A Reproducible Comparison of Trend Estimation Methods on S&P 500 Log Prices

Heriberto Espino-Montelongo

June 4, 2026

Abstract

This paper compares a train-validation smoothness selector for penalized trend estimation against classical and simple benchmark methods on historical S&P 500 prices. The study uses the local `trend_estimation` Python library as a reproducible software dependency. Because real financial data do not provide an observed true trend, the empirical comparison is based on temporal out-of-sample forecasting errors and realized roughness diagnostics rather than claims about true-trend recovery. The manuscript also distinguishes Guerrero smoothness, an input index for finite-difference penalized smoothers, from realized roughness, an output diagnostic available for every fitted trend.

1 Introduction

Trend estimation is a basic task in time series analysis, but trend smoothness is rarely known in advance. Standard time-series and forecasting references emphasize that trend extraction, prediction, and structural interpretation are related but distinct goals [Chatfield, 2003, Brockwell and Davis, 2016, Hamilton, 1994, Makridakis et al., 1998]. This motivates procedures that choose smoothness from data while preserving a strict temporal evaluation protocol. This paper compares the proposed train-validation smoothness selection approach with a time-weighted validation variant and several classical or simple benchmarks.

The comparison is implemented as a reproducible subproject that uses `trend_estimation` as a normal Python library dependency [Espino-Montelongo, 2026]. The purpose is not to claim universal superiority from one ticker, but to provide a concrete empirical case study and a template for broader comparisons. The paper is deliberately explicit about terminology: Guerrero smoothness is used only for finite-difference penalized smoothers, whereas realized roughness is reported as a diagnostic for every fitted trend.

2 Methods

The proposed estimator uses penalized least squares. For an observed series $y_1, \ldots, y_n$, the fitted trend $\hat{\tau}$ is obtained by balancing fidelity to the observations against a finite-difference roughness penalty. A typical objective has the form

$$\sum_{t=1}^{n}(y_t - \tau_t)^2 + \lambda \|D^d \tau\|_2^2,$$

where D^d is a difference operator of order d and λ controls smoothness. This formulation follows the finite-difference penalized smoothing tradition associated with Whittaker-Henderson graduation, Guerrero's smoothness index, and modern penalized-spline and perfect-smoother treatments [Whittaker, 1922, Henderson, 1924, Guerrero, 2007, Eilers and Marx, 1996, Eilers, 2003, Wahba, 1990].

The train-validation selector fits candidate penalized trends on the training segment, forecasts the validation segment, and selects the candidate with the lowest validation loss. The time-weighted variant uses a validation loss that can emphasize later validation observations. The baseline methods are an HP-style trend [Hodrick and Prescott, 1997, King and Rebelo, 1993], a Whittaker-Henderson-style penalized trend [Whittaker, 1922, Henderson, 1924], a trailing moving average, simple exponential smoothing, and a polynomial trend forecaster. Moving-average and exponential-smoothing benchmarks are included because they are common forecasting baselines [Brown, 1959, Holt, 2004, Winters, 1960, Gardner, 1985, 2006, Hyndman et al., 2008]. The polynomial benchmark is a low-dimensional regression baseline [Draper and Smith, 1998].

Guerrero's contribution is used here as a common smoothness scale for the methods whose fitted trends solve a finite-difference penalized least-squares problem. The train-validation selector and the time-weighted selector search over Guerrero smoothness values and choose the value that minimizes their validation loss. HPTrend and WhittakerTrend do not learn smoothness from the validation set in this paper; instead, their conventional penalty choices are converted back to the same Guerrero index so they can be interpreted on the same model-side scale. Thus the four Guerrero-compatible methods share the same penalty geometry, but they differ in how the penalty is selected.

The temporal protocol is strict. The series is split into 60 percent training, 20 percent validation, and 20 percent test observations. Hyperparameters are selected using only the training and validation segments. Final test forecasts are produced after refitting on the combined training and validation data.

3 Algorithm

The central algorithm fits a penalized trend for each candidate Guerrero smoothness setting, evaluates out-of-sample validation forecasts, and then refits the selected model before test evaluation. This section summarizes the mechanism behind the penalized estimator, the numerical linear algebra used to fit it, and the validation search used to choose among candidates. The penalized objective belongs to the same finite-difference smoothing family as Whittaker-Henderson graduation, Guerrero smoothing, and penalized spline methods [Whittaker, 1922, Henderson, 1924, Guerrero, 2007, Eilers and Marx, 1996, Eilers, 2003, Wahba, 1990].

Let $z = (z_1, \ldots, z_N)^\top$ denote the observed training series, let d be the finite-difference order, and let

$$K \in \mathbb{R}^{(N-d)\times N}$$

be the order-d differencing matrix. For a penalty value $\lambda > 0$, the penalized trend estimate is defined by

$$\hat{t}(\lambda) = \arg\min_{t\in\mathbb{R}^N} \left\{ \|z - t\|_2^2 + \lambda \|Kt\|_2^2 \right\}. \tag{1}$$

The first term measures fidelity to the observed series and the second term penalizes roughness. Larger values of λ produce smoother fitted trends.

The objective in (1) is quadratic. Define

$$A(\lambda) = I_N + \lambda K^\top K.$$

Taking the derivative with respect to t and setting it equal to zero gives

$$A(\lambda)\hat{t}(\lambda) = z.$$

Since $A(\lambda)$ is positive definite for $\lambda > 0$, the fitted trend is

$$\hat{t}(\lambda) = S(\lambda)z, \qquad S(\lambda) = \left(I_N + \lambda K^\top K\right)^{-1}. \tag{2}$$

Here $S(\lambda)$ is the smoothing matrix.

