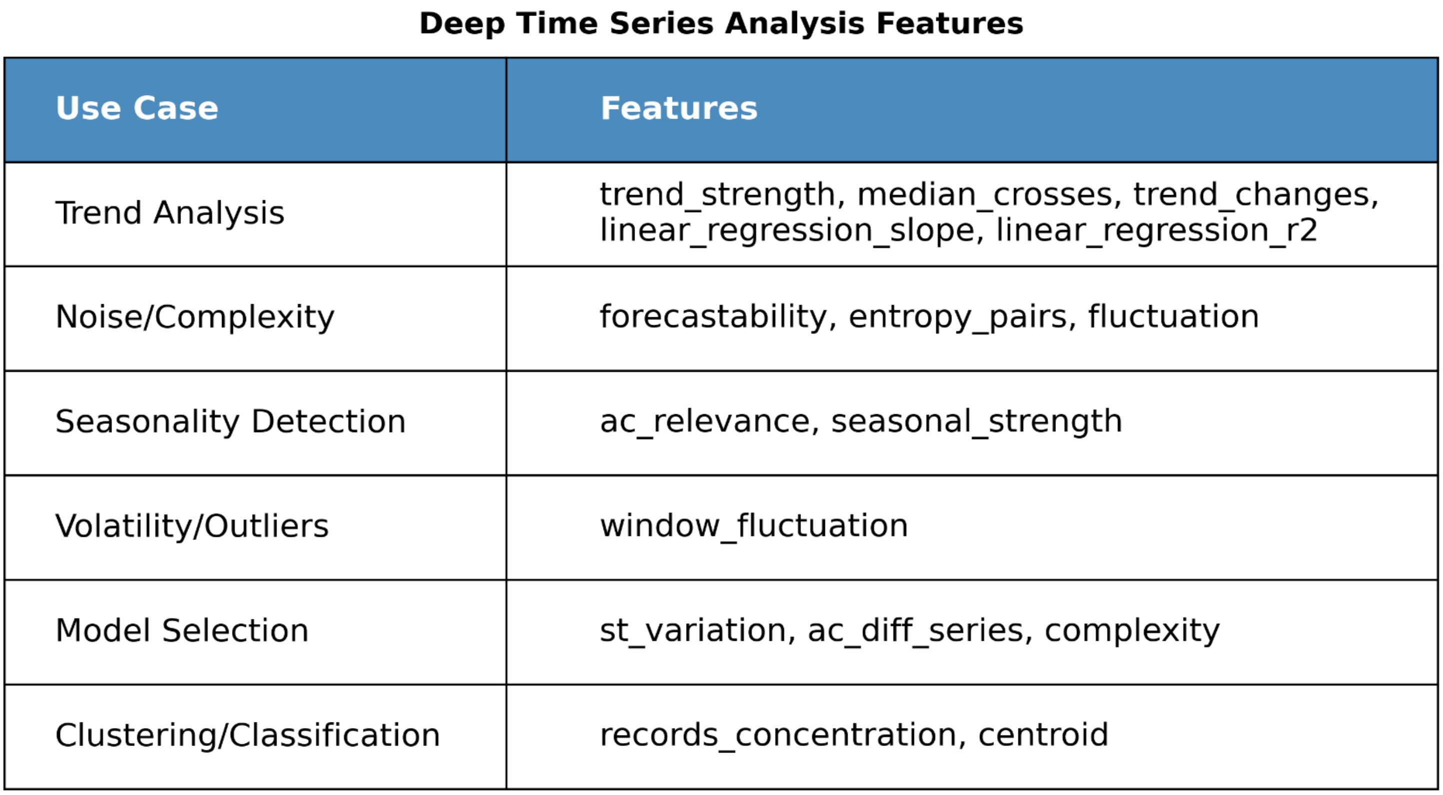
Deep Time-Series Analysis:

Features



Deep Time Series Analysis Features

Use Case	Features
Trend Analysis	trend_strength, median_crosses, trend_changes, linear_regression_slope, linear_regression_r2
Noise/Complexity	forecastability, entropy_pairs, fluctuation
Seasonality Detection	ac_relevance, seasonal_strength
Volatility/Outliers	window_fluctuation
Model Selection	st_variation, ac_diff_series, complexity
Clustering/Classification	records_concentration, centroid

Trend Analysis

- **Old: stl_features (trend_strenght)**
- **New: trend_strength (Trend Strength)**

