb_scargle(values)).false_alarm_prob AS lomb_fap,

    (ts_sazed_period(values)).period AS sazed_period,
    (ts_sazed_period(values)).snr AS sazed_snr,

    -- Decomposition methods
    (ts_stl_period(values)).period AS stl_period,
    (ts_stl_period(values)).seasonal_strength AS stl_strength,

    (ts_ssa_period(values)).period AS ssa_period,
    (ts_ssa_period(values)).variance_explained AS ssa_variance,

    -- Pattern-based methods
    (ts_matrix_profile_period(values)).period AS mp_period,
    (ts_matrix_profile_period(values)).confidence AS mp_confidence,

    -- Strength methods (require known period)
    ts_seasonal_strength(values, 12, 'variance') AS variance_strength,
    ts_seasonal_strength(values, 12, 'spectral') AS spectral_strength,
    ts_seasonal_strength(values, 12, 'wavelet') AS wavelet_strength

FROM wide_format_data;
```

Extract Confidence Scores

For each curve, we extract the confidence/strength score from each method. Scores are normalized to [0, 1] where possible for fair comparison.

```
Extracted scores for 550 curves
```

Score Distributions by Ground Truth
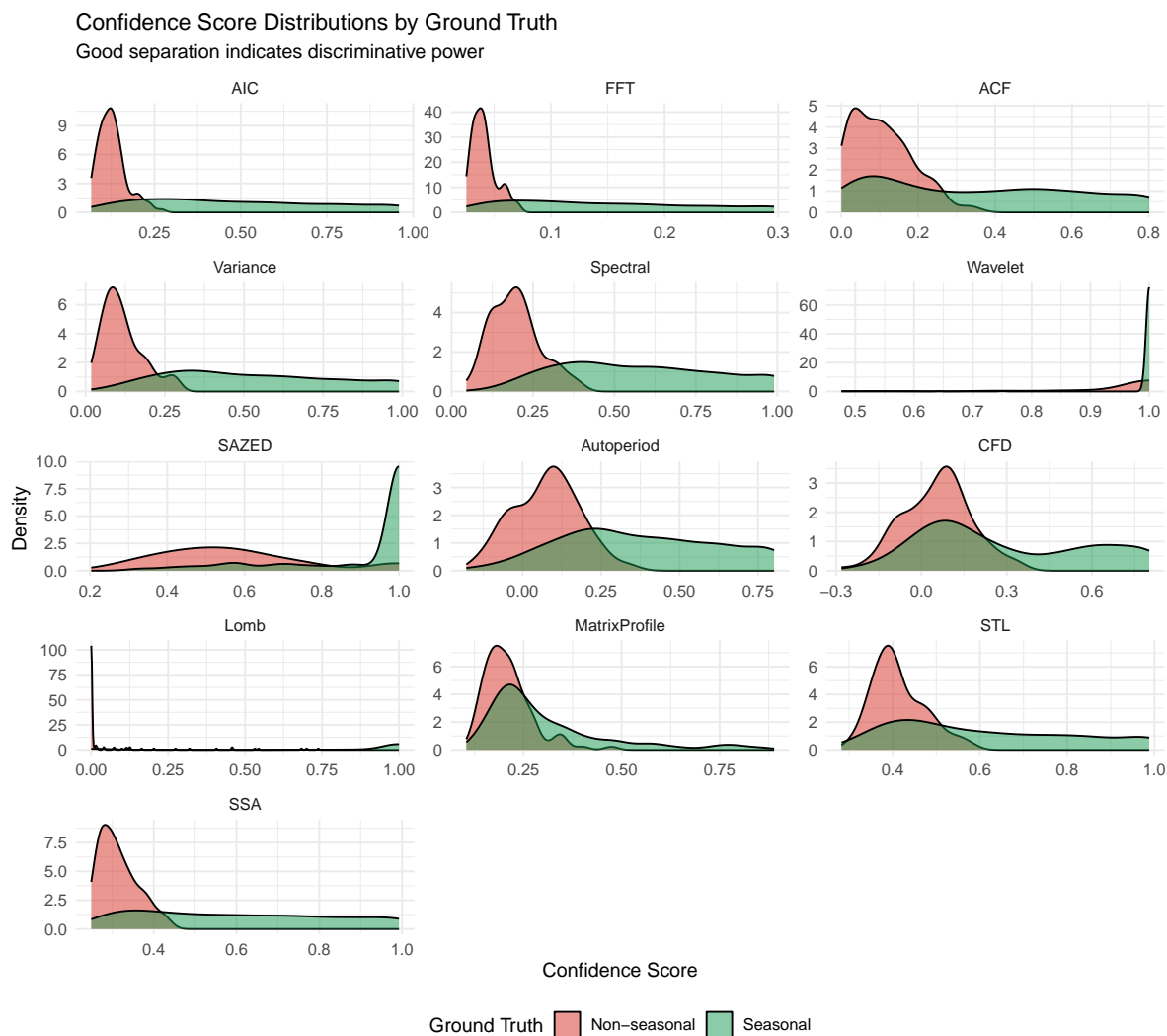
Figure 3: Distribution of confidence scores by ground truth (seasonal vs non-seasonal)

Threshold Calibration

For methods like Matrix Profile and SSA that lack natural calibration, we use null distribution calibration. By running each method on white noise samples, we find the threshold that produces exactly 5% false positives.