3.1 Spectral solution of the penalized system

Although (2) is written with an inverse, the implementation does not form a dense inverse directly. For a fixed sample size N and order d, it first constructs the penalty matrix

$$P_d = K^\top K.$$

This matrix is symmetric positive semidefinite, so it admits an orthogonal eigendecomposition

$$P_d = Q\Delta Q^\top, \qquad \Delta = \mathrm{diag}(\delta_1, \ldots, \delta_N).$$

Then

$$I_N + \lambda P_d = Q(I_N + \lambda\Delta)Q^\top,$$

and the fitted trend can be computed as

$$\hat{t}(\lambda) = Q\mathrm{diag}\left(\frac{1}{1+\lambda\delta_1}, \ldots, \frac{1}{1+\lambda\delta_N}\right)Q^\top z. \tag{3}$$

This reduces each candidate fit to projections onto the eigenvectors, elementwise division by $1+\lambda\delta_i$, and a projection back to the original time domain. Since the same Q and δ_i are reused for many candidate λ values at the same N and d, the paper pipeline benefits from caching the solver for repeated fits. The same eigenvalues also provide the effective-degrees-of-freedom trace used in Guerrero's index, so the numerical linear algebra for fitting and the numerical linear algebra for tuning are shared rather than duplicated.

For Guerrero-style penalized trends, the implementation may also estimate a constant d-th difference drift. In that case the right-hand side is adjusted iteratively and the drift estimate m is updated from the fitted trend until the change is smaller than a numerical tolerance. HPTrend and WhittakerTrend are run as classical fixed-drift baselines with this drift set to zero.

3.2 Mapping Guerrero smoothness to λ

The paper reports Guerrero smoothness for finite-difference penalized smoothers, but the linear system is solved in terms of λ. The eigendecomposition above gives a numerically convenient expression for the trace of the smoothing matrix:

$$\mathrm{tr}\{S(\lambda)\} = \sum_{i=1}^{N} \frac{1}{1+\lambda\delta_i}.$$

For $d \geq 1$, the implementation uses the normalized Guerrero index

$$s(\lambda; N, d) = \frac{1 - N^{-1}\sum_{i=1}^{N}(1+\lambda\delta_i)^{-1}}{1 - d/N}, \tag{4}$$

which maps the attainable range to a scale close to $[0, 1)$ for the finite sample and difference order [Guerrero, 2007]. Given a requested Guerrero smoothness value s, the code finds the corresponding λ by bisection:

1. Set an initial bracket $[0, 1]$.
2. Increase the upper endpoint by factors of ten until the target smoothness is bracketed.
3. Bisect the interval until the trace-based smoothness is within the numerical tolerance.

This step is monotone because increasing λ decreases the effective degrees of freedom $\mathrm{tr}\{S(\lambda)\}$ and therefore increases the Guerrero smoothness index.

3.3 How Guerrero's idea defines the penalized methods

Guerrero's idea is not a separate estimator in the pipeline; it is the trace-based scale used to parameterize and compare finite-difference penalized smoothers. Once N and d are fixed, every penalty value λ corresponds to a Guerrero smoothness value through the smoothing matrix trace. The implementation uses this correspondence in four related ways.

For TrainValidationSelector, the candidate set is written directly as pairs (d, s), where s is the Guerrero smoothness index. Each candidate s is converted to its corresponding λ, the penalized system is solved on the training segment, and the resulting trend is extrapolated into the validation segment. The selected model is the candidate with the smallest validation error. In this sense, Guerrero's index becomes the hyperparameter that the temporal validation split chooses.

For TimeWeightedValidationSelector, the same Guerrero-indexed candidate set is used. The difference is only in the validation criterion: later validation observations receive larger weights, so the selected Guerrero smoothness is the one that performs best for the more recent part of the validation window. The underlying finite-difference penalty, the s-to-λ conversion, and the linear solve are otherwise the same as in TrainValidationSelector.

For HPTrend, the trend is also a finite-difference penalized least-squares smoother, usually with a second-difference penalty. The classical HP presentation specifies a penalty λ rather than a Guerrero smoothness index. In the paper pipeline, that fixed λ is passed through the same trace formula to report the implied Guerrero smoothness. HPTrend is therefore Guerrero-compatible for interpretation, even though the validation set is not used to choose its penalty.

For WhittakerTrend, the fitted trend again solves the same algebraic family of linear systems, with a finite-difference penalty and a fixed penalty setting. As with HPTrend, the fixed λ determines a fitted smoother matrix and therefore an implied Guerrero smoothness index. This lets WhittakerTrend be reported beside the validation-selected methods without pretending that it ran a validation search.

This is why the paper reports Guerrero smoothness for TrainValidationSelector, TimeWeightedValidationSelector, HPTrend, and WhittakerTrend, but leaves that field blank for moving averages, exponential smoothing, and polynomial forecasting. Those latter methods can still be evaluated by forecast error and realized roughness, but they do not have a native $S_d(\lambda; N)$ because they are not defined by the penalized system in (1).

