Wavelet21 Optimization Plan — Steps (0) Residual-First & (A) Period-Aware Contrasts

Generated on 2025-09-16 18:14

Overview

- Goal: Implement (0) residual-first pipeline and (A) period-aware wavelet contrasts to produce richer, leakage-safe, and informative features for the structural-break classifier.
- Scope: Minimal, surgical edits in methods/wavelet21/, plus small notebook hooks in GeneratingFeatures2.ipynb and Model_X.ipynb.

Files to touch

- methods/wavelet21/feature_extractor.py – add H0 (null) modeling, compute MODWT on standardized residuals, add period-aware contrasts.
- methods/wavelet21/config.py – add NullModelCfg defaults and attach to WaveletConfig.
- methods/wavelet21/batch_processor.py – pass meta diagnostics through to output (optional light change).
- (optional) methods/base/utils.py – place helper diagnostics if you want sharing across methods.

(0) Residual-first pipeline – Logic (brief)

- Fit a single H0 on the full series (Period0+1), obtain standardized residuals $\varepsilon_t = a_t / \hat{\sigma}_t$.

- Run MODWT on ε_t (not on raw prices/levels). This removes predictable trend/volatility so wavelet energy shifts reflect genuine departures from H0 near the boundary.

- Persist light H0 diagnostics as features: Ljung–Box p on ε_t, ARCH-LM proxy p on ε_t^2, chosen error law (Normal vs Student-t) and ν when t.

- This deconfounds dispersion contrasts and stabilizes boundary-local features.

(0) Step-by-step Implementation

1) Add a null-model configuration

   - dataclass NullModelCfg(model='arima', arima_order=(1,0,1), resid_law='t', min_len=300).

2) Implement fit_null_model(full_vals, cfg) in feature_extractor.py

   - Default: ARIMA(1,0,1) via statsmodels, statespace fit.

   - Standardize residuals by robust scale (std + 1e-12).

   - Compute diagnostics:

     • Ljung–Box p on ε_t (lags=20).

     • Ljung–Box p on ε_t^2 (proxy for ARCH-LM).

     • Residual law: 't' (with a default ν≈8) or 'normal' according to cfg.

  - Return (eps, meta). If series is short (<min_len), fallback to demean/scale.

3) Use residuals for MODWT

  - In extract_wavelet_features(), call fit_null_model() first, then run MODWT on ε_t for J levels and the selected family (e.g., 'sym4').

4) Pass diagnostics to features

  - Add columns: h0_ljungbox_p, h0_archlm_p, h0_err_is_t, h0_t_nu.

5) Minimal config plumbing

  - In WaveletConfig, add null_model: NullModelCfg().

(A) Period-aware wavelet contrasts — Logic (brief)

- On residual MODWT coefficients W_{j,t}, compare pre vs post dispersion per level j.

- Use continuous contrasts for stability and direction:

  • log variance ratio: log(Var(Wj_post)/Var(Wj_pre))

  • log MAD ratio: log(MAD(Wj_post)/MAD(Wj_pre))

- Optionally add one robust location effect (Hedges’ g) on Wj (post - pre)/s_pooled.

(A) Step-by-step Implementation

1) Period masks once

   - pre_idx = where(period==0), post_idx = where(period==1).

2) Per-level contrasts in feature_extractor.py

   - For each level j:

     • v0, v1 = var(Wj_pre), var(Wj_post) with ddof=1; add 1e-12 for safety.

     • m0, m1 = normalized MAD(Wj_pre/post); add 1e-12.

     • Features:

       wav_{family}_L{j}_var_logratio = log(v1/v0)

       wav_{family}_L{j}_mad_logratio = log(m1/m0)

     • Optional:

       wav_{family}_L{j}_hedges_g

3) Naming convention

   - wav_{family}_L{level}_{stat}, e.g., wav_sym4_L2_mad_logratio.

4) Keep existing boundary-local features (if present)

   - They now operate on residual MODWT; threshold calibration will come later.

Notebook hooks (minimal)

- GeneratingFeatures2.ipynb:
  • Configure WaveletConfig + NullModelCfg (sym4, J=3, alpha=0.05, null_model='arima', resid_law='t').
  • Run small batch (~200 ids) and write Wavelet.parquet/csv and Wavelet_meta.parquet.
  • Sanity cell: describe wavelet logratios and assert no constant columns.

- Model_X.ipynb:
  • No change required to run; you can toggle feature groups later for ablations.

QA checklist

- Wavelet.csv (or parquet) contains:
  • h0_ljungbox_p, h0_archlm_p, h0_err_is_t, h0_t_nu.
  • wav_*_L{j}_var_logratio, wav_*_L{j}_mad_logratio for j=1..J.
- No all-NaN or constant columns in the wavelet block.
- Spot-check a few IDs: residual standardization applied; contrasts populated and finite.

Why this improves AUC/Brier

- Deconfounding: residual-first removes baseline structure; contrasts target true regime shifts.
- Continuous signals: log variance/MAD ratios provide graded evidence that tree models exploit.
- Leakage-safe: single transform across both segments; fair pre/post comparison.

Next steps (after (0) and (A))

- Calibrate boundary exceedances by residual law & window length; enrich cache keys.
- Add db8 and J sweep to 1..3; guard feature growth via correlation pruning and model selection.
- Add Brown–Forsythe/t-test pre/post features; keep isotonic calibration in Model_X.