```
Calibrating detection thresholds on null distribution (white noise)...
```

```
Calibrated thresholds:
  MatrixProfile: 0.5000
  SSA: 0.5000
```

Precision-Recall Analysis

Given the class imbalance (82% seasonal vs 18% non-seasonal), **Precision-Recall (PR) curves are more informative than ROC curves**. PR curves focus on positive class performance and are not inflated by the large number of true negatives.

Table 3: Precision-Recall Analysis Summary (sorted by PR AUC)

Method	PR AUC	ROC AUC	Optimal Threshold	Recall	Precision
Variance	0.992	0.962	0.215	0.896	0.982
Spectral	0.989	0.952	0.335	0.822	0.991
AIC	0.987	0.937	0.213	0.822	0.991
FFT	0.986	0.935	0.063	0.816	0.991
Lomb	0.985	0.931	0.570	0.787	0.993
SSA	0.976	0.892	0.425	0.713	0.996
Autoperiod	0.969	0.863	0.202	0.727	0.978
SAZED	0.956	0.858	0.794	0.773	0.974
STL	0.954	0.801	0.501	0.607	0.982
ACF	0.950	0.782	0.268	0.571	0.996
CFD	0.936	0.738	0.268	0.447	0.993
MatrixProfile	0.923	0.719	0.229	0.604	0.941
Wavelet	0.852	0.608	0.991	0.996	0.829

Precision-Recall Curves (Faceted by Method Family)

Precision–Recall Curves by Method Family
Dashed line = random classifier (82% baseline precision)
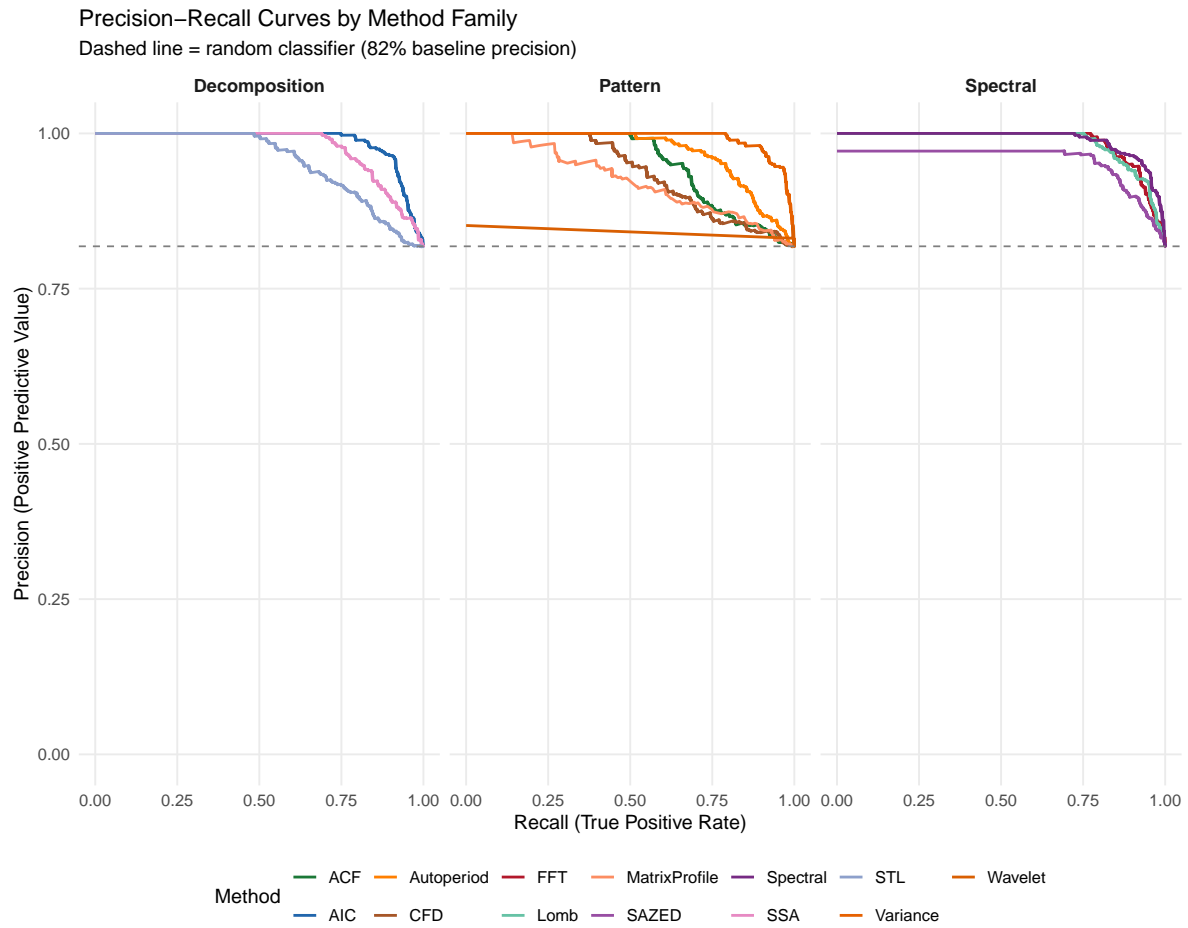
Figure 4: Precision-Recall Curves by Method Family: Pattern-based methods (Variance, Wavelet) achieve highest PR AUC. Spectral methods show variable precision at high recall. The baseline (random classifier) is shown as the dashed line at 82% precision.

PR AUC Comparison

Precision–Recall AUC by Method

Higher PR AUC indicates better performance on imbalanced data

Method	PR AUC
Wavelet	0.852
MatrixProfile	0.923
CFD	0.936
ACF	0.950
STL	0.954
SAZED	0.956
Autoperiod	0.969
SSA	0.976
Lomb	0.985
FFT	0.986
AIC	0.987
Spectral	0.989
Variance	0.992
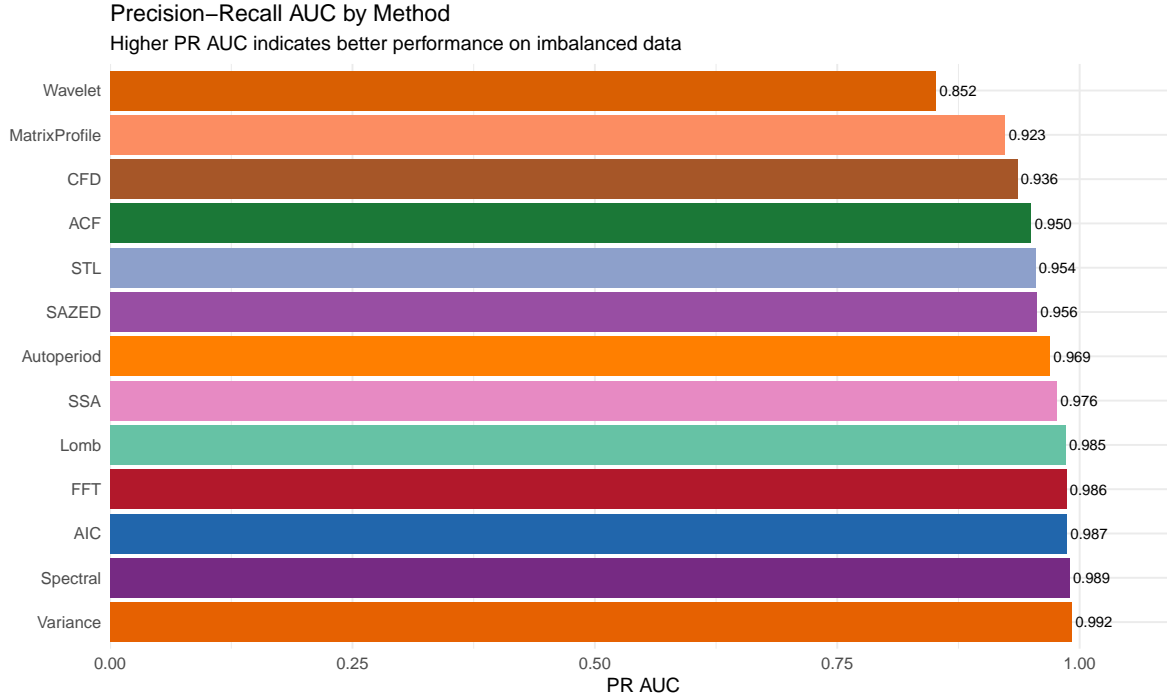
Figure 5: PR AUC Comparison: Variance and Wavelet methods achieve the highest precision-recall performance. PR AUC is more appropriate than ROC AUC for imbalanced datasets.

Optimal Threshold Estimation

The **optimal classification threshold** for each method is determined using **Youden's J statistic** (Youden, 1950), which maximizes the sum of sensitivity and specificity:

$$J = \text{Sensitivity} + \text{Specificity} - 1 = \text{TPR} - \text{FPR}$$

The threshold that maximizes J represents the point on the ROC curve farthest from the diagonal (random classifier), providing the best trade-off between detecting true seasonality (sensitivity) and avoiding false positives (specificity).

Why Youden's J?

- **Balanced approach**: Does not favor sensitivity over specificity
- **Threshold-independent**: Works for any score distribution

- **Standard practice**: Widely used in diagnostic test evaluation

Alternative threshold selection methods include:

- **Fixed sensitivity**: Set threshold to achieve target sensitivity (e.g., 0.90)
- **Cost-based**: Minimize misclassification cost when FP and FN have different costs
- **Prevalence-adjusted**: Account for class imbalance in threshold selection

The `pROC::coords()` function with `best` method implements Youden's J optimization.

Reference: Youden, W. J. (1950). Index for rating diagnostic tests. *Cancer*, 3(1), 32-35.

Runtime Performance

Computational efficiency matters for production deployments. We benchmark each method's execution time per 1000 series using our synthetic dataset.
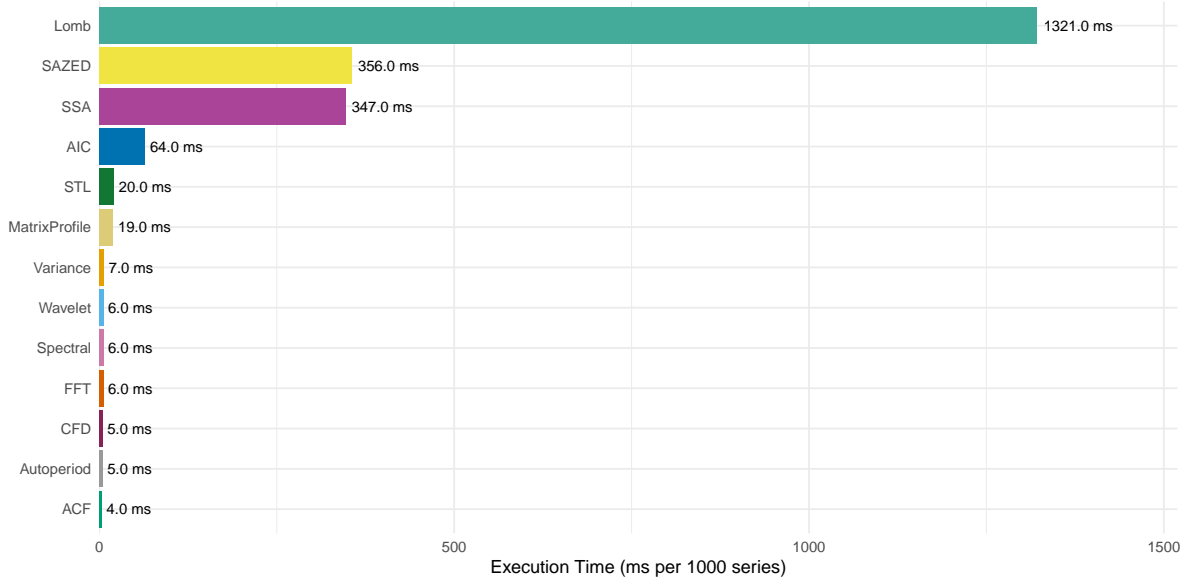
Figure 6: Execution Time per 1000 Series: Variance-based methods are fastest, while decomposition methods (STL, SSA) have higher computational cost. All methods complete in under 500ms for 1000 series, making them suitable for batch processing.

Classification Performance

Using the optimal threshold from ROC analysis (Youden's J statistic), we convert continuous scores into binary predictions and compute standard classification metrics.

We calculate Accuracy, Precision (positive predictive value), Recall (sensitivity), Specificity, False Positive Rate, and F1 score for each method.

Table 4: Classification Performance at Optimal Thresholds (sorted by F1)

Method	Accuracy	Precision	Recall	Specificity	FPR	F1
Variance	0.900	0.981	0.896	0.92	0.08	0.936
Wavelet	0.855	0.852	0.996	0.22	0.78	0.918
AIC	0.847	0.989	0.822	0.96	0.04	0.898
Spectral	0.847	0.989	0.822	0.96	0.04	0.898
FFT	0.842	0.989	0.816	0.96	0.04	0.894
Lomb	0.820	0.992	0.787	0.97	0.03	0.877
SAZED	0.793	0.967	0.773	0.88	0.12	0.859
Autoperiod	0.758	0.970	0.727	0.90	0.10	0.831
SSA	0.762	0.994	0.713	0.98	0.02	0.831
STL	0.664	0.972	0.607	0.92	0.08	0.747
MatrixProfile	0.627	0.910	0.604	0.73	0.27	0.726
ACF	0.645	0.992	0.571	0.98	0.02	0.725
CFD	0.542	0.985	0.447	0.97	0.03	0.615

Performance Comparison

Classification Performance Metrics by Method
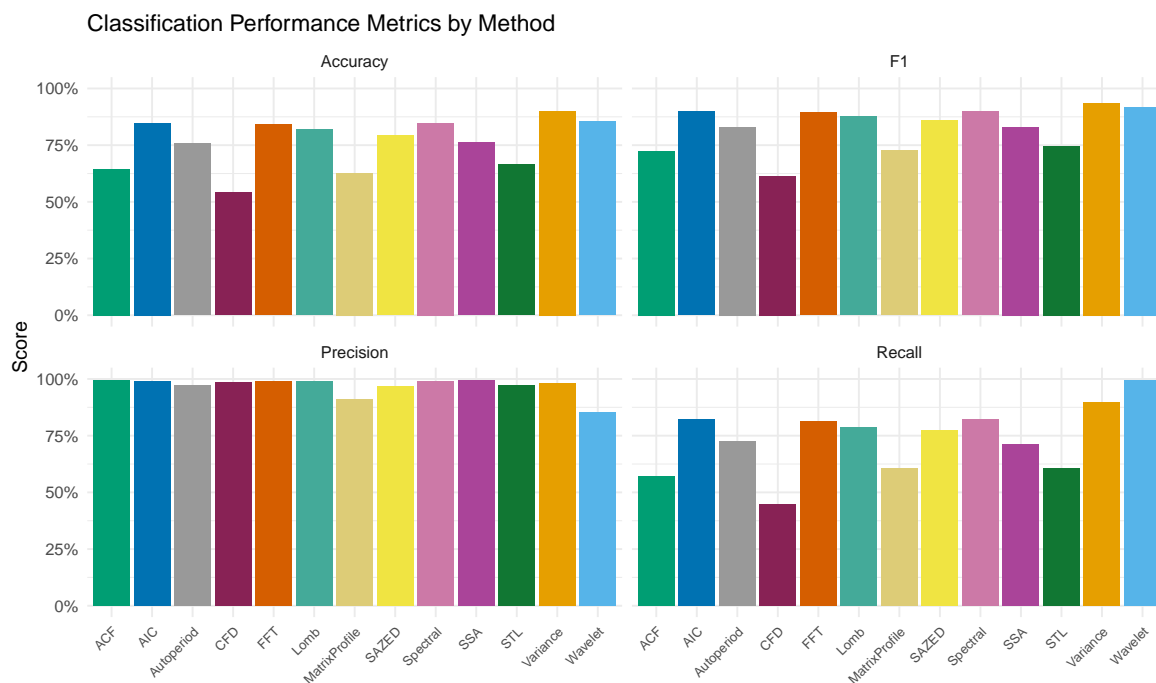
Figure 7: Classification metrics comparison across methods

Confusion Matrices

Confusion matrices show the detailed breakdown of true positives, false positives, true negatives, and false negatives for the top-performing methods.
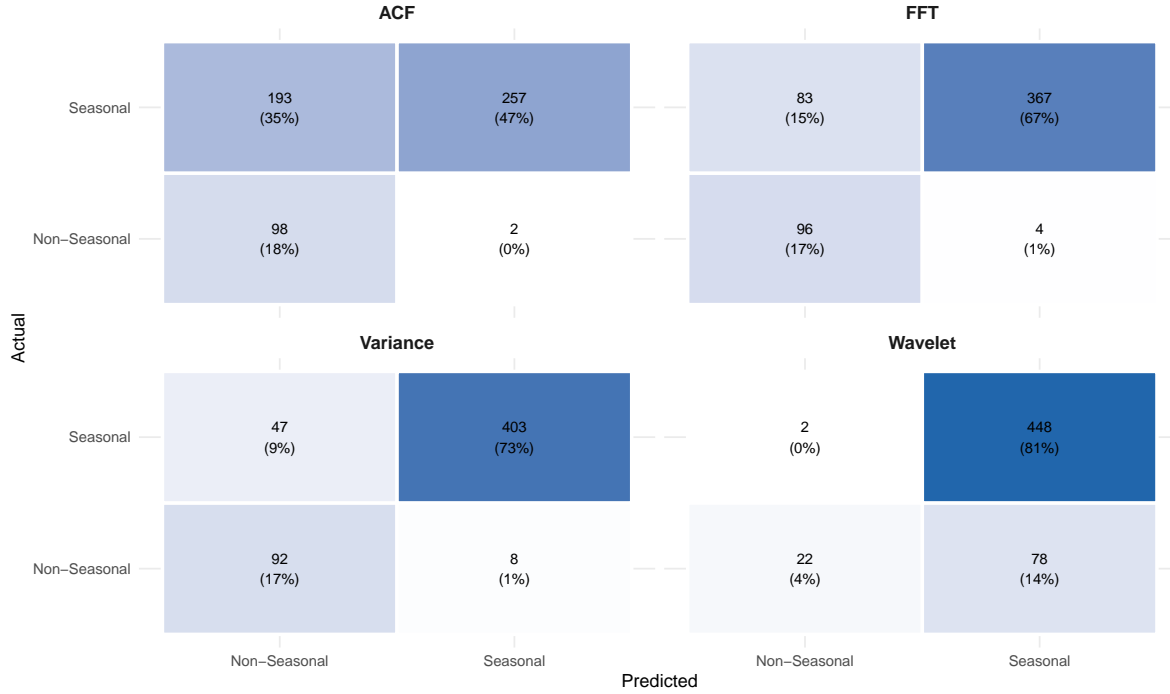
Figure 8: Confusion Matrices for Top 4 Methods: Variance and Wavelet achieve the best balance of sensitivity and specificity. Note that false negatives (missing seasonal patterns) are generally more costly than false positives in forecasting applications.

Statistical Significance: McNemar Tests

McNemar's test compares paired binary predictions between methods. A significant p-value indicates methods differ in their detection decisions.

Note: p-values adjusted for 78 pairwise comparisons using Benjamini-Hochberg (FDR) correction

Table 5: Significant McNemar Test Results (adjusted $p < 0.05$)

Method 1	Method 2	Chi-sq	p (raw)	p (adjusted)
CFD	Wavelet	320.0031	0.0000	0.0000
ACF	Wavelet	265.0037	0.0000	0.0000
STL	Wavelet	239.1004	0.0000	0.0000
MatrixProfile	Wavelet	206.7854	0.0000	0.0000
SSA	Wavelet	199.0439	0.0000	0.0000
CFD	Variance	197.3767	0.0000	0.0000

Method 1	Method 2	Chi-sq	p (raw)	p (adjusted)
Autoperiod	Wavelet	183.1295	0.0000	0.0000
Lomb	Wavelet	163.1445	0.0000	0.0000
AIC	CFD	158.6722	0.0000	0.0000
SAZED	Wavelet	158.2849	0.0000	0.0000
CFD	Spectral	156.9286	0.0000	0.0000
CFD	FFT	155.6836	0.0000	0.0000
FFT	Wavelet	151.0573	0.0000	0.0000
AIC	Wavelet	150.0066	0.0000	0.0000
Spectral	Wavelet	150.0066	0.0000	0.0000
CFD	SAZED	150.1562	0.0000	0.0000
ACF	Variance	144.3101	0.0000	0.0000
CFD	Lomb	141.7423	0.0000	0.0000
Autoperiod	CFD	120.1655	0.0000	0.0000
Variance	Wavelet	113.0087	0.0000	0.0000
STL	Variance	108.0584	0.0000	0.0000
ACF	AIC	107.4050	0.0000	0.0000
ACF	Spectral	107.4050	0.0000	0.0000
CFD	SSA	106.2901	0.0000	0.0000
ACF	FFT	104.4153	0.0000	0.0000
ACF	SAZED	97.0874	0.0000	0.0000
ACF	Lomb	90.4712	0.0000	0.0000
SSA	Variance	80.5213	0.0000	0.0000
ACF	Autoperiod	72.3049	0.0000	0.0000
AIC	STL	72.3419	0.0000	0.0000
Spectral	STL	72.3419	0.0000	0.0000
FFT	STL	69.4825	0.0000	0.0000
MatrixProfile	Variance	61.6050	0.0000	0.0000
CFD	STL	56.0777	0.0000	0.0000
Autoperiod	Variance	55.5104	0.0000	0.0000
Lomb	STL	55.1471	0.0000	0.0000
SAZED	STL	54.8108	0.0000	0.0000
ACF	SSA	53.6351	0.0000	0.0000
CFD	MatrixProfile	51.0751	0.0000	0.0000
AIC	SSA	45.4545	0.0000	0.0000
Lomb	Variance	45.3065	0.0000	0.0000
ACF	CFD	39.9452	0.0000	0.0000
Spectral	SSA	39.6825	0.0000	0.0000
FFT	SSA	39.4464	0.0000	0.0000
Autoperiod	STL	34.3750	0.0000	0.0000
FFT	Variance	33.0652	0.0000	0.0000
Spectral	Variance	31.6098	0.0000	0.0000

Method 1	Method 2	Chi-sq	p (raw)	p (adjusted)
AIC	Variance	30.1395	0.0000	0.0000
AIC	MatrixProfile	29.9235	0.0000	0.0000
MatrixProfile	Spectral	28.9735	0.0000	0.0000
SAZED	SSA	28.8000	0.0000	0.0000
SAZED	Variance	28.7356	0.0000	0.0000
FFT	MatrixProfile	27.6978	0.0000	0.0000
Lomb	SSA	25.9286	0.0000	0.0000
Autoperiod	Spectral	21.9661	0.0000	0.0000
SSA	STL	20.5000	0.0000	0.0000
MatrixProfile	SAZED	20.1117	0.0000	0.0000
Lomb	MatrixProfile	18.2528	0.0000	0.0000
AIC	Autoperiod	17.7534	0.0000	0.0000
Autoperiod	FFT	16.0147	0.0001	0.0001
AIC	Lomb	11.1304	0.0008	0.0011
ACF	MatrixProfile	8.7414	0.0031	0.0039
FFT	Lomb	8.4500	0.0037	0.0045
Lomb	Spectral	8.2581	0.0041	0.0049
Autoperiod	MatrixProfile	7.7784	0.0053	0.0063
ACF	STL	7.1129	0.0077	0.0090
Autoperiod	SAZED	6.4533	0.0111	0.0129
Autoperiod	Lomb	6.0167	0.0142	0.0163

McNemar P-Value Heatmap (FDR-Adjusted)

McNemar Test P–Values Between Methods
FDR–adjusted (Benjamini–Hochberg); Red = significant difference (adj. p < 0.05)

	ACF	AIC	Autoperiod	CFD	FFT	Lomb	MatrixProfile	SAZED	Spectral	SSA	STL	Variance	Wavelet
Wavelet	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	
Variance	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001		<.001
STL	0.009	<.001	<.001	<.001	<.001	<.001	0.211	<.001	<.001	<.001		<.001	<.001
SSA	<.001	<.001	0.107	<.001	<.001	<.001	0.089	<.001	<.001		<.001	<.001	<.001
Spectral	<.001	0.860	<.001	<.001	0.729	0.005	<.001	0.107		<.001	<.001	<.001	<.001
SAZED	<.001	0.087	0.013	<.001	0.206	0.798	<.001		0.107	<.001	<.001	<.001	<.001
MatrixProfile	0.004	<.001	0.006	<.001	<.001	<.001		<.001	<.001	0.089	0.211	<.001	<.001
Lomb	<.001	0.001	0.016	<.001	0.005		<.001	0.798	0.005	<.001	<.001	<.001	<.001
FFT	<.001	0.602	<.001	<.001		0.005	<.001	0.206	0.729	<.001	<.001	<.001	<.001
CFD	<.001	<.001	<.001		<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001	<.001
Autoperiod	<.001	<.001		<.001	<.001	0.016	0.006	0.013	<.001	0.107	<.001	<.001	<.001
AIC	<.001		<.001	<.001	0.602	0.001	<.001	0.087	0.860	<.001	<.001	<.001	<.001
ACF		<.001	<.001	<.001	<.001	<.001	0.004	<.001	<.001	<.001	0.009	<.001	<.001

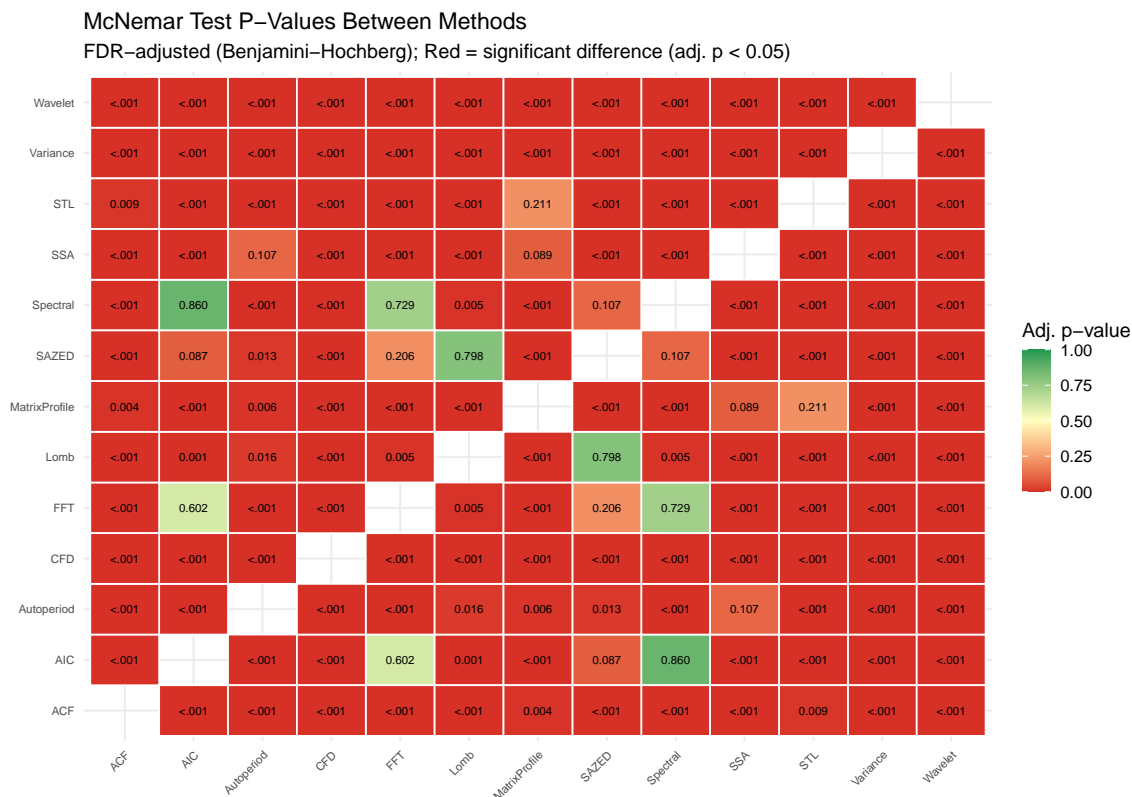Adj. p–value: 1.00, 0.75, 0.50, 0.25, 0.00

Figure 9: McNemar test p-values (FDR-adjusted) between method pairs (red = significant difference)

Fisher's g-test Comparison

Fisher's g-test is a classical statistical test for periodicity that tests whether the maximum periodogram value is significantly larger than expected under the null hypothesis of white noise. We compare its false positive rate against the heuristic FFT method.

Table 6: False Positive Rates: Fisher's g-test vs Heuristic FFT

Scenario	Fisher's g FPR	FFT FPR
AR(1) phi=0.5	0.060	0
AR(1) phi=0.9	0.775	0
White Noise	0.005	0

Fisher's g-test vs FFT: False positive rate comparison on white noise and AR(1) noise scenarios. Fisher's g-test provides statistical guarantees while FFT uses a heuristic threshold.
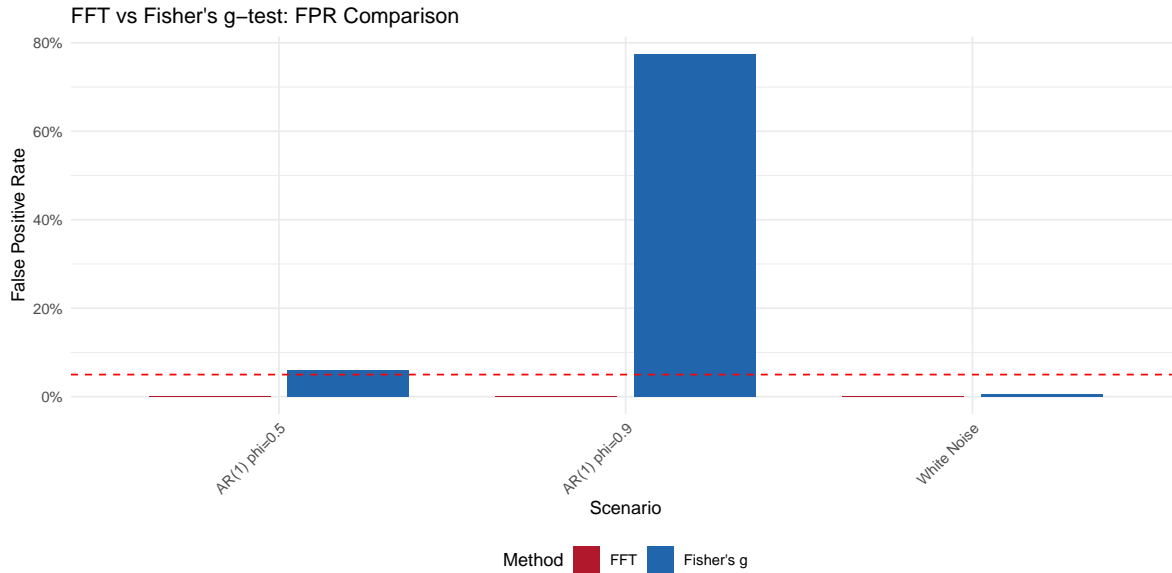

Figure 10: Fisher's g-test vs FFT: False positive rate comparison on white noise and AR(1) noise scenarios. Fisher's g-test provides statistical guarantees while FFT uses a heuristic threshold.

Key Finding: Fisher's g-test maintains the target 5% FPR on white noise, while the heuristic FFT threshold may not be well-calibrated. However, both methods can have elevated FPR on AR(1) processes due to spurious spectral peaks.

Computational Complexity

Understanding how execution time scales with series length is crucial for production deployments. We benchmark each method across different series lengths to characterize their computational complexity.