For validation, let R be the matrix that selects validation observations and let n_{val} be the number of validation points. The validation residual is

$$r(\lambda) = Rz - RS(\lambda)z,$$

and the validation mean squared error is

$$f(\lambda) = \frac{1}{n_{\text{val}}} r(\lambda)^\top r(\lambda). \tag{5}$$

The derivative calculation in the companion report follows from

$$S'(\lambda) = -S(\lambda) K^\top K S(\lambda),$$

which implies

$$r'(\lambda) = R S(\lambda) K^\top K S(\lambda) z.$$

Therefore,

$$f'(\lambda) = \frac{2}{n_{\text{val}}} \left[R\left(I_N - S(\lambda)\right) z\right]^\top \left[R S(\lambda) K^\top K S(\lambda) z\right]. \tag{6}$$

3.4 Validation search and local minima

The derivative in (6) explains the shape of the validation objective, but the implemented paper pipeline uses a robust direct search over candidate Guerrero smoothness values. For each candidate difference order d and smoothness value s, the algorithm:

1. converts s to λ using the bisection procedure above;
2. fits the penalized trend on the training segment only;
3. extrapolates the fitted trend into the validation segment;
4. computes the validation loss between the forecast and the observed validation log prices.

For the S&P 500 comparison, the paper script evaluates orders $d \in \{1, 2, 3\}$ and a fixed grid of Guerrero smoothness values between 0.05 and 0.98. The broader library default uses a denser grid, but the paper uses a smaller reproducible grid to keep the empirical study fast and stable.

The selector records the full validation curve and detects local minima on the one-dimensional smoothness grid for each order. A grid point is treated as a local minimum when its validation score is no larger than the scores of its neighbors; boundary points can also be local minima if they improve on their single neighbor. The library can optionally refine each detected bracket by golden-section search. In the paper pipeline, refinement is disabled so that the saved validation curves, local minima, and selected value all correspond exactly to the stated reproducible grid.

The time-weighted selector uses the same candidate search, but replaces the ordinary mean squared validation loss with a weighted validation loss. In the paper run, the weights follow an exponential decay scheme, so later validation observations receive relatively more emphasis than earlier validation observations. This changes only the scoring rule, not the penalized smoothing linear algebra. In forecasting terminology, this is a validation-rule change rather than a different smoothing model [Makridakis et al., 1998, Hyndman et al., 2008].

3.5 Forecasting from a fitted penalized trend

After fitting a penalized trend on the training segment, the package forecasts future trend values by extending the finite-difference structure at the right tail. If the selected order is d, the forecast recursively imposes a constant estimated d-th difference m. For $d = 1$, this is a constant slope extension; for $d = 2$, it is a constant second-difference extension; and for higher orders it extends the corresponding finite-difference recurrence. These forecasts are used for validation scoring and, after refitting on training plus validation, for final test evaluation. The benchmark forecasts are intentionally simpler: trailing moving averages, exponential smoothing, and polynomial extrapolation are used as transparent baselines rather than as a complete forecasting taxonomy [Brown, 1959, Holt, 2004, Winters, 1960, Gardner, 1985, 2006, Draper and Smith, 1998].

3.6 Final comparison protocol

The complete numerical procedure for selector-based penalized methods is:

1. split the series temporally into training, validation, and test segments;
2. fit and score each (d, s) candidate on training-to-validation forecasts;
3. choose the candidate with the lowest validation loss, without using the test segment;
4. refit the chosen penalized smoother on training plus validation;
5. forecast the test segment and compute MAE, RMSE, SMAPE, and related metrics;
6. compute realized roughness $R_d(\hat{t}) = \|D^d\hat{t}\|_2^2$ from the fitted training-plus-validation trend.

The non-Guerrero baselines use the same temporal protocol, but they do not have a Guerrero smoothness search. The moving-average baseline uses its fixed trailing window, the exponential-smoothing baseline uses its fixed α, and the polynomial forecaster uses its fixed degree. They still receive test forecasts and realized roughness diagnostics, but their Guerrero smoothness entries are left blank.

The Guerrero smoothness index used by the implementation is

$$S_d(\lambda; N) = 1 - \frac{\operatorname{tr}[S(\lambda)]}{N}, \qquad S(\lambda) = \left(I_N + \lambda K^\top K\right)^{-1}.$$

This index is meaningful for the finite-difference penalized smoothers in the comparison: the train-validation selector, the time-weighted selector, HPTrend, and WhittakerTrend. It is not a native parameter of moving averages, exponential smoothing, or polynomial forecasting.

In practice, solving $f'(\lambda) = 0$ does not give a simple general closed form. The implementation therefore uses numerical validation search over the package's normalized Guerrero smoothness parameter, converts the selected index to λ, and then refits the final penalized trend on the combined training and validation sample. The comparative pipeline applies the same temporal selection principle to the proposed train-validation selector and its time-weighted variant, while fixed-parameter baselines are fit on the same training and refit samples.

4 Data

The empirical data are daily S&P 500 observations from `yfinance` [Aroussi, 2024], using ticker `^GSPC` from 2010-01-01 onward. The analysis uses

$$y_t = \log(P_t),$$

where P_t is the adjusted close price when available and the close price otherwise. The raw and processed data are cached under `paper/data/` so that the experiment does not require a download on every run.

5 Metrics

Forecast accuracy is reported using MAE, MSE, RMSE, MAPE, and SMAPE. The main target is the observed log-price series. Real financial data do not provide an observed true trend, so this report does not evaluate true-trend recovery on the S&P 500 series. These forecast-error summaries are standard descriptive tools in empirical forecasting comparisons [Makridakis et al., 1998, Hyndman et al., 2008].