Source: tsfeatures

Description: Computes the strength of trend in a time series

Formula: $\text{trend} = 1 - \frac{\text{Var}(e_t)}{\text{Var}(f_t + e_t)}$ and

- **Old: crossing_points**
- **New: median_crosses (Median Crosses)**

Source: tsfeatures

Description: Computes the number of times a time series crossed the median line

Formula:

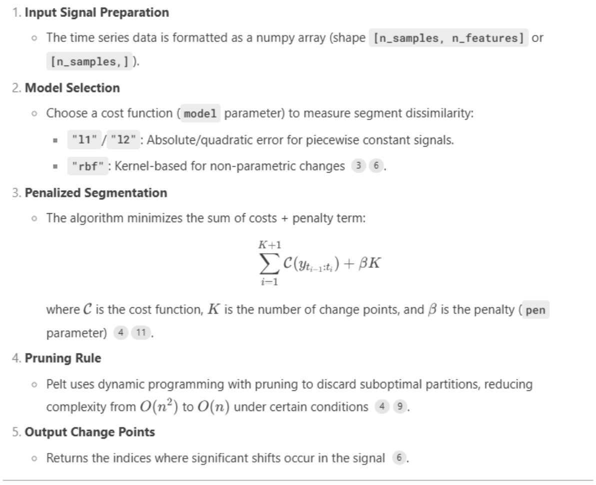
1. Compute the median line
2. Return the number of times the series values cross the median line

- **Old: Pelt**
- **New: trend_changes (Trend Changes)**

Source: ruptures

Description: Detection of trend changing points: Linearly penalized segmentation

Formula:



1. **Input Signal Preparation**
   - The time series data is formatted as a numpy array (shape `[n_samples, n_features]` or `[n_samples,]`).
2. **Model Selection**
   - Choose a cost function (`model` parameter) to measure segment dissimilarity:
     - `"l1"` / `"l2"`: Absolute/quadratic error for piecewise constant signals.
     - `"rbf"`: Kernel-based for non-parametric changes [3] [6].
3. **Penalized Segmentation**
   - The algorithm minimizes the sum of costs + penalty term:

     $$\sum_{i=1}^{K+1} \mathcal{C}(y_{t_{i-1}:t_i}) + \beta K$$

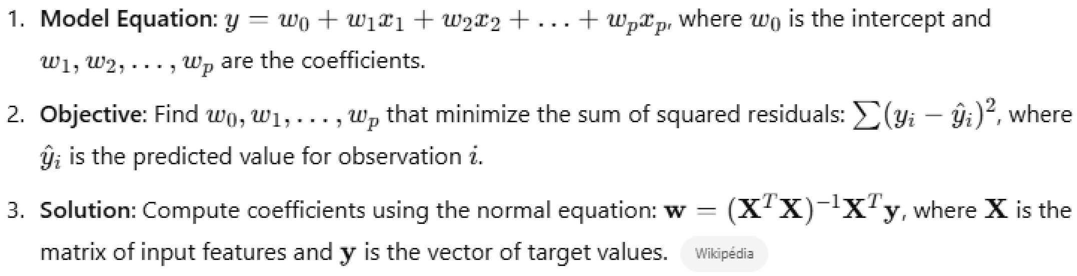
     where $\mathcal{C}$ is the cost function, K is the number of change points, and β is the penalty (`pen` parameter) [4] [11].
4. **Pruning Rule**
   - Pelt uses dynamic programming with pruning to discard suboptimal partitions, reducing complexity from $O(n^2)$ to $O(n)$ under certain conditions [4] [9].
5. **Output Change Points**
   - Returns the indices where significant shifts occur in the signal [6].

- **Old: Linear Regression**
- **New: linear_regression_slope (Linear Regression Slope) & linear_regression_r2 (Linear Regression R2)**

Source: scikit-learn

Description: Computes the linear regression of a time series using the Ordinary Least Squares

Formula:



1. **Model Equation**: $y = w_0 + w_1x_1 + w_2x_2 + \ldots + w_px_p$, where w_0 is the intercept and $w_1, w_2, \ldots, w_p$ are the coefficients.
2. **Objective**: Find $w_0, w_1, \ldots, w_p$ that minimize the sum of squared residuals: $\sum(y_i - \hat{y}_i)^2$, where $\hat{y}_i$ is the predicted value for observation i.
3. **Solution**: Compute coefficients using the normal equation: $\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$, where $\mathbf{X}$ is the matrix of input features and $\mathbf{y}$ is the vector of target values. Wikipédia