```
Complexity benchmark: 0/25 measurements successful
```

```
Warning: No valid timing measurements obtained.
```

Table 7: Computational Scaling Summary: Time increase from 60 to 480 points

Method	Min Length	Max Length	Time at 60pts (ms)	Time at 480pts (ms)	Scaling Factor
Autoperiod	30	480	NA	NA	NA
FFT	30	480	NA	NA	NA
MatrixProfile	30	480	NA	NA	NA
STL	30	480	NA	NA	NA
Variance	30	480	NA	NA	NA

Challenge Scenarios

Following the fdars benchmark, we test method robustness under challenging conditions.

Challenge 1: Linear Trends

Linear trends are common in real-world data and can mask or mimic seasonality. We add trends of varying slopes (0.1, 0.3, 0.5 per time unit) to test robustness. Methods that operate on differenced data (CFD-Autoperiod) should be more robust.

```
Generated 450 curves with trends
```
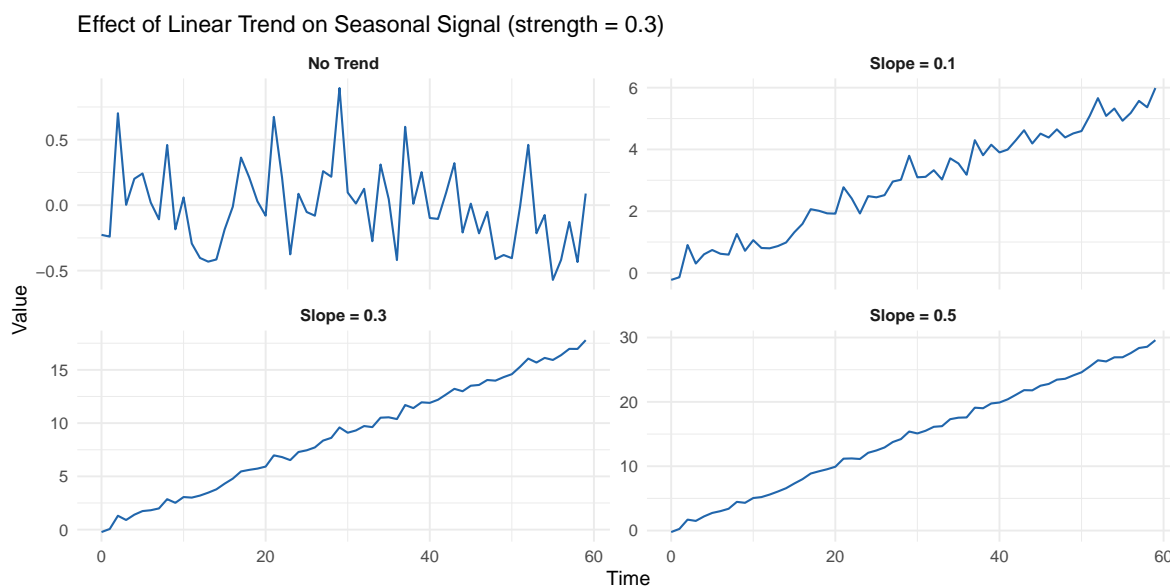

Figure 11: Effect of linear trends on seasonal signal (faceted by slope). Each panel shows the same seasonal pattern with different trend magnitudes added.

Challenge 1b: Non-linear Trends

Beyond linear trends, real-world data often exhibits non-linear trend patterns. We test quadratic (accelerating/decelerating) and sinusoidal (cyclical) trends that can confuse period detection methods.