For finite-difference penalized least-squares smoothers, the reported Guerrero smoothness value is the model-side index

$$S_d(\lambda; N) = 1 - \frac{\operatorname{tr}\left[\left(I_N + \lambda D_d^\top D_d\right)^{-1}\right]}{N}, \qquad d \geq 1.$$

It is used to choose or interpret the penalty λ once the sample size N and difference order d are fixed [Guerrero, 2007].

As a separate output diagnostic, the paper reports realized roughness

$$R_d(\hat{\tau}) = \|D^d \hat{\tau}\|_2^2.$$

Realized roughness can be computed for any fitted trend and should not be identified with Guerrero smoothness.

6 Results

Figure 1 shows the temporal split used in the experiment. Figure 2 shows the fitted trends before the test period. Figure 3 shows validation curves for the selector-based methods. Figure 4 displays test forecasts, while Figures 5a and 5b summarize test errors. The realized roughness diagnostic is shown in Figure 6.

Figure 1 documents the chronological evaluation design. The training segment is used to fit candidate models, the validation segment is used only for hyperparameter selection, and the test segment is held out until final evaluation. This ordering is important because randomly mixing financial time points would leak future information into the model-selection step.

Figure 2 compares the fitted trends on the train-plus-validation sample before test forecasting begins. Curves that stay close to the observed series have lower in-sample smoothing bias but may carry more short-run variation, while flatter curves impose stronger smoothness. The figure is therefore a visual complement to the parameter and realized-roughness tables: it shows what each smoothing choice does to the observed log-price path.

Figure 3 shows how validation error changes as the Guerrero smoothness index varies for each candidate difference order. The selected point is the minimum of the displayed validation criterion for each selector. In this run, the order-three curves are much larger than the order-one and order-two curves, so the plotted y-axis is limited to reveal the smaller-scale structure that determines the useful comparison among the better-performing candidates.

Figure 4 overlays each method's test-period forecast on the held-out observed log prices. This figure should be read together with the test-error table: a forecast that visually tracks the level and slope of the test period tends to have lower RMSE and SMAPE, while forecasts that remain too flat or extrapolate the wrong slope accumulate larger errors.

Figures 5a and 5b rank methods by two test-set error summaries. RMSE emphasizes larger forecast mistakes because errors are squared before averaging, while SMAPE reports a scale-free

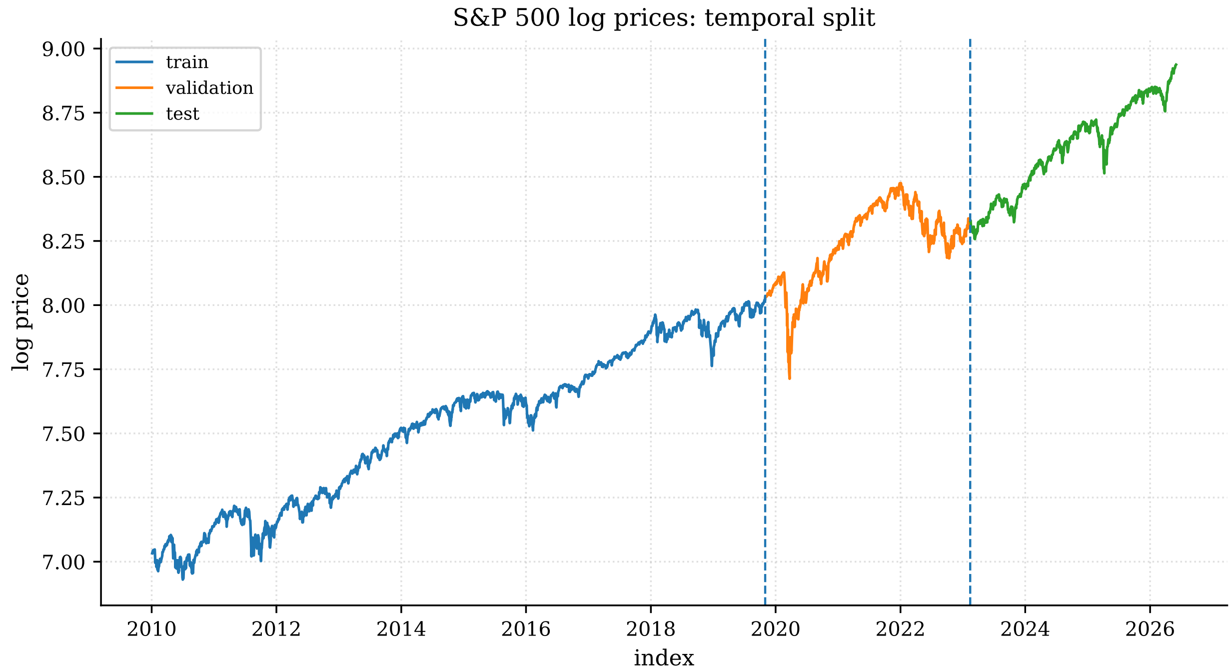


Figure 1: Temporal train, validation, and test split for S&P 500 log prices.

