Enhanced Cycle Scanner with Drift Handling: Methodology and Innovations

By Eduard Samokhvalov, Inspired by Lars von Thienen

2025-04-27

Abstract

This white paper presents an advanced implementation of the Enhanced Cycle Scanner Algorithm for financial time series, with a focus on Bitcoin price data. Building on the foundational work of Lars von Thienen, this implementation introduces robust handling of trend, evolving seasonality (drift), and multi-timeframe analysis, leveraging modern statistical and computational techniques. The result is a more adaptive, scalable, and statistically rigorous approach to cycle detection and validation in non-stationary financial markets.

Table of Contents

1. Introduction

Cycle analysis is a powerful tool in financial time series analysis, enabling the detection of recurring patterns and rhythms in market data. Lars von Thienen's work on the Bartels test and dominant cycle detection has been influential in this field. However, real-world financial data often exhibit non-stationarity, evolving seasonality, and structural drift, which require more robust and adaptive methods. This white paper details an enhanced cycle scanner that addresses these challenges by combining state-of-the-art statistical modeling, robust cycle validation, and scalable computation.

2. Background: Lars von Thienen's Method

Lars von Thienen's approach combines spectral analysis (FFT) with the Bartels test to identify and validate dominant cycles in financial data.

Bartels Test:

The Bartels test is a non-parametric test for randomness. For a time series $x_1, x_2, \ldots, x_n$, the Bartels statistic is:

$$RVN = \sum_{i=1}^{n-1} (x_{i+1} - x_i)^2 / \sum_{i=1}^{n} (x_i - \mathrm{mean}(x))^2$$

A low RVN value suggests non-randomness, indicating the presence of cycles.

Classic Cycle Detection:

- Detrend the series
- Apply FFT to find dominant frequencies
- Use the Bartels test to validate cycles

Limitations:

- Assumes stationarity
- Handles only static seasonality
- Not robust to drift or evolving cycles

3. Mathematical Formulation

State Space Model (SSM):

A time series y_t can be decomposed as:

$$y_t = T_t + S_t + C_t + \epsilon_t$$

Where:

- T_t: Trend component
- S_t: Seasonal component (can drift)
- C_t: Cyclic component
- ϵ_t: Noise

SSM Equations:

- Trend: T_{t+1} = T_t + eta_t

- Seasonality: S_{t+1} = S_t + zeta_t

Where eta_t, zeta_t are Gaussian noise terms.

Dynamic Harmonic Regression (DHR):

DHR models seasonality as a sum of time-varying Fourier terms:

S_t = sum_{k=1}^K [a_k(t) * cos(2*pi*k*t/P) + b_k(t) * sin(2*pi*k*t/P)]

Where a_k(t), b_k(t) can evolve over time.

Fast Fourier Transform (FFT):

FFT is used to identify dominant frequencies in the residuals after removing trend and seasonality.

Sustainability Score:

For each timeframe:

Sustainability Score = Significance * Normalized Strength

Where:

- Significance: 1 - Bartels RVN

- Normalized Strength: Relative power of the dominant cycle

4. Justification for Methodological Enhancements

The enhancements in this implementation are motivated by best practices and recent advances in time series analysis:

- **State Space Models (SSM):** Widely used for modeling non-stationary and evolving processes ([Durbin & Koopman, 2012](https://www.springer.com/gp/book/9780199641178)). SSMs allow for explicit modeling of trend and time-varying seasonality, which is crucial for financial data subject to drift and regime changes.

- **Dynamic Harmonic Regression (DHR):** Recommended for handling complex, drifting seasonal patterns

([Hyndman et al., 2008](https://otexts.com/fpp2/dhr.html)). DHR enables the seasonal component to adapt over time, capturing evolving market rhythms.

- **FFT and Bartels Test:** The combination of spectral analysis and non-parametric randomness testing is a best practice for robust cycle detection ([von Thienen, 2015](https://www.cycle-trader.com/), [Gencay et al., 2001](https://www.springer.com/gp/book/9780122796707)).

- **Parallelization:** Modern computational best practices recommend parallel processing for large-scale, multi-timeframe analysis ([McKinney, 2010](https://conference.scipy.org/proceedings/scipy2010/pdfs/mckinney.pdf)).

- **Reproducibility and Automation:** All data, code, and results are cached and saved for reproducibility, following open science principles.

These choices are supported by the academic literature and are considered best practice in both econometrics and quantitative finance.

5. Enhanced Methodology

Data Acquisition & Preprocessing:

- Download all available 1-minute Bitcoin price data since 2020
- Cache raw data locally
- Resample to arbitrary timeframes using pandas

Decomposition:

- Use SSM and DHR to decompose log-price into trend, seasonality (with drift), and residuals
- Allow the seasonal component to evolve over time (drift handling)

Cycle Detection:

- Apply FFT to the residuals

- Validate cycles using the Bartels test
- Rank cycles by significance and strength

Multi-Timeframe & Parallel Analysis:

- Analyze a wide range of timeframes in parallel
- Rank by sustainability score
- Generate interactive HTML reports and a summary heatmap

Reporting:

- All results are saved in a dedicated 'reports/' folder

6. Implementation Details

- Python Libraries: statsmodels, pandas, scipy, plotly, fpdf, numpy
- Parallelization: ProcessPoolExecutor for multi-core analysis
- Caching: All 1-minute data is cached locally
- Reproducibility: All reports and the white paper are saved in the 'reports/' folder
- Automation: No manual intervention required after setup

Example: Resampling Code

```
import pandas as pd
df_1min = pd.read_csv('btc_1min_cache.csv', parse_dates=['datetime'])
df_1h = df_1min.set_index('datetime').resample('1H').agg({'open':'first','high':'max','low':'min','close':'last','volume':'sum'}).dropna()
```

7. Results & Interpretation

- The Enhanced Cycle Scanner identifies and ranks the most sustainable cycles for each timeframe
- The summary heatmap and top-ranked timeframes highlight where cyclic structure is most robust
- Reports are interactive and easy to interpret

How to Read the Reports:

- Panel 1: Decomposition (trend, seasonality, log-price)

- Panel 2: Seasonality drift

- Panel 3: Cycle forecast and projection

- Panel 4: Periodogram

- Summary heatmap: Sustainability across timeframes

8. Comparative Summary: Strengths and Weaknesses

Aspect	Lars von Thienen's Method	Enhanced Cycle Scanner
Trend Handling	Simple detrending	State Space Model (SSM) for robust trend extraction
Seasonality	Static, fixed period	Dynamic, drift-capable (SSM/DHR)
Cycle Validation	Bartels test only	Bartels test + power ranking
Drift Handling	Not handled	Explicitly modeled in SSM/DHR
Multi-Timeframe	Manual, slow	Automated, parallelized
Reporting	Manual, limited	Automated, interactive HTML & PDF
Reproducibility	Not emphasized	Full caching, reproducible workflow
Scalability	Limited	Highly scalable (multi-core)
Weaknesses	Sensitive to non-stationarity, drift	Complexity, requires more computation

9. References

- Lars von Thienen, 'The Bartels Test and the Dominant Cycle', https://www.cycle-trader.com/

- Durbin, J., & Koopman, S. J. (2012). Time Series Analysis by State Space Methods. Oxford University Press.

- Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice. https://otexts.com/fpp2/dhr.html

- Gencay, R., Selcuk, F., & Whitcher, B. (2001). An Introduction to Wavelets and Other Filtering Methods in Finance and Economics. Academic Press.

- McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference.

- Statsmodels, pandas, scipy, plotly, fpdf documentation

- [Add further references as needed]