```
Generated 450 curves with quadratic trends
```

```
Generated 450 curves with sinusoidal trends

Generated 450 curves with exponential trends

Generated 450 curves with logarithmic trends

Generated 450 curves with step trends

Generated 450 curves with piecewise trends
```
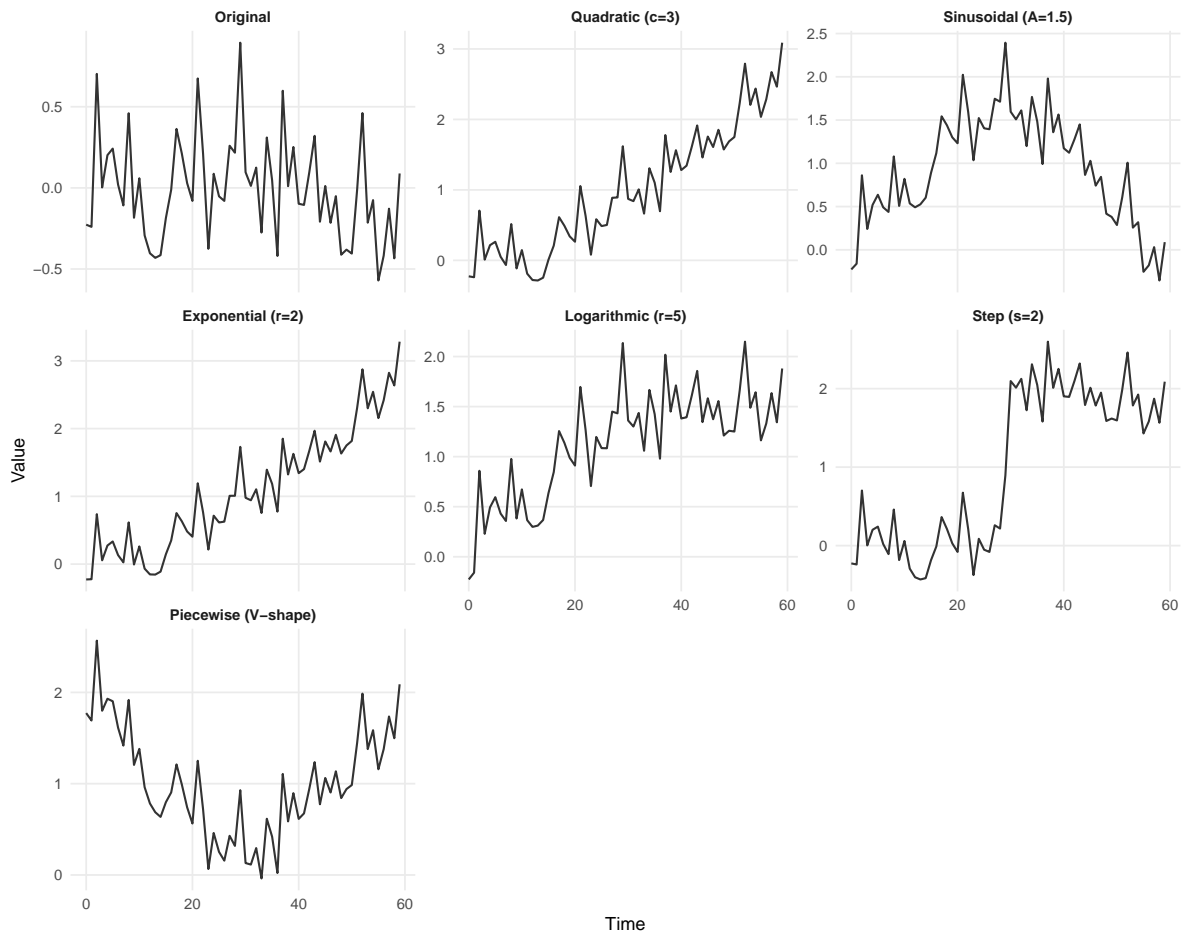

Figure 12: Effect of non-linear trends on seasonal signal (strength = 0.3). Six trend types representing common real-world patterns.

Challenge 2: Red Noise (AR(1) Process)

Why test Red Noise? Financial and climate time series often exhibit autocorrelation (red noise), which can produce spurious spectral peaks that fool FFT-based methods into detecting false seasonality. This is a critical test for specificity—methods that "hallucinate" seasonality in autocorrelated noise will generate many false positives in real-world applications.

We replace white noise with AR(1) noise with coefficients $\phi = 0.0$ (baseline), 0.3, 0.5, 0.7, and 0.9 (extreme). Higher ϕ means stronger autocorrelation.

`Generated 750 curves with AR(1) noise`
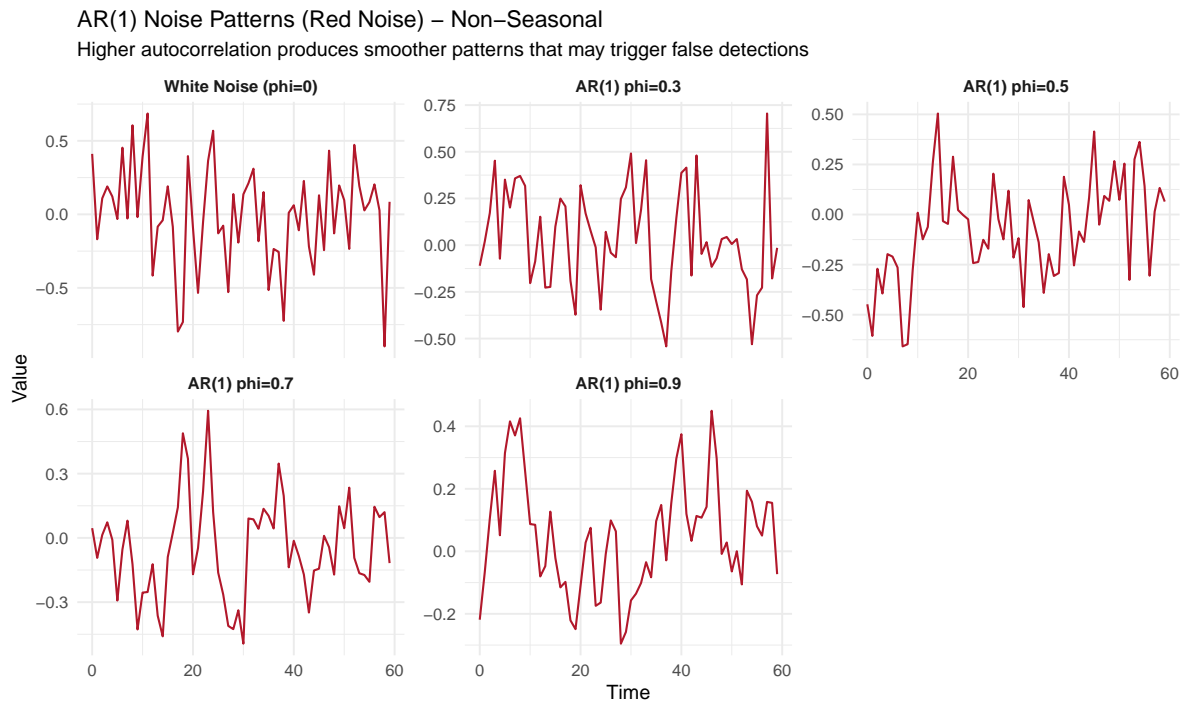
Figure 13: AR(1) noise patterns (faceted). Higher autocorrelation (phi) produces smoother patterns that can fool spectral methods into detecting false periodicity.

Challenge 3: Outlier Contamination

Why test Outliers? Sensor data, financial time series, and operational metrics frequently contain point anomalies from equipment failures, data entry errors, or exceptional events. Robust seasonality detection methods should maintain accuracy despite these extreme values.

28

We inject outliers with probability 5-10% and magnitude 3-5 standard deviations. Methods using variance-based measures may be particularly vulnerable, while differencing-based methods may amplify outlier effects.

```
Generated 450 curves with outliers
```
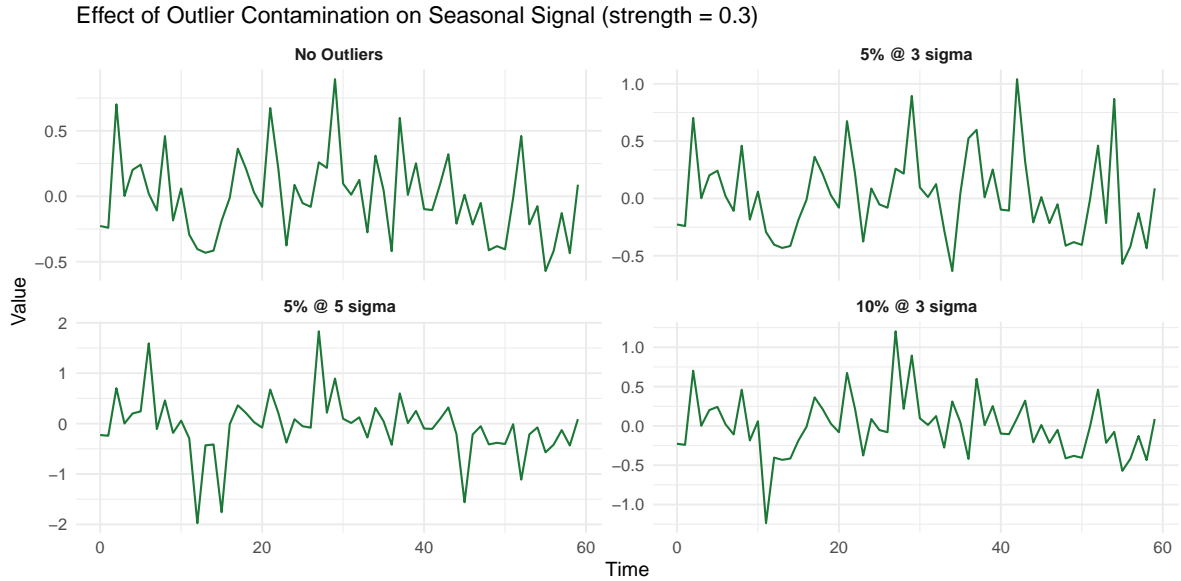

Figure 14: Effect of outlier contamination (faceted). Each panel shows the same seasonal pattern with different outlier configurations applied.

Challenge 4: Shape Robustness

Real-world seasonality is rarely a perfect sinusoid. We test methods' ability to detect non-sinusoidal periodic patterns including square waves and triangle waves, which contain higher harmonic content.

```
Generated 450 curves with different shapes
```
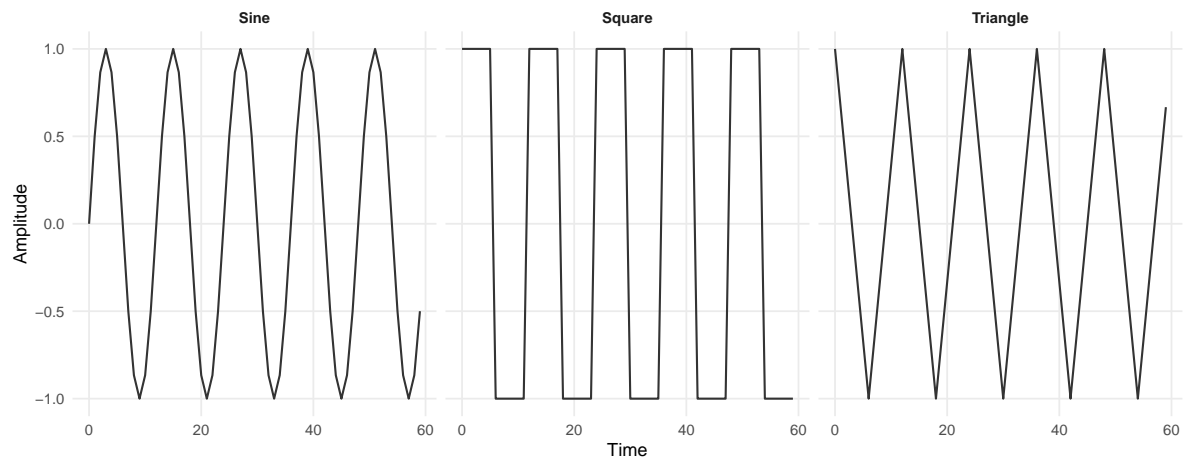
Figure 15: Seasonal waveform shapes (period = 12, normalized amplitude)

Challenge 5: Amplitude Modulation

Non-stationary seasonality with time-varying amplitude is common in real-world data (e.g., emerging or fading seasonal patterns). We test methods' robustness to amplitude modulation patterns.