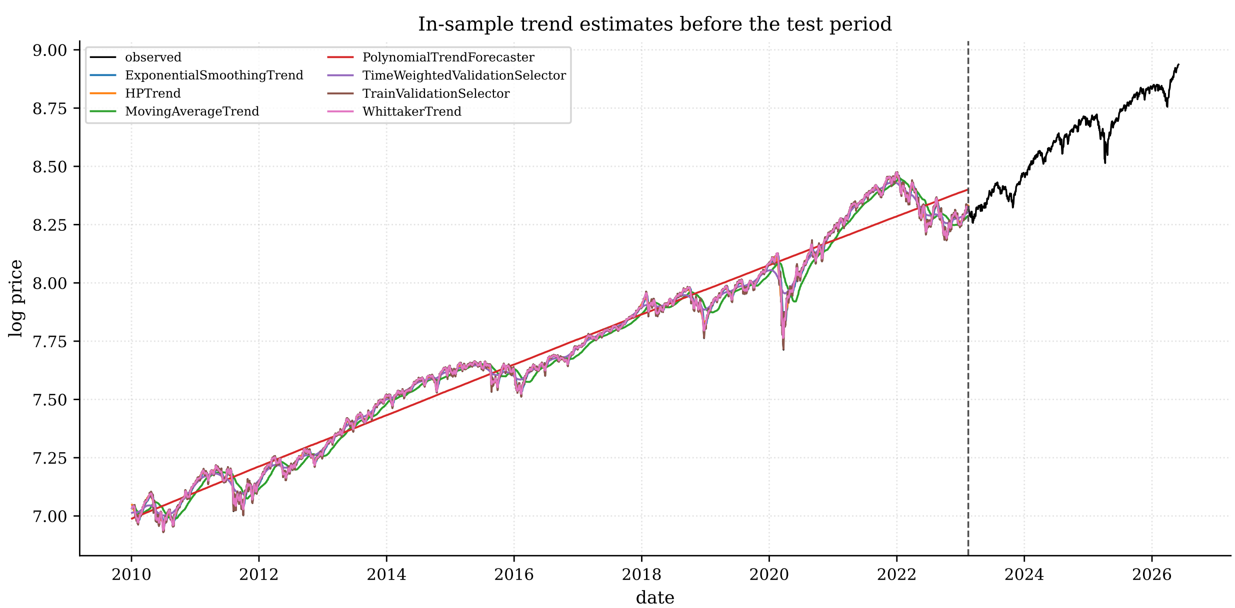


Figure 2: In-sample trend estimates fitted before the test period.

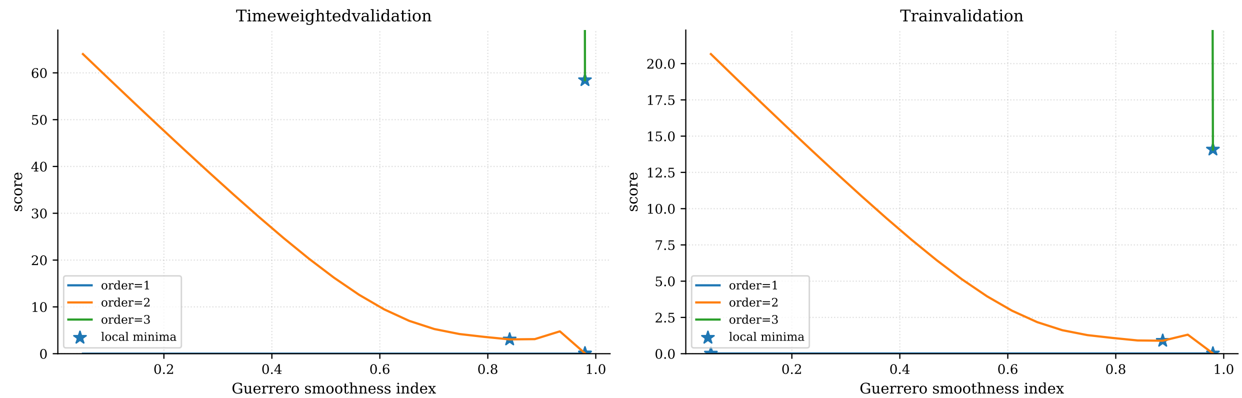


Figure 3: Validation curves over the Guerrero smoothness index for selector-based penalized methods. The vertical scale is clipped to keep the lower-order curves readable when the order-three curve is much larger.