Noise/Complexity

- **Old: special_entropy**
- **New: forecastability (Series Forecastability)**

Source: antropy

Description: Forecastability measure of a time-series (Shannon entropy)

Formula: $-\int_{-\pi}^{\pi} \hat{f}(\lambda) \log \hat{f}(\lambda)\, d\lambda,$

- **Old: entropy_pairs**
- **New: entropy_pairs (Entropy Pairs)**

Source: catch22

Description: Computes the probability of a two letter sequence, being the possible letters A, B and C, that represent the third lowest values of the series, the third median values and the third highest values

Formula:

1. Converts each value in the time series into one of three symbols ('A', 'B', or 'C') using an equi-probable binning in which the lowest 3rd of values are assigned 'A', the middle 3rd 'B', and the highest 3rd of values are given 'C'.
2. It then analyses the probabilities of all two-letter sequences ('AA', 'AB', 'BB', ...) and outputs the entropy of this set of probabilities.

- **Old: high_fluctuation**
- **New: fluctuation (Series Fluctuation)**

New Name: Series Fluctuation (fluctuation)

Source: catch22

Description: Computes the proportion of difference magnitudes that are greater than 4% of the standard deviation of the time series.

Seasonality Detection

- **Old: acf_timescale**
- **New: ac_relevance (Autocorrelation Relevance)**

Source: catch22

Description: Captures the first 1/e crossing of the auto-correlation function. Measures the first time lag at which the autocorrelation function drops below 1/e.

Formula:

1. Compute the autocorrelation function
2. Return the first 1/e crossing

- **Old: stl_features (seasonal_strenght)**
- **New: seasonal_strength (Seasonal Strength)**

Source: tsfeatures

Description: Computes the strength of trend in a time series

Formula:

$$\text{seasonal.strength} = 1 - \frac{\operatorname{Var}\left(e_t\right)}{\operatorname{Var}\left(s_{i,t} + e_t\right)}$$

Volatility/Outliers

- **Old: rs_range**
- **New: window_fluctuation (Window Fluctuation)**

Source: catch22

Description: Computes the fluctuation in defined timescale window

Formula:

1. Compute a cumulative sum of the time series
2. Compute the level of fluctuation (e.g., root-mean-square deviations from local low-order trends) across windows corresponding to a given timescale. Different methods exist for detrending time-series windows at a given timescale, including (relevant to these two features):
3. Rescaled range analysis removes a line connecting the endpoints of each window and computes the range of the remaining points (Caccia et al., Physica A, 1997)
4. DFA fits a k-order polynomial to each window and computes the residuals from this fit.
5. Looks for linear scaling in the log(timescale)–log(fluctuation) plot.

Model Selection

- **Old: Trev**
- **New: st_variation (Short-Term Variation)**

Source: catch22

Description: Computes the average across the time series of the cube of successive time-series differences.

Formula:

$\langle (xi+1-xi)3 \rangle t$, for time-series values x, at all time points, t

- **Old: acf_feature (x_acf1)**
- **New: ac (Autocorrelation)**

Source: tsfeatures

Description: Returns the autocorrelation function.

Formula:

1. Compute and return the autocorrelation

- **Old: acf_feature (diff1_ac10)**
- **New: diff_series (Differenced Series)**

Source: tsfeatures

Description: Returns the autocorrelation of the differenced series.

Formula:

1. Compute the differenced series
2. Compute and return autocorrelation of the differenced series

- **New: complexity (Series Complexity)**

Source: NEW

Description: Computes the complexity of a time-series to help choosing the parameters in nearest-neighbor algorithms (using CIDM)

Formula:

$$CE(Q) = \sqrt{\sum_{i=1}^{n-1}(q_i - q_{i+1})^2}$$

Clustering/Classification

- **Old: mode10**
- **New: rec_concentration (Records Concentration)**

Source: catch22

Description: Captures the relative position of the most probable value in relation to the mean

Formula:

1. z-score the input time series.
2. Compute a histogram using 10 bins.
3. Return the location of the bin with the most counts.

- **Old: calc_centroid**
- **New: centroid (Series Centroid)**

New Name: Series Centroid (centroid)

Source: tsfel

Description: Computes the centroid along the time axis

Formula:

$$C_t = \frac{\sum_{i=1}^{N} t_i \cdot |y_i|}{\sum_{i=1}^{N} |y_i|}$$