```
Generated 600 curves with amplitude modulation
```
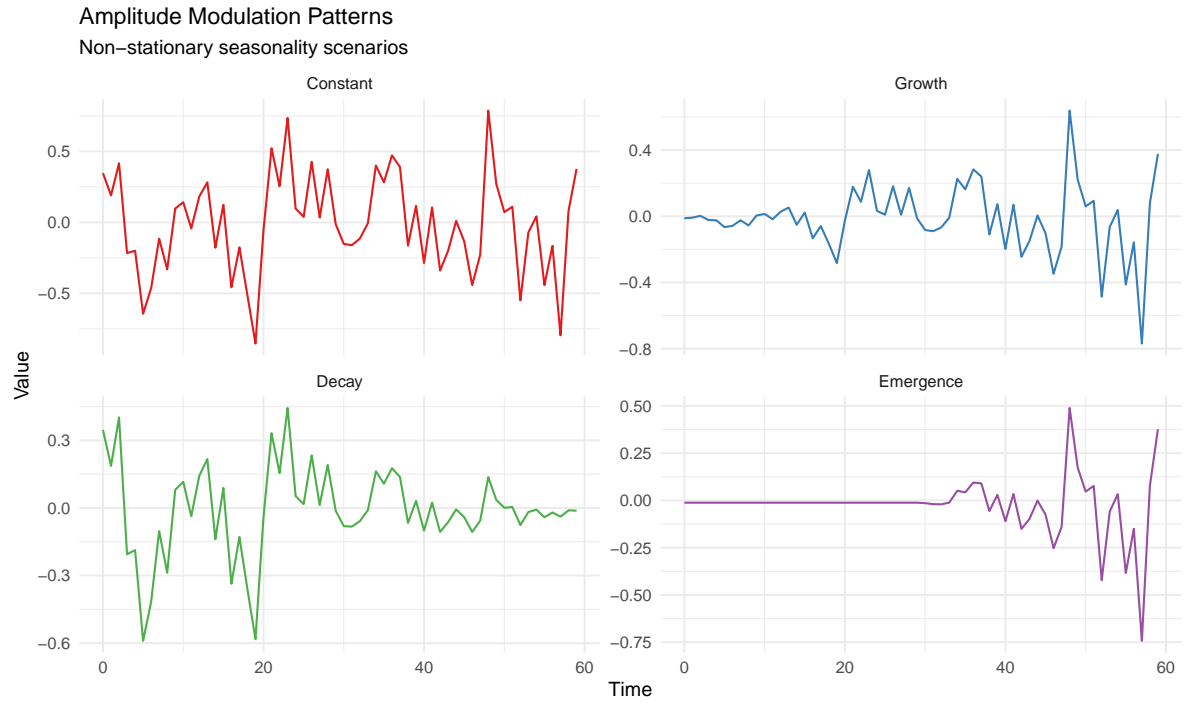
Figure 16: Amplitude modulation patterns applied to seasonal signal (strength = 0.5)

Challenge Scenario Performance

We evaluate all 13 methods on each challenge scenario, broken down by the specific challenge parameter to understand performance degradation.

Trend Robustness

Trend Robustness: PR AUC vs Trend Slope
Lower PR AUC at higher slopes indicates sensitivity to trends
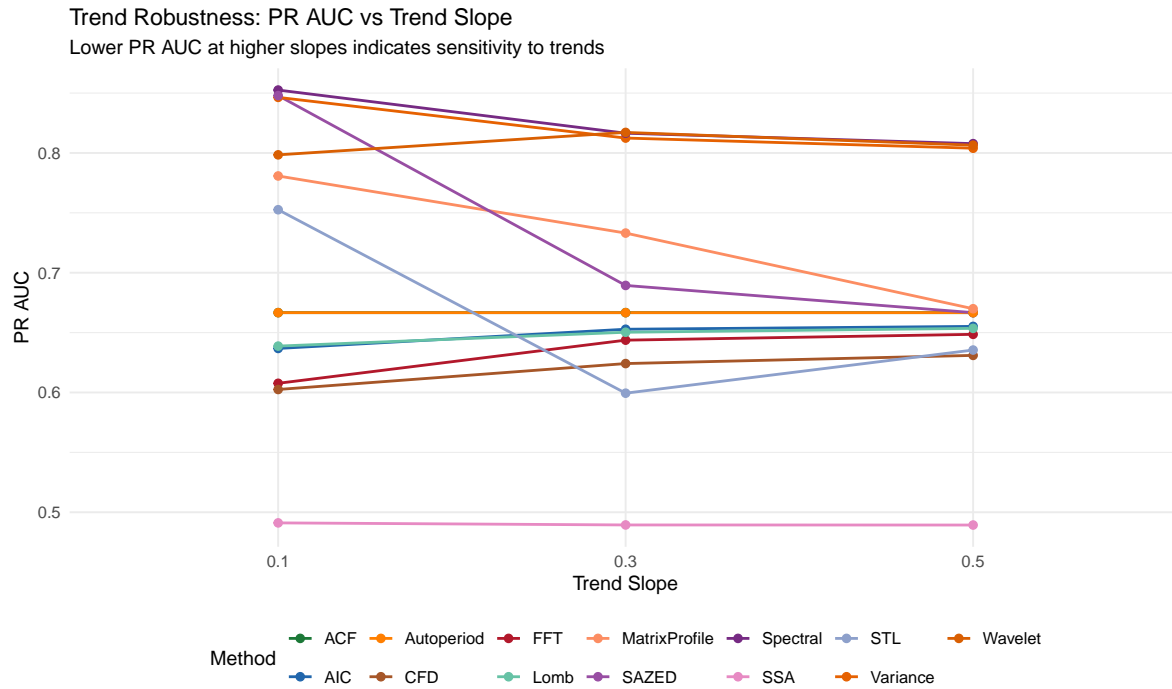
Figure 17: Method performance degradation with increasing trend slope

Red Noise Robustness (AR(1) False Positive Rate)

For red noise tests, we focus on **non-seasonal curves only** to measure false positive rate (FPR) — i.e., how often methods incorrectly detect seasonality in autocorrelated noise.
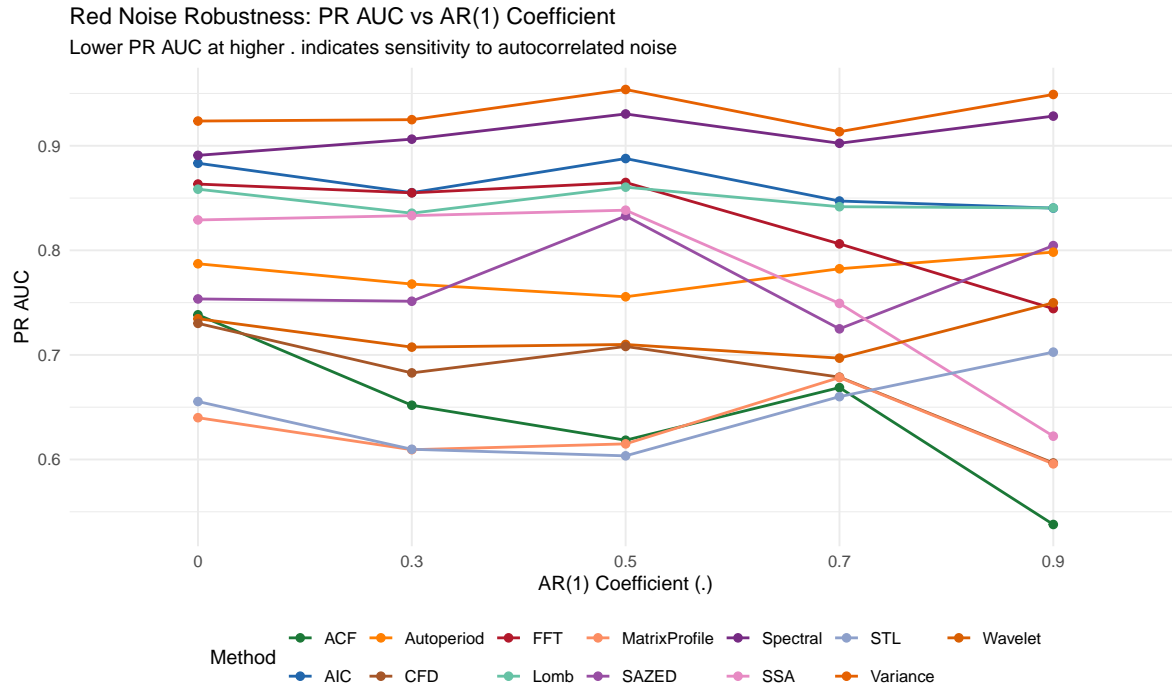
Figure 18: AUC with increasing AR(1) autocorrelation

False Positive Rate by Noise Type

The key concern with red noise is false positives—detecting seasonality where none exists. We compare FPR on purely non-seasonal series with white noise ($\phi=0$) vs extreme red noise ($\phi=0.9$).
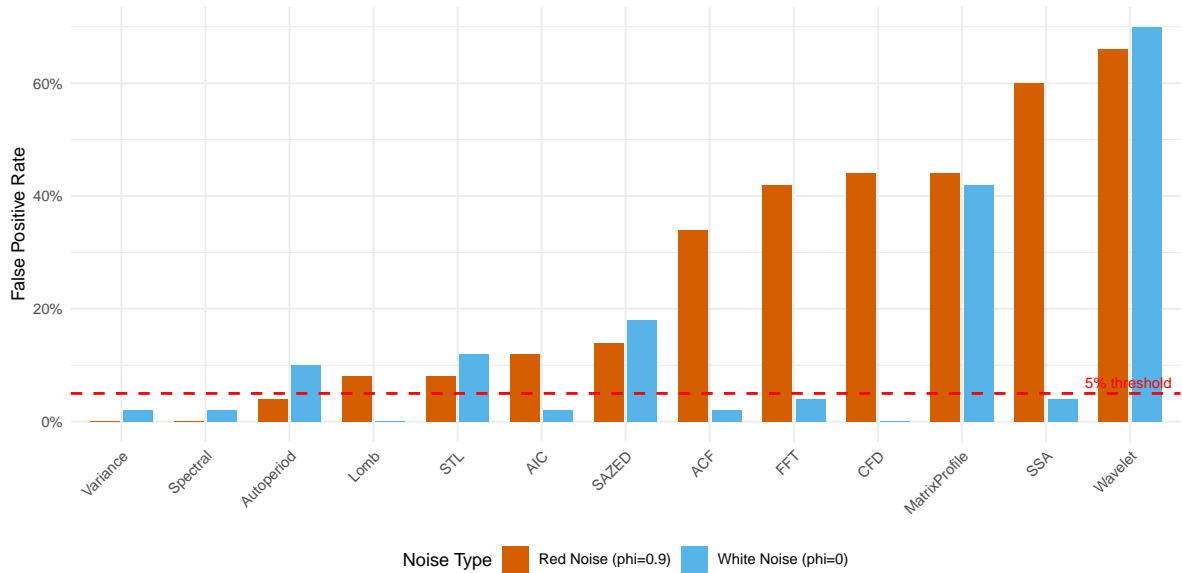
Figure 19: False Positive Rates: Methods that 'hallucinate' seasonality show elevated FPR in red noise. A well-calibrated detector should maintain FPR at most 5% (dashed line) regardless of noise structure.

Outlier Robustness
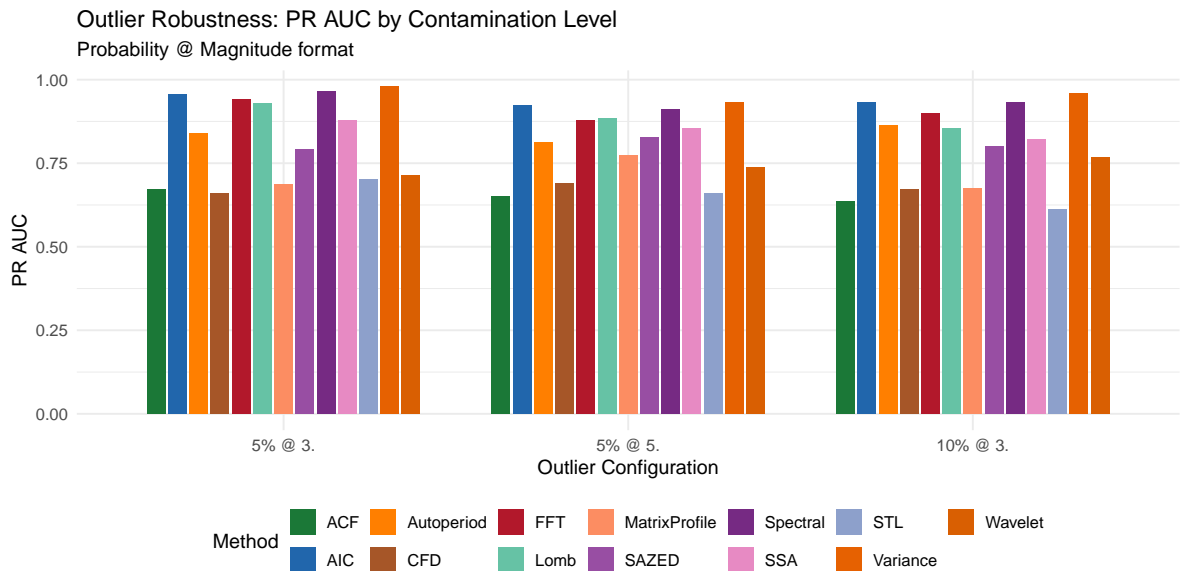
Figure 20: Method performance under different outlier configurations

Non-linear Trend Robustness

Quadratic Trend Robustness
PR AUC vs quadratic curvature coefficient
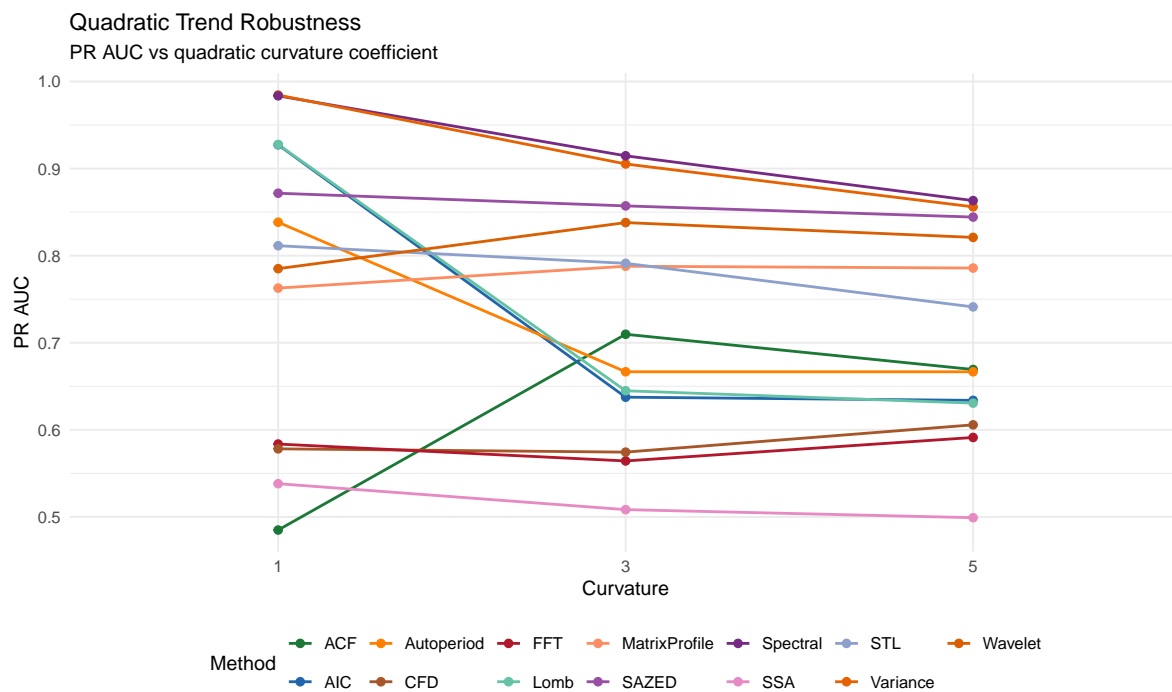
Figure 21: Method performance with increasing quadratic trend curvature
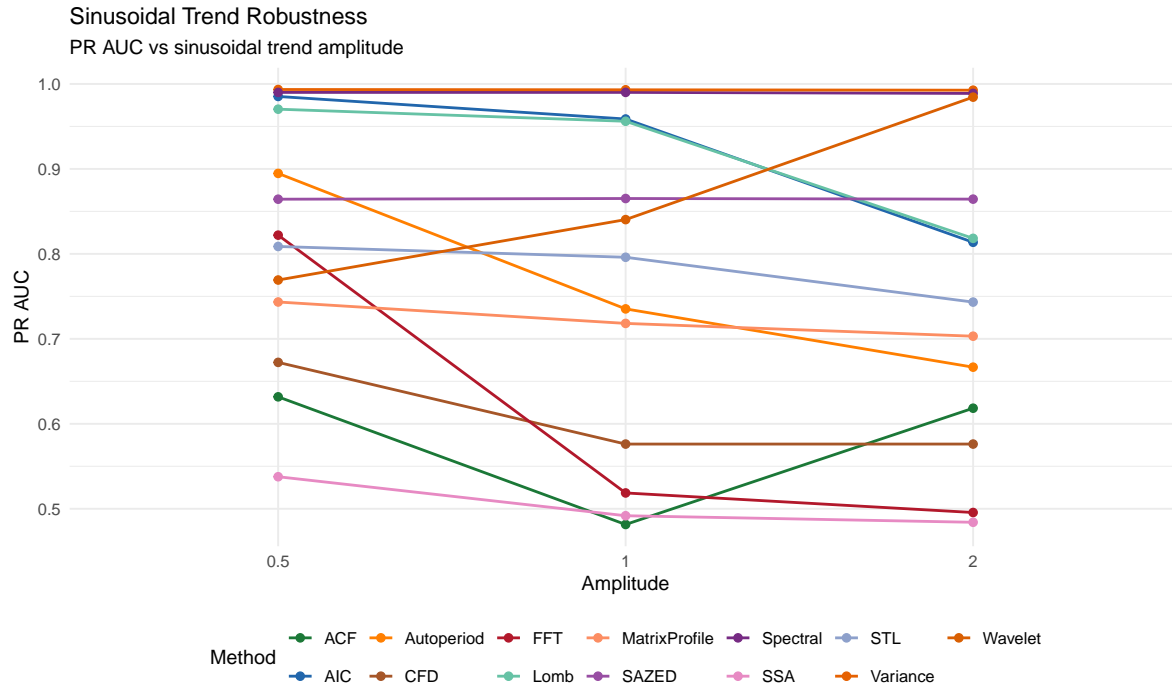
Figure 22: Method performance with increasing sinusoidal trend amplitude
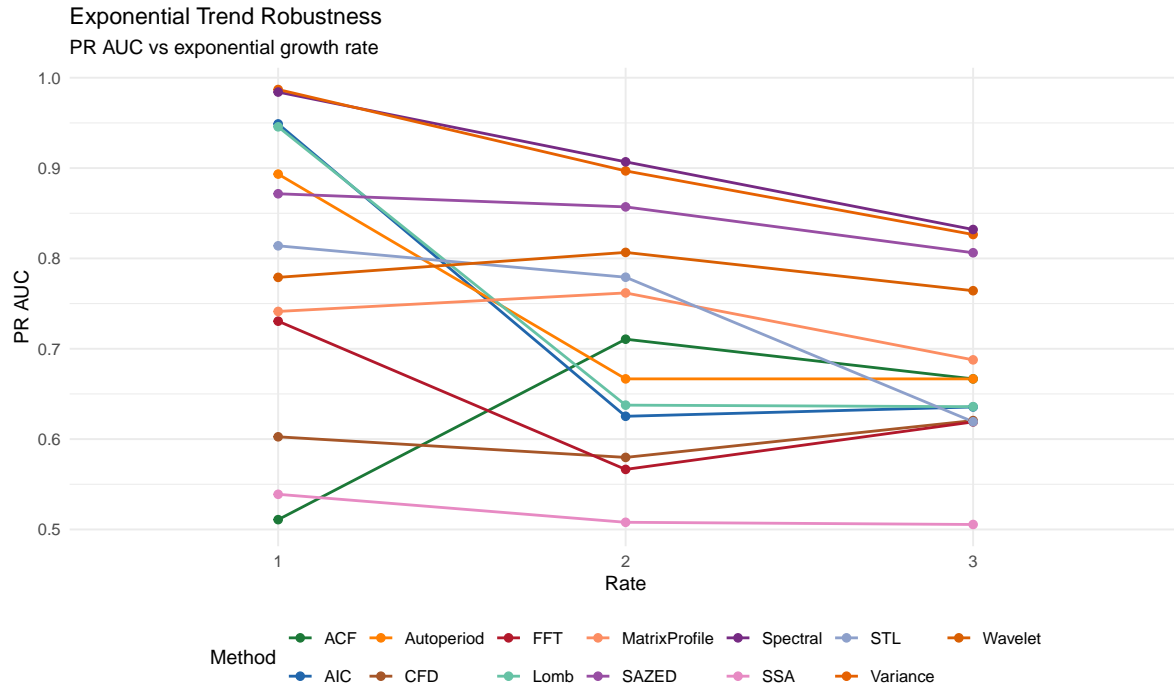
Figure 23: Method performance with increasing exponential trend rate
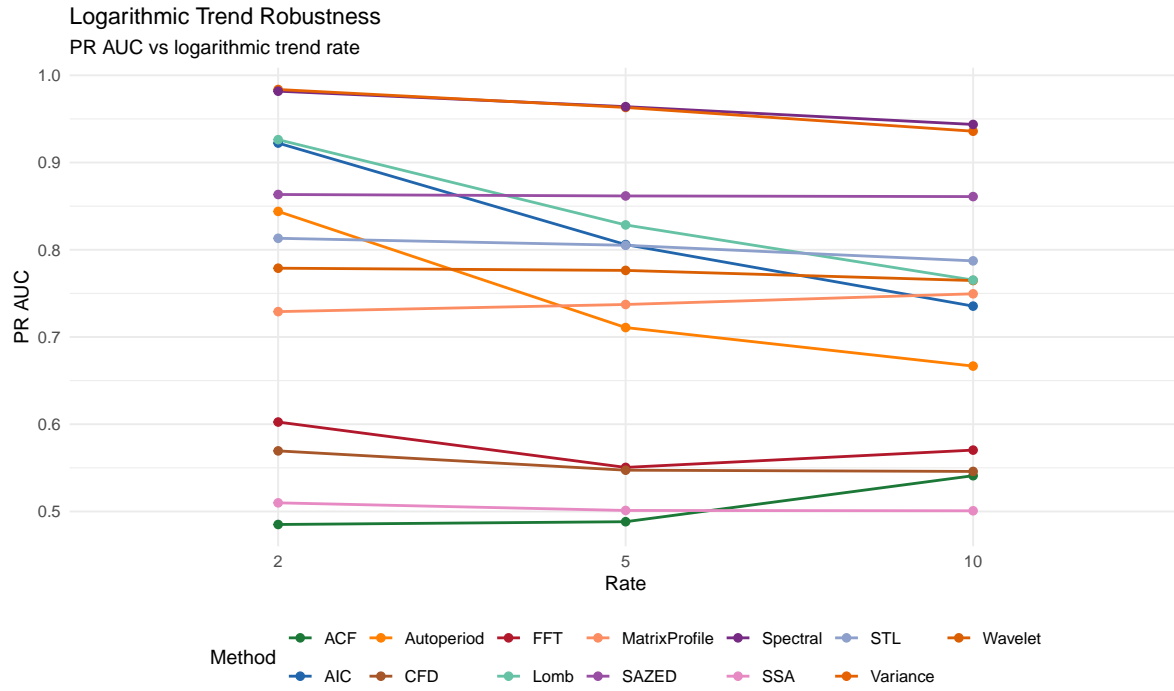
Figure 24: Method performance with increasing logarithmic trend rate
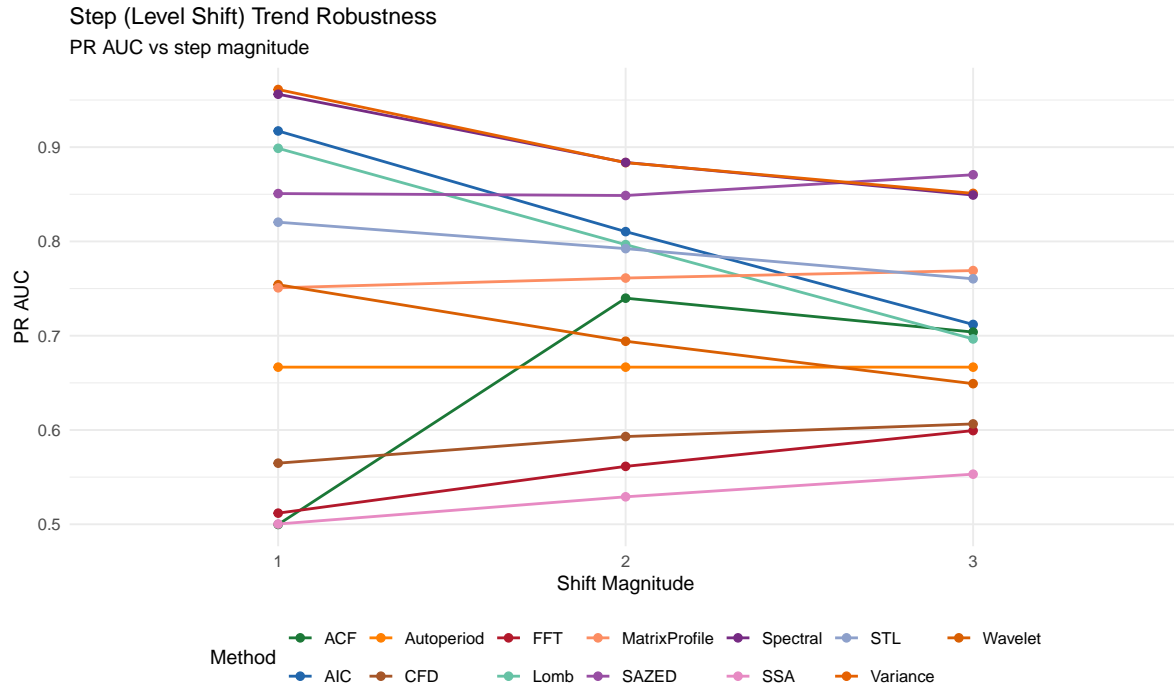
Figure 25: Method performance with increasing step (level shift) magnitude
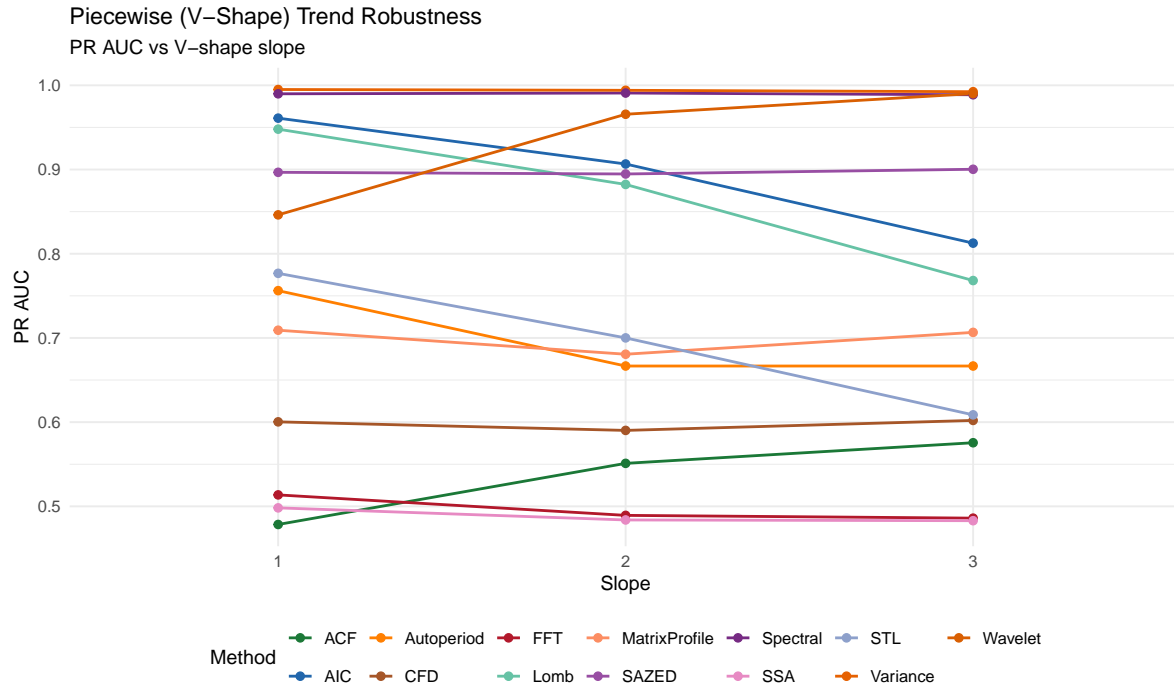
Figure 26: Method performance with increasing piecewise (V-shape) trend slope

Shape Robustness

Shape Robustness: PR AUC by Waveform Shape
Sine wave is baseline; square/triangle have higher harmonics
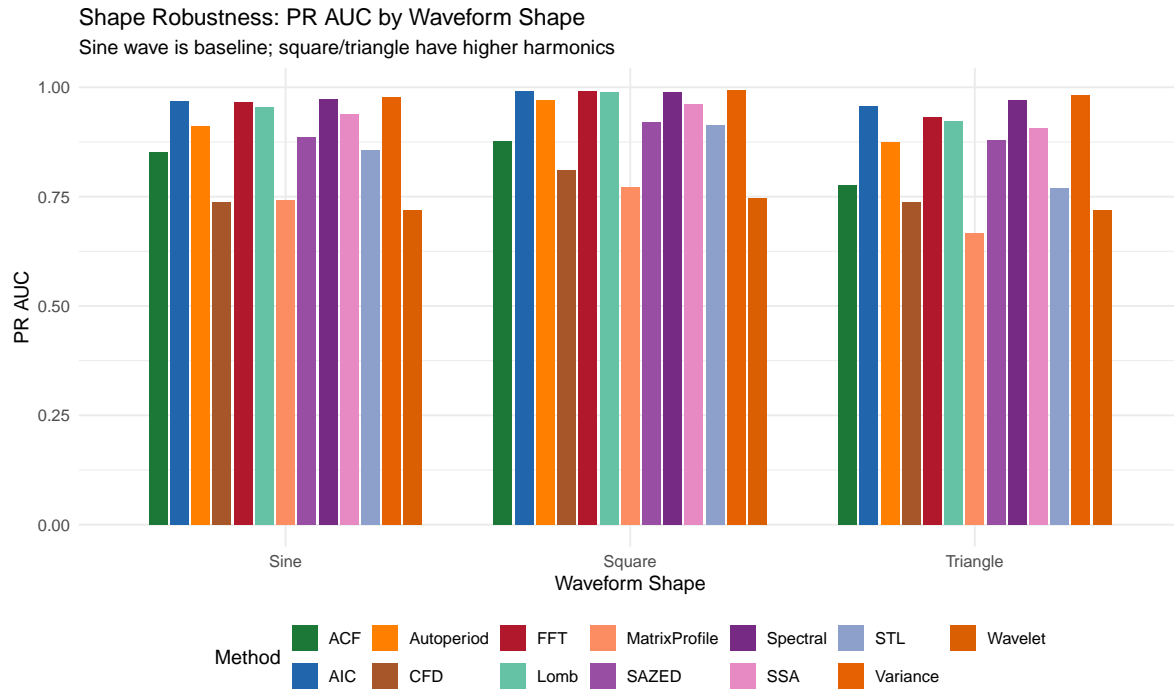
Figure 27: Method performance across waveform shapes

Amplitude Modulation Robustness
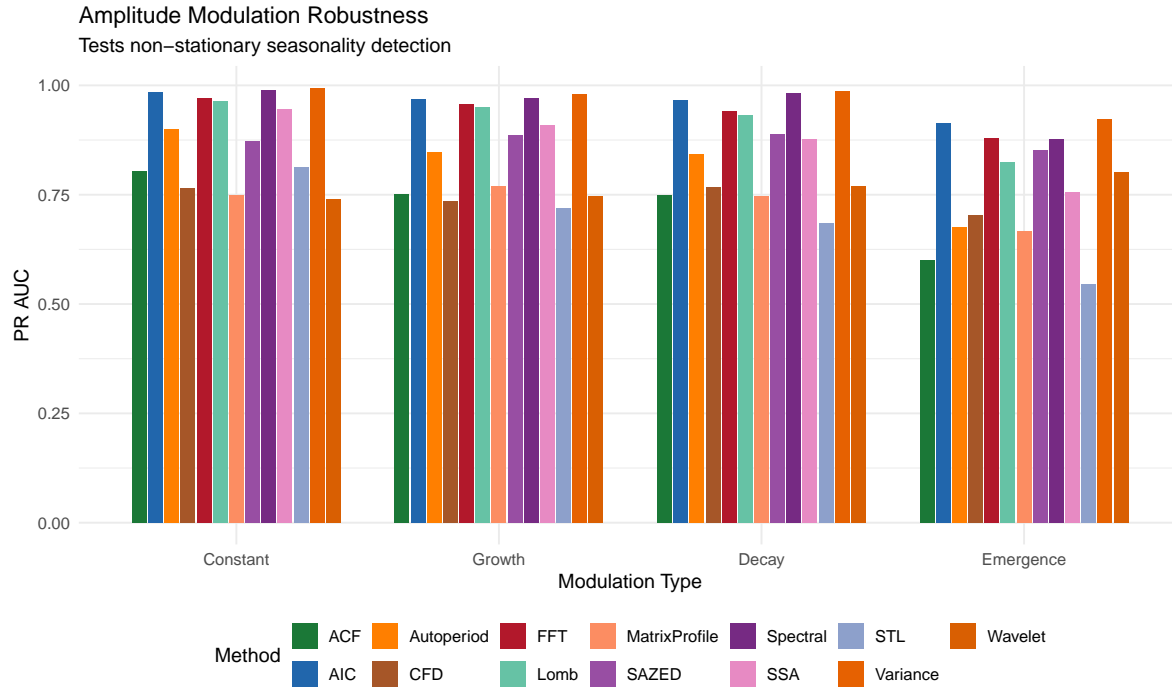
Figure 28: Method performance under amplitude modulation

Challenge Summary Table

Table 8: Extended Challenge Scenario Summary: PR AUC and F1 Statistics by Method

Challenge	Method	Mean PR AUC	Mean F1	Min PR AUC	Max PR AUC
Amplitude Modulation	Variance	0.971	0.923	0.923	0.995
Amplitude Modulation	AIC	0.958	0.867	0.913	0.984
Amplitude Modulation	Spectral	0.955	0.910	0.877	0.989
Amplitude Modulation	FFT	0.937	0.829	0.878	0.971
Amplitude Modulation	Lomb	0.918	0.805	0.825	0.963

42

Challenge	Method	Mean PR AUC	Mean F1	Min PR AUC	Max PR AUC
Amplitude Modulation	SAZED	0.875	0.775	0.851	0.888
Amplitude Modulation	SSA	0.872	0.722	0.757	0.946
Amplitude Modulation	Autoperiod	0.817	0.706	0.675	0.901
Amplitude Modulation	Wavelet	0.764	0.860	0.741	0.801
Amplitude Modulation	CFD	0.743	0.711	0.702	0.768
Amplitude Modulation	MatrixProfile	0.733	0.672	0.667	0.769
Amplitude Modulation	ACF	0.726	0.420	0.599	0.803
Amplitude Modulation	STL	0.691	0.614	0.547	0.814
Exponential Trends	Spectral	0.908	0.772	0.832	0.984
Exponential Trends	Variance	0.903	0.752	0.826	0.987
Exponential Trends	SAZED	0.845	0.761	0.806	0.872
Exponential Trends	Wavelet	0.783	0.788	0.764	0.807
Exponential Trends	Autoperiod	0.742	0.615	0.667	0.893
Exponential Trends	Lomb	0.740	0.626	0.636	0.946
Exponential Trends	STL	0.737	0.531	0.619	0.814
Exponential Trends	AIC	0.737	0.543	0.625	0.949
Exponential Trends	MatrixProfile	0.730	0.722	0.688	0.762
Exponential Trends	FFT	0.639	0.391	0.566	0.731
Exponential Trends	ACF	0.629	0.533	0.511	0.711
Exponential Trends	CFD	0.601	0.340	0.580	0.620
Exponential Trends	SSA	0.517	0.228	0.505	0.539
Linear Trends	Spectral	0.826	0.605	0.808	0.853
Linear Trends	Variance	0.821	0.606	0.804	0.846
Linear Trends	Wavelet	0.807	0.648	0.798	0.817
Linear Trends	SAZED	0.735	0.699	0.667	0.848
Linear Trends	MatrixProfile	0.728	0.544	0.670	0.781
Linear Trends	ACF	0.667	0.571	0.667	0.667
Linear Trends	Autoperiod	0.667	0.571	0.667	0.667
Linear Trends	STL	0.662	0.561	0.599	0.753
Linear Trends	AIC	0.648	0.493	0.637	0.655
Linear Trends	Lomb	0.648	0.465	0.639	0.654
Linear Trends	FFT	0.633	0.599	0.608	0.649

Challenge	Method	Mean PR AUC	Mean F1	Min PR AUC	Max PR AUC
Linear Trends	CFD	0.619	0.626	0.602	0.631
Linear Trends	SSA	0.490	0.362	0.489	0.491
Logarithmic Trends	Spectral	0.963	0.882	0.944	0.982
Logarithmic Trends	Variance	0.961	0.895	0.936	0.984
Logarithmic Trends	SAZED	0.862	0.784	0.861	0.863
Logarithmic Trends	Lomb	0.840	0.677	0.765	0.926
Logarithmic Trends	AIC	0.821	0.651	0.735	0.922
Logarithmic Trends	STL	0.802	0.624	0.787	0.813
Logarithmic Trends	Wavelet	0.773	0.862	0.765	0.779
Logarithmic Trends	Autoperiod	0.740	0.399	0.667	0.844
Logarithmic Trends	MatrixProfile	0.739	0.459	0.729	0.750
Logarithmic Trends	FFT	0.574	0.466	0.551	0.603
Logarithmic Trends	CFD	0.554	0.466	0.546	0.569
Logarithmic Trends	ACF	0.505	0.207	0.485	0.541
Logarithmic Trends	SSA	0.504	0.311	0.501	0.510
Outliers	Variance	0.957	0.882	0.933	0.979
Outliers	AIC	0.937	0.804	0.925	0.956
Outliers	Spectral	0.936	0.831	0.911	0.965
Outliers	FFT	0.907	0.772	0.879	0.940
Outliers	Lomb	0.890	0.715	0.855	0.930