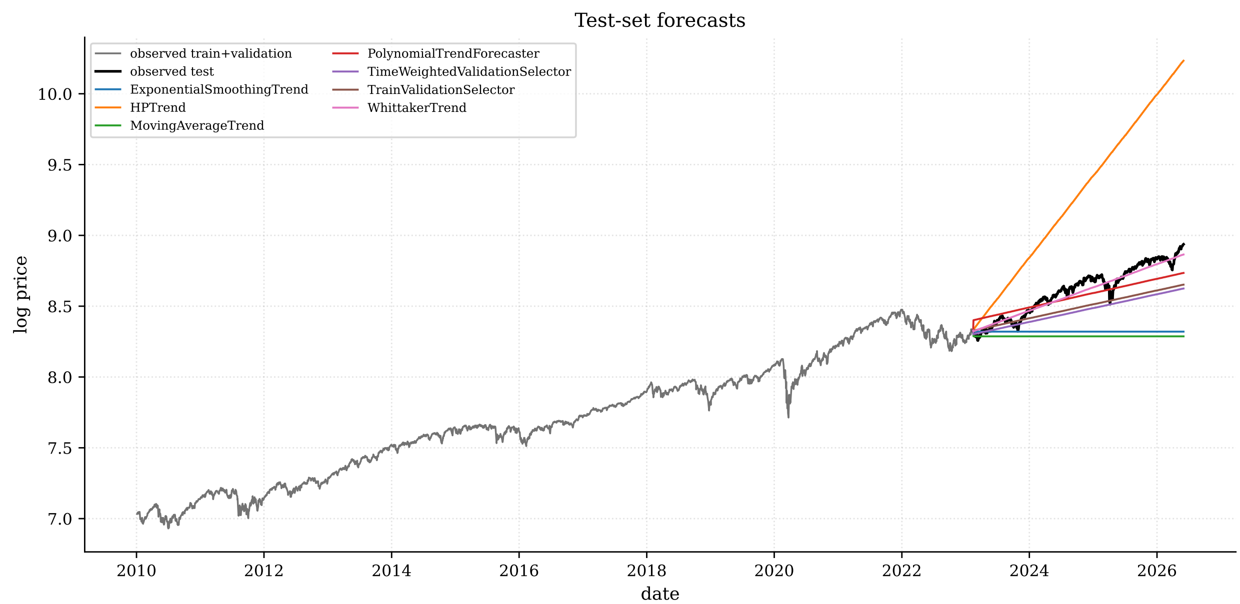
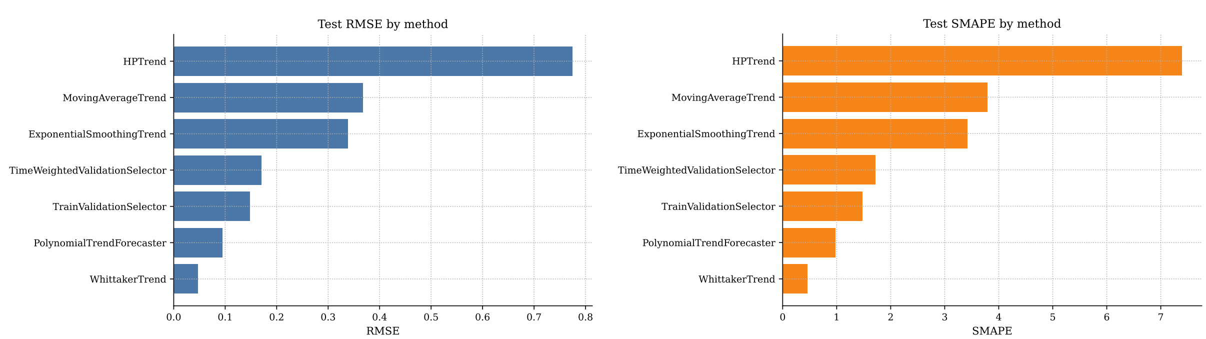


Figure 4: Test-set forecasts from the compared methods.



(a) RMSE

(b) SMAPE

Figure 5: Test error summaries.

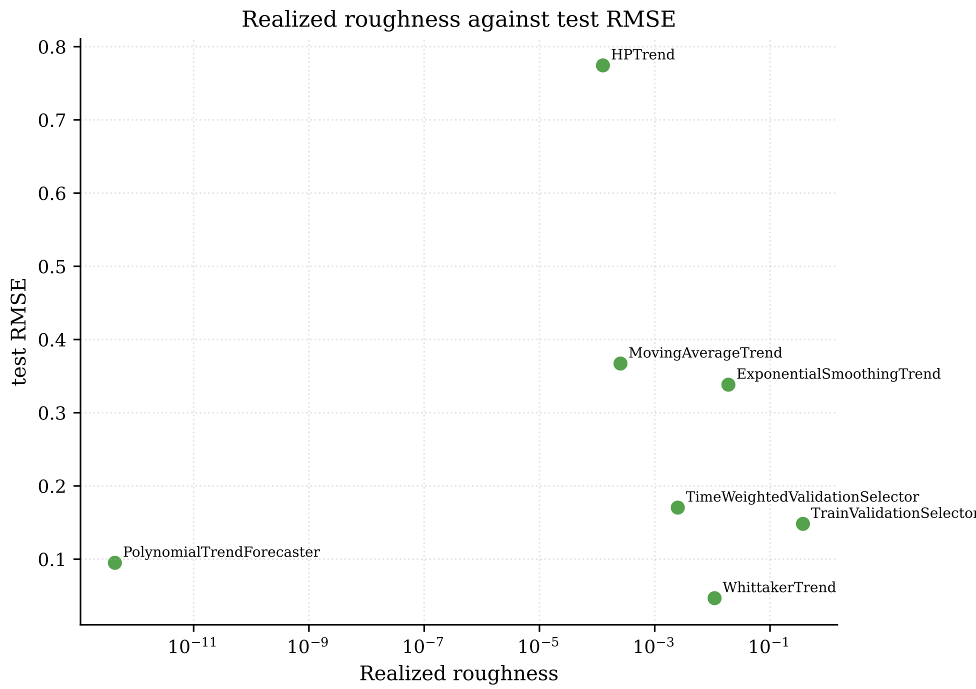


Figure 6: Realized roughness diagnostic against test RMSE.

percentage-style error. A method that performs well in both panels is more robustly accurate on the chosen test window than a method that improves only one metric.

Figure 6 separates forecast performance from realized roughness. The horizontal axis is the diagnostic $R_d(\hat{\tau}) = \|D^d\hat{\tau}\|_2^2$, computed after fitting each trend; it is not Guerrero smoothness. Points far to the left are smoother in realized finite-difference terms, while lower points have better test RMSE. The scatter therefore shows whether extra smoothness coincides with better forecasting in this empirical split.

6.1 Tables

Table 1 reports the final held-out test results. The columns MAE, RMSE, and SMAPE summarize forecast accuracy on observations that were not used for hyperparameter selection. The parameter columns should be read method by method: Guerrero smoothness and λ are meaningful only for finite-difference penalized smoothers, while window, alpha, and degree describe the moving-average, exponential-smoothing, and polynomial baselines.