Outliers	SSA	0.850	0.682	0.820	0.877
Outliers	Autoperiod	0.838	0.733	0.813	0.863
Outliers	SAZED	0.806	0.610	0.791	0.829
Outliers	Wavelet	0.740	0.848	0.714	0.767
Outliers	MatrixProfile	0.711	0.555	0.674	0.774
Outliers	CFD	0.673	0.653	0.659	0.688
Outliers	STL	0.658	0.427	0.612	0.702
Outliers	ACF	0.652	0.527	0.636	0.671
Piecewise (V-Shape)	Variance	0.994	0.961	0.992	0.995
Piecewise (V-Shape)	Spectral	0.990	0.943	0.989	0.991
Piecewise (V-Shape)	Wavelet	0.934	0.939	0.846	0.990
Piecewise (V-Shape)	SAZED	0.897	0.812	0.895	0.900
Piecewise (V-Shape)	AIC	0.893	0.789	0.813	0.961
Piecewise (V-Shape)	Lomb	0.866	0.629	0.768	0.948
Piecewise (V-Shape)	MatrixProfile	0.699	0.547	0.681	0.709
Piecewise (V-Shape)	Autoperiod	0.697	0.458	0.667	0.756
Piecewise (V-Shape)	STL	0.695	0.384	0.609	0.777
Piecewise (V-Shape)	CFD	0.598	0.391	0.590	0.602
Piecewise (V-Shape)	ACF	0.535	0.174	0.478	0.576
Piecewise (V-Shape)	FFT	0.496	0.323	0.486	0.514

Challenge	Method	Mean PR AUC	Mean F1	Min PR AUC	Max PR AUC
Piecewise (V-Shape)	SSA	0.488	0.239	0.483	0.498
Quadratic Trends	Spectral	0.920	0.785	0.863	0.984
Quadratic Trends	Variance	0.915	0.792	0.856	0.984
Quadratic Trends	SAZED	0.858	0.782	0.844	0.872
Quadratic Trends	Wavelet	0.815	0.826	0.785	0.838
Quadratic Trends	STL	0.781	0.553	0.741	0.811
Quadratic Trends	MatrixProfile	0.779	0.601	0.763	0.788
Quadratic Trends	Lomb	0.734	0.558	0.631	0.928
Quadratic Trends	AIC	0.733	0.510	0.634	0.927
Quadratic Trends	Autoperiod	0.724	0.554	0.667	0.838
Quadratic Trends	ACF	0.621	0.295	0.485	0.710
Quadratic Trends	CFD	0.586	0.343	0.574	0.606
Quadratic Trends	FFT	0.580	0.386	0.564	0.591
Quadratic Trends	SSA	0.515	0.254	0.499	0.538
Red Noise (AR1)	Variance	0.933	0.800	0.913	0.954
Red Noise (AR1)	Spectral	0.912	0.798	0.891	0.930
Red Noise (AR1)	AIC	0.863	0.686	0.840	0.888
Red Noise (AR1)	Lomb	0.847	0.649	0.836	0.860
Red Noise (AR1)	FFT	0.827	0.623	0.744	0.865
Red Noise (AR1)	Autoperiod	0.778	0.676	0.756	0.798
Red Noise (AR1)	SSA	0.774	0.627	0.622	0.838
Red Noise (AR1)	SAZED	0.773	0.678	0.725	0.833
Red Noise (AR1)	Wavelet	0.720	0.831	0.697	0.750
Red Noise (AR1)	CFD	0.679	0.494	0.597	0.730
Red Noise (AR1)	STL	0.646	0.588	0.603	0.703
Red Noise (AR1)	ACF	0.643	0.354	0.538	0.738
Red Noise (AR1)	MatrixProfile	0.628	0.488	0.596	0.678
Shape Robustness	Variance	0.985	0.923	0.978	0.994
Shape Robustness	Spectral	0.977	0.917	0.970	0.989
Shape Robustness	AIC	0.972	0.887	0.956	0.991
Shape Robustness	FFT	0.962	0.871	0.931	0.991
Shape Robustness	Lomb	0.955	0.843	0.921	0.988
Shape Robustness	SSA	0.936	0.810	0.906	0.963
Shape Robustness	Autoperiod	0.919	0.787	0.875	0.972
Shape Robustness	SAZED	0.895	0.768	0.878	0.921
Shape Robustness	STL	0.846	0.635	0.770	0.914
Shape Robustness	ACF	0.835	0.549	0.777	0.877
Shape Robustness	CFD	0.761	0.528	0.736	0.811
Shape Robustness	Wavelet	0.728	0.843	0.719	0.746
Shape Robustness	MatrixProfile	0.727	0.591	0.666	0.772

Challenge	Method	Mean PR AUC	Mean F1	Min PR AUC	Max PR AUC
Sinusoidal Trends	Variance	0.993	0.962	0.993	0.993
Sinusoidal Trends	Spectral	0.990	0.965	0.989	0.990
Sinusoidal Trends	AIC	0.919	0.817	0.814	0.985
Sinusoidal Trends	Lomb	0.915	0.819	0.818	0.970
Sinusoidal Trends	Wavelet	0.865	0.917	0.769	0.984
Sinusoidal Trends	SAZED	0.865	0.793	0.864	0.865
Sinusoidal Trends	STL	0.783	0.649	0.743	0.809
Sinusoidal Trends	Autoperiod	0.766	0.503	0.667	0.895
Sinusoidal Trends	MatrixProfile	0.722	0.559	0.703	0.743
Sinusoidal Trends	FFT	0.612	0.532	0.496	0.822
Sinusoidal Trends	CFD	0.608	0.469	0.576	0.672
Sinusoidal Trends	ACF	0.577	0.559	0.482	0.632
Sinusoidal Trends	SSA	0.505	0.367	0.484	0.538
Step (Level Shift)	Variance	0.899	0.747	0.851	0.961
Step (Level Shift)	Spectral	0.896	0.748	0.849	0.956
Step (Level Shift)	SAZED	0.857	0.666	0.849	0.871
Step (Level Shift)	AIC	0.813	0.723	0.712	0.917
Step (Level Shift)	Lomb	0.797	0.417	0.697	0.899
Step (Level Shift)	STL	0.791	0.527	0.760	0.820
Step (Level Shift)	MatrixProfile	0.760	0.603	0.751	0.769
Step (Level Shift)	Wavelet	0.699	0.713	0.649	0.754
Step (Level Shift)	Autoperiod	0.667	0.571	0.667	0.667
Step (Level Shift)	ACF	0.648	0.253	0.500	0.740
Step (Level Shift)	CFD	0.588	0.543	0.565	0.606
Step (Level Shift)	FFT	0.558	0.469	0.512	0.599
Step (Level Shift)	SSA	0.528	0.399	0.500	0.553

Challenge Robustness Heatmap
Mean PR AUC across challenge parameters

Method	Amplitude Modulation	Exponential Trends	Linear Trends	Logarithmic Trends	Outliers	Piecewise (V~Shape)	Quadratic Trends	Red Noise (AR1)	Shape Robustness	Sinusoidal Trends	Step (Level Shift)
Wavelet	0.76	0.78	0.81	0.77	0.74	0.93	0.81	0.72	0.73	0.86	0.70
Variance	0.97	0.90	0.82	0.96	0.96	0.99	0.92	0.93	0.98	0.99	0.90
STL	0.69	0.74	0.66	0.80	0.66	0.70	0.78	0.65	0.85	0.78	0.79
SSA	0.87	0.52	0.49	0.50	0.85	0.49	0.52	0.77	0.94	0.50	0.53
Spectral	0.95	0.91	0.83	0.96	0.94	0.99	0.92	0.91	0.98	0.99	0.90
SAZED	0.87	0.84	0.73	0.86	0.81	0.90	0.86	0.77	0.89	0.86	0.86
MatrixProfile	0.73	0.73	0.73	0.74	0.71	0.70	0.78	0.63	0.73	0.72	0.76
Lomb	0.92	0.74	0.65	0.84	0.89	0.87	0.73	0.85	0.95	0.91	0.80
FFT	0.94	0.64	0.63	0.57	0.91	0.50	0.58	0.83	0.96	0.61	0.56
CFD	0.74	0.60	0.62	0.55	0.67	0.60	0.59	0.68	0.76	0.61	0.59
Autoperiod	0.82	0.74	0.67	0.74	0.84	0.70	0.72	0.78	0.92	0.77	0.67
AIC	0.96	0.74	0.65	0.82	0.94	0.89	0.73	0.86	0.97	0.92	0.81
ACF	0.73	0.63	0.67	0.50	0.65	0.54	0.62	0.64	0.84	0.58	0.65

Challenge Scenario

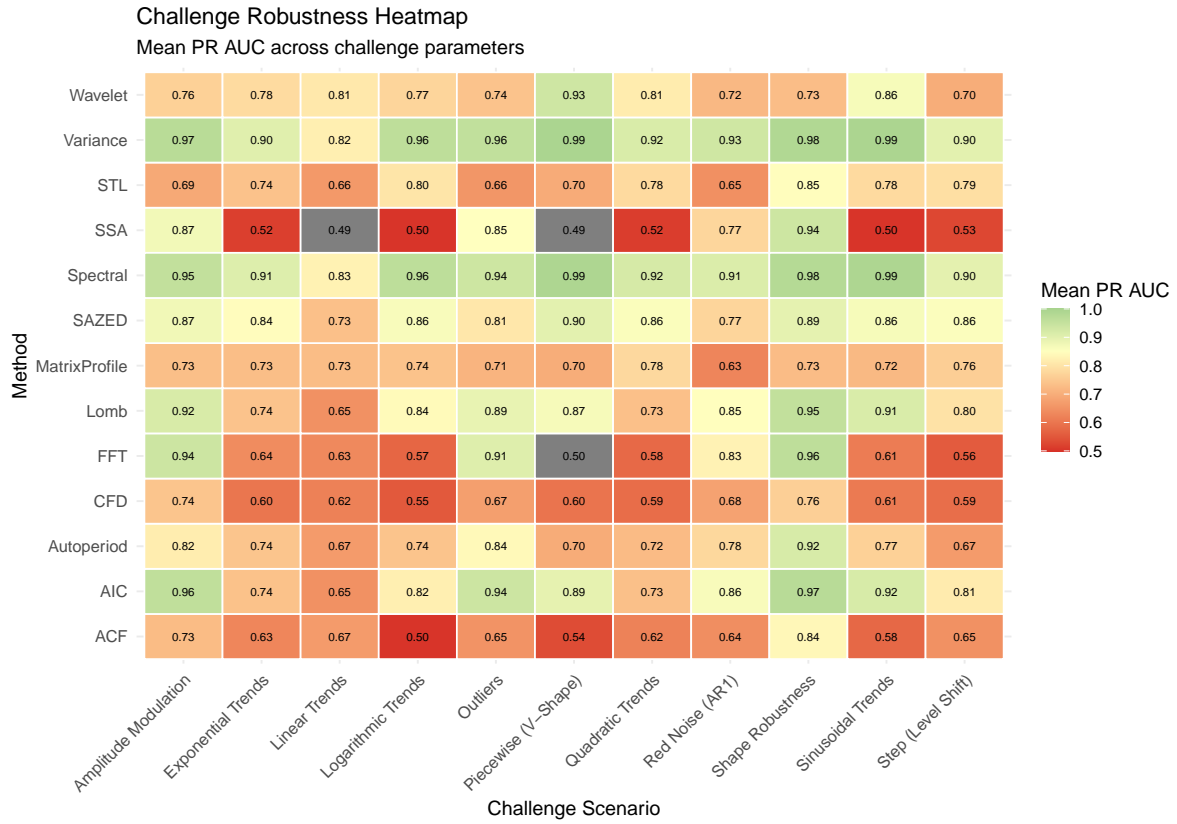Mean PR AUC: 1.0 / 0.9 / 0.8 / 0.7 / 0.6 / 0.5

Figure 29: Challenge robustness heatmap: Mean PR AUC by method and challenge type

M4 Real-World Validation

To validate synthetic benchmark results, we apply all detection methods to real-world time series from the M4 Competition Monthly dataset. Monthly series are expected to exhibit annual seasonality (period = 12).

```
Note: M4 validation skipped - M4comp2018 R package not available
Install with: install.packages('M4comp2018', repos = 'https://cloud.r-project.org')
```

Failure Analysis: Corner Cases

Understanding when the best-performing method fails provides actionable insights for practitioners. We analyze cases where the Variance method (highest F1) made incorrect predictions.

False Negatives: Missed Seasonality

These are seasonal series that Variance failed to detect. False negatives are particularly costly in forecasting, as missing seasonality leads to systematically biased predictions.
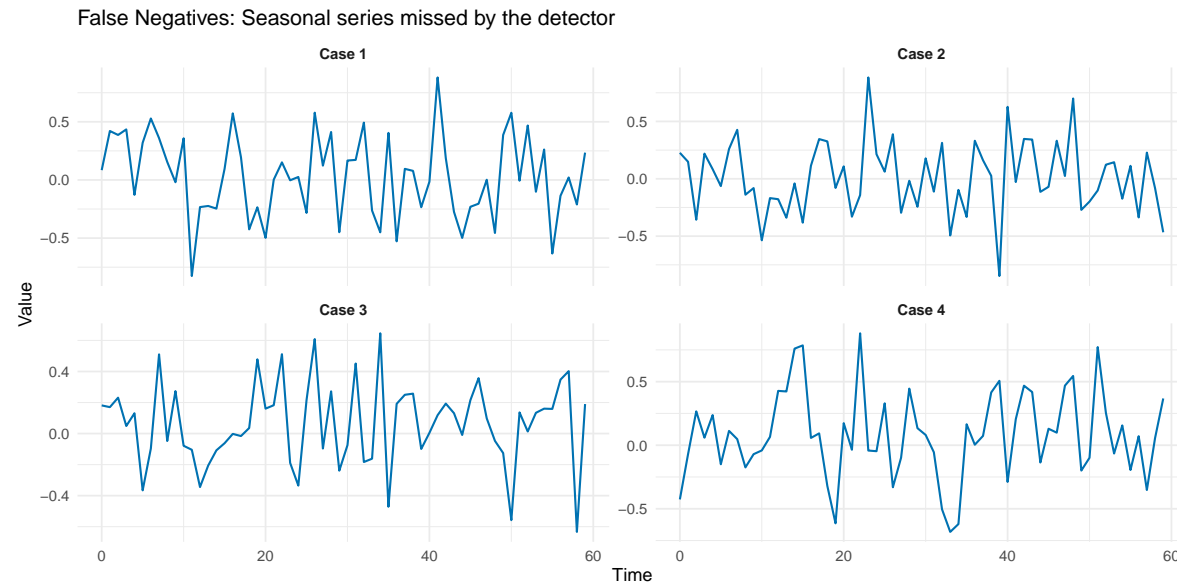

Figure 30: False Negative Examples: Seasonal patterns missed by Variance method. Common causes include low signal-to-noise ratio, irregular amplitudes, or dominant trends masking the seasonal component.

False Positives: Hallucinated Seasonality

These are non-seasonal series incorrectly classified as seasonal. While less critical than false negatives in forecasting, high false positive rates indicate the method may be over-sensitive.
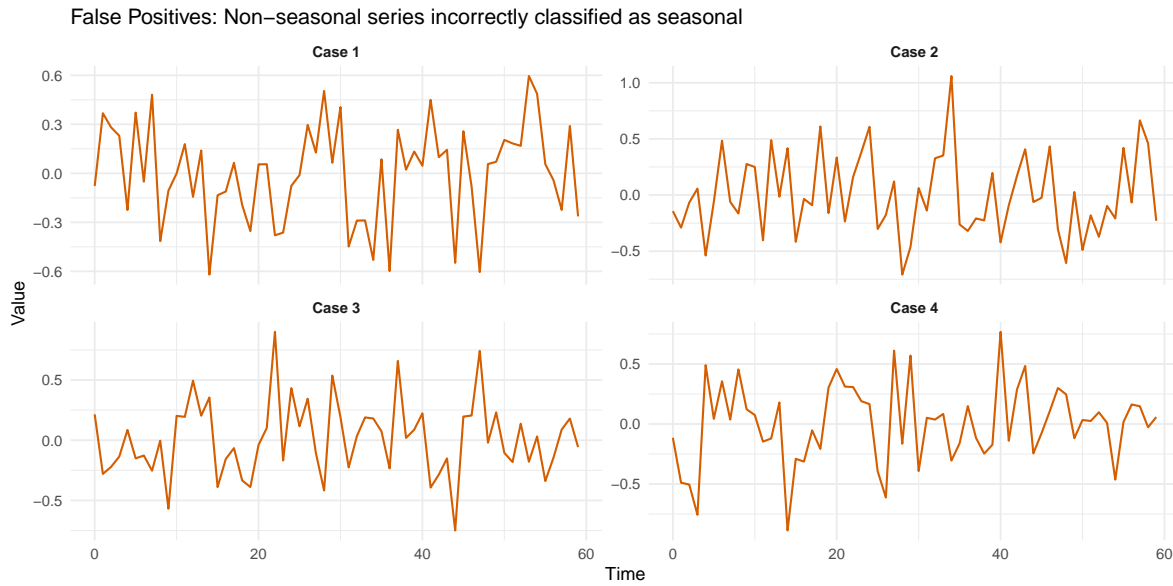

Figure 31: False Positive Examples: Non-seasonal series incorrectly classified as seasonal by Variance method. These cases often contain trends or autocorrelation patterns that mimic periodicity.

Summary and Conclusions

Final Rankings

Table 9: Final Method Rankings by F1 Score

Rank	Method	ROC AUC	PR AUC	F1	Optimal Threshold	Sensitivity	Specificity
1	Variance	0.962	0.992	0.936	0.215	0.896	0.92
2	Wavelet	0.608	0.852	0.918	0.991	0.996	0.22
3	Spectral	0.952	0.989	0.898	0.335	0.822	0.96
4	AIC	0.937	0.987	0.898	0.213	0.822	0.96
5	FFT	0.935	0.986	0.894	0.063	0.816	0.96
6	Lomb	0.931	0.985	0.877	0.570	0.787	0.97
7	SAZED	0.858	0.956	0.859	0.794	0.773	0.88
8	Autoperiod	0.863	0.969	0.831	0.202	0.727	0.90
9	SSA	0.892	0.976	0.831	0.425	0.713	0.98
10	STL	0.801	0.954	0.747	0.501	0.607	0.92
11	MatrixProfile	0.719	0.923	0.726	0.229	0.604	0.73
12	ACF	0.782	0.950	0.725	0.268	0.571	0.98
13	CFD	0.738	0.936	0.615	0.268	0.447	0.97

Key Findings

Best Overall Method: Variance (F1 = 0.936, ROC AUC = 0.962, PR AUC = 0.992)

Recommendations

Use Case	Recommended Method	Rationale
General detection	Wavelet or Variance	Highest F1 scores across challenges
Quick screening	FFT	Fast with good accuracy on clean data
Noisy data	ACF or Autoperiod	Robust to AR(1) autocorrelated noise
Data with trends	CFD-Autoperiod	Uses differencing to remove trends
Irregular sampling	Lomb-Scargle	Handles gaps and uneven spacing

Use Case	Recommended Method	Rationale
Non-stationary amplitude	Wavelet or SSA	Time-frequency localization
Non-sinusoidal patterns	Autoperiod or ACF	Less sensitive to harmonic content
Emerging seasonality	Wavelet	Detects time-localized patterns

Cleanup