Table 1: Held-out test accuracy and method-specific settings.

model	n_obs	MAE	RMSE	SMAPE	best_order	guerrero_smoothness	best_lambda	window	alpha	degree
TrainValidationSelector	827	0.127263	0.148216	1.475928	1.0	0.05	0.027008			
TimeWeightedValidationSelector	827	0.147868	0.170415	1.717913	1.0	0.98	615.756247			
HPTrend	827	0.675167	0.774414	7.389961	2.0	0.944195	1600.0			
WhittakerTrend	827	0.039826	0.04672	0.462446	2.0	0.75	4.874751			
MovingAverageTrend	827	0.322219	0.367195	3.792746				60.0		
ExponentialSmoothingTrend	827	0.290947	0.338309	3.416307					0.2	
PolynomialTrendForecaster	827	0.083899	0.09498	0.973113						2.0

Table 2 reports validation-period performance before the final test refit. These values explain why the validation-based selectors chose their reported settings, but they are not final generalization errors. Comparing this table with Table 1 shows whether a method that looked good during validation continued to perform well on the held-out test window.

Table 2: Validation-period accuracy used for model-selection comparison.

model	n_obs	MAE	RMSE	SMAPE	best_order	guerrero_smoothness	best_lambda	window	alpha	degree
TrainValidationSelector	825	0.096959	0.118151	1.176689	1.0	0.05	0.027008			
TimeWeightedValidationSelector	825	0.103816	0.126905	1.260935	1.0	0.98	615.756247			
HPTrend	825	0.627408	0.742694	7.196558	2.0	0.944195	1600.0			
WhittakerTrend	825	0.939478	1.099213	10.529626	2.0	0.75	4.874751			
MovingAverageTrend	825	0.25886	0.290361	3.17557				60.0		
ExponentialSmoothingTrend	825	0.237732	0.269659	2.911698					0.2	
PolynomialTrendForecaster	825	0.133972	0.161477	1.629947						2.0

Table 3 collects the selected or fixed model settings in one place. The selector rows show the chosen difference order, Guerrero smoothness index, and corresponding λ. HPTrend and WhittakerTrend show their fixed finite-difference penalty settings and implied Guerrero indices. The non-Guerrero benchmarks deliberately have blank Guerrero smoothness entries, because their tuning parameters are window length, exponential alpha, or polynomial degree rather than $S_d(\lambda; N)$.

Table 3: Selected or fixed model parameters.

model	kind	best_order	guerrero_smoothness	best_lambda	degree	window	alpha
TrainValidationSelector	selector	1.0	0.05	0.027008			
TimeWeightedValidationSelector	selector	1.0	0.98	615.756247			
HPTrend	estimator	2.0	0.944195	1600.0			
WhittakerTrend	estimator	2.0	0.75	4.874751			
MovingAverageTrend	estimator					60.0	
ExponentialSmoothingTrend	estimator						0.2
PolynomialTrendForecaster	polynomial_forecaster				2.0		

Table 4 reports realized roughness for every fitted trend using the stated roughness order. Unlike Guerrero smoothness, this diagnostic is computed from the fitted trend itself and can therefore be reported for all methods. The accompanying test RMSE column makes it possible to compare the smoothness of the fitted curve with its forecasting accuracy.

7 Discussion

The results should be interpreted as one empirical comparison on one financial index over one historical period. Forecasting accuracy and trend recovery are different goals, and a method that forecasts well over the selected test window need not recover a latent economic trend. Validation-based selection can also overfit the validation period, especially when the candidate grid is wide or the market regime changes between validation and test.

Guerrero smoothness is a model-side index defined only for finite-difference penalized least-squares smoothers. It maps a smoothing penalty λ into a comparable scale $S_d(\lambda; N)$ once the sample size N and difference order d are fixed. By contrast, realized roughness is an output diagnostic $R_d(\hat{\tau}) = \|D^d \hat{\tau}\|_2^2$ that can be computed for any fitted trend. Therefore, moving-average,

Table 4: Realized roughness diagnostics and test RMSE.

model	roughness_order	realized_roughness	test_RMSE
TrainValidationSelector	1	0.372955	0.148216
TimeWeightedValidationSelector	1	0.002511	0.170415
HPTrend	2	0.000126	0.774414
WhittakerTrend	2	0.010925	0.04672
MovingAverageTrend	2	0.000254	0.367195
ExponentialSmoothingTrend	2	0.018961	0.338309
PolynomialTrendForecaster	2	0.0	0.09498

exponential-smoothing, and polynomial benchmarks do not have a native Guerrero smoothness index, although their fitted trends can still be compared through realized roughness.

Future sensitivity checks should vary the time period, include additional tickers, and compare against synthetic experiments where the true trend is known. Additional baselines such as Kalman filters, structural time-series models, ARIMA models, STL/LOESS, smoothing splines, and structural-break models would also make the comparison broader [Kalman, 1960, Harvey, 1989, Box et al., 2008, Cleveland, 1979, Cleveland et al., 1990, Wahba, 1990].

8 Reproducibility

The empirical pipeline is distributed with the local `trend_estimation` package and paper scripts [Espino-Montelongo, 2026]. From the repository root, activate the environment and install the local package:

```
conda activate trend_estimation
python -m pip install -e ".[dev,finance]"
```

Then run:

```
python paper/build_paper.py --ticker ^GSPC --start 2010-01-01 --no-compile
```

If `latexmk` is installed, compile with:

```
latexmk -pdf -interaction=nonstopmode -outdir=paper/build paper/main.tex
```

9 Conclusion

This paper establishes a reproducible comparison between train-validation penalized trend selection, a time-weighted variant, and several benchmark methods on S&P 500 log prices. The current evidence is descriptive and case-specific. Broader empirical and synthetic experiments are needed before making stronger methodological claims.

References

Ran Aroussi. yfinance: Download market data from yahoo! finance's api. `https://github.com/ranaroussi/yfinance`, 2024. Python package.

George E. P. Box, Gwilym M. Jenkins, and Gregory C. Reinsel. *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Hoboken, NJ, 4 edition, 2008. doi: 10.1002/9781118619193. URL `https://doi.org/10.1002/9781118619193`.

Peter J. Brockwell and Richard A. Davis. *Introduction to Time Series and Forecasting*. Springer, Cham, 3 edition, 2016. doi: 10.1007/978-3-319-29854-2. URL `https://doi.org/10.1007/978-3-319-29854-2`.

Robert Goodell Brown. *Statistical Forecasting for Inventory Control*. McGraw-Hill, New York, 1959.

Chris Chatfield. *The Analysis of Time Series: An Introduction*. Chapman & Hall/CRC, Boca Raton, 6 edition, 2003. doi: 10.4324/9780203491683. URL `https://doi.org/10.4324/9780203491683`.

Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. STL: A seasonal-trend decomposition procedure based on LOESS. *Journal of Official Statistics*, 6(1):3–73, 1990.

William S. Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74(368):829–836, 1979. doi: 10.1080/01621459.1979.10481038. URL `https://doi.org/10.1080/01621459.1979.10481038`.

Norman R. Draper and Harry Smith. *Applied Regression Analysis*. John Wiley & Sons, New York, 3 edition, 1998. doi: 10.1002/9781118625590. URL `https://doi.org/10.1002/9781118625590`.

Paul H. C. Eilers. A perfect smoother. *Analytical Chemistry*, 75(14):3631–3636, 2003. doi: 10.1021/ac034173t. URL `https://doi.org/10.1021/ac034173t`.

Paul H. C. Eilers and Brian D. Marx. Flexible smoothing with B-splines and penalties. *Statistical Science*, 11(2):89–121, 1996. doi: 10.1214/ss/1038425655. URL `https://doi.org/10.1214/ss/1038425655`.

Heriberto Espino-Montelongo. trend_estimation: Penalized trend estimation with validation-based smoothness selection. `https://github.com/heritaco/Trend-Estimation`, 2026. Python package and reproducible paper pipeline.

Everette S. Gardner. Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1): 1–28, 1985. doi: 10.1002/for.3980040103. URL `https://doi.org/10.1002/for.3980040103`.

Everette S. Gardner. Exponential smoothing: The state of the art—part II. *International Journal of Forecasting*, 22(4):637–666, 2006. doi: 10.1016/j.ijforecast.2006.03.005. URL `https://doi.org/10.1016/j.ijforecast.2006.03.005`.

Victor M. Guerrero. Time series smoothing by penalized least squares. *Statistics & Probability Letters*, 77(12):1225–1234, 2007. doi: 10.1016/j.spl.2007.03.006. URL `https://doi.org/10.1016/j.spl.2007.03.006`.

James D. Hamilton. *Time Series Analysis*. Princeton University Press, Princeton, 1994. doi: 10.1515/9780691218632. URL https://doi.org/10.1515/9780691218632.

Andrew C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, Cambridge, 1989. doi: 10.1017/CBO9781107049994. URL https://doi.org/10.1017/CBO9781107049994.

Robert Henderson. A new method of graduation. *Transactions of the Actuarial Society of America*, 25:29–40, 1924.

Robert J. Hodrick and Edward C. Prescott. Postwar U.S. business cycles: An empirical investigation. *Journal of Money, Credit and Banking*, 29(1):1–16, 1997. doi: 10.2307/2953682. URL https://doi.org/10.2307/2953682.

Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10, 2004. doi: 10.1016/j.ijforecast.2003.09.015. URL https://doi.org/10.1016/j.ijforecast.2003.09.015. Reprint of the 1957 Office of Naval Research memorandum.

Rob J. Hyndman, Anne B. Koehler, J. Keith Ord, and Ralph D. Snyder. *Forecasting with Exponential Smoothing: The State Space Approach*. Springer Series in Statistics. Springer, Berlin, Heidelberg, 2008. doi: 10.1007/978-3-540-71918-2. URL https://doi.org/10.1007/978-3-540-71918-2.

Rudolf E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. doi: 10.1115/1.3662552. URL https://doi.org/10.1115/1.3662552.

Robert G. King and Sergio T. Rebelo. Low frequency filtering and real business cycles. *Journal of Economic Dynamics and Control*, 17(1–2):207–231, 1993. doi: 10.1016/S0165-1889(06)80010-2. URL https://doi.org/10.1016/S0165-1889(06)80010-2.

Spyros Makridakis, Steven C. Wheelwright, and Rob J. Hyndman. *Forecasting: Methods and Applications*. John Wiley & Sons, New York, 3 edition, 1998.

Grace Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics, Philadelphia, 1990. doi: 10.1137/1.9781611970128. URL https://doi.org/10.1137/1.9781611970128.

Edmund T. Whittaker. On a new method of graduation. *Proceedings of the Edinburgh Mathematical Society*, 41:63–75, 1922. doi: 10.1017/S0013091500077853. URL https://doi.org/10.1017/S0013091500077853.

Peter R. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342, 1960. doi: 10.1287/mnsc.6.3.324. URL https://doi.org/10.1287/mnsc.6.3.324.