```
[1] 0
```

```
[1] 0
```

```
[1] 0
```

Session Info

```
R version 4.5.2 (2025-10-31)
Platform: x86_64-pc-linux-gnu
Running under: Manjaro Linux

Matrix products: default
BLAS:   /usr/lib/libblas.so.3.12.0
LAPACK: /usr/lib/liblapack.so.3.12.0  LAPACK version 3.12.0

locale:
 [1] LC_CTYPE=de_DE.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=de_DE.UTF-8
 [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=de_DE.UTF-8
 [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C

time zone: Europe/Berlin
tzcode source: system (glibc)

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base
```

```
other attached packages:
 [1] PRROC_1.4      rlang_1.1.6    pROC_1.19.0.1 scales_1.4.0  knitr_1.51
 [6] purrr_1.2.0    tidyr_1.3.2    dplyr_1.1.4    ggplot2_4.0.1 duckdb_1.4.3
[11] DBI_1.2.3

loaded via a namespace (and not attached):
 [1] gtable_0.3.6       jsonlite_2.0.0    compiler_4.5.2    tidyselect_1.2.1
 [5] Rcpp_1.1.0         yaml_2.3.12       fastmap_1.2.0     R6_2.6.1
 [9] labeling_0.4.3     generics_0.1.4    tibble_3.3.0      pillar_1.11.1
[13] RColorBrewer_1.1-3 xfun_0.54         S7_0.2.0          otel_0.2.0
[17] cli_3.6.5          withr_3.0.2       magrittr_2.0.4    digest_0.6.39
[21] grid_4.5.2         lifecycle_1.0.4   vctrs_0.6.5       evaluate_1.0.5
[25] glue_1.8.0         farver_2.1.2      codetools_0.2-20  rmarkdown_2.30
[29] tools_4.5.2        pkgconfig_2.0.3   htmltools_0.5.9
```