Extremum Constrained Anchor Path Regressor for Multi-Step Time Series Forecasting

Sofien Kaabar, CFA
Quant Atlas
www.quant-atlas.com

June 2026

Abstract

This paper introduces the Extremum Constrained Anchor Path Regressor, abbreviated as ECAPR, as a descriptive framework for multi-step time series forecasting. The method is designed to produce a future trajectory rather than a single endpoint forecast. ECAPR combines feature-based ensemble regression, nearest-neighbor analog retrieval, structural path estimation, and texture-preserving path deformation.

The central idea is that a useful forecast should describe both the expected destination of the series and the plausible shape of the path followed to reach it. To do this, ECAPR predicts structural properties of the future path, including terminal return, realized volatility, roughness, and the approximate timing of future highs and lows. It then selects a historical analog path and deforms it so that its texture is preserved while its drift and volatility are adjusted to match the predicted structural profile.

This paper is descriptive in nature. Its purpose is to present the intuition, mathematics, and implementation logic of ECAPR rather than to claim empirical superiority. For this reason, the paper does not provide a full back-test across time series. The objective is to offer a basic framework upon which more sophisticated forecasting algorithms can be built.

1 Introduction

Time series forecasting is often presented as the problem of predicting the next value of a sequence. In many applications, however, a single endpoint forecast is not sufficient. The intermediate path can matter as much as the final value. This is especially true when the path itself carries information about risk, instability, drawdown, local extrema, or the timing of a move.

For example, two forecasts may share the same terminal value but describe very different futures. One path may rise smoothly, while another may first decline sharply before recovering. These two paths are not structurally equivalent, even if they end at the same point.

ECAPR is designed around this distinction. Instead of forecasting only the value at the end of the horizon, the model forecasts the full future path:

$$\widehat{P}_{t+1:t+h} = \left(\widehat{P}_{t+1}, \widehat{P}_{t+2}, \ldots, \widehat{P}_{t+h}\right),$$

where h is the forecast horizon.

The model is applied to time series data. Although the implementation is naturally suited to financial series, the framework is general and can be adapted to any ordered sequence where the structure of the future path matters.

2 Time Series Setting

Let

$$\{P_t\}_{t=1}^T$$

be a univariate time series, where P_t is the observed value at time t. At a given forecast origin t, the goal is to forecast the next h observations:

$$Y_t = (P_{t+1}, P_{t+2}, \ldots, P_{t+h}).$$

Instead of modeling the future path directly in price space, ECAPR works mainly with log returns from the forecast origin:

$$R_{t,j} = \log\left(\frac{P_{t+j}}{P_t}\right), \qquad j = 1, 2, \ldots, h.$$

The future cumulative return path is therefore:

$$R_t = (R_{t,1}, R_{t,2}, \ldots, R_{t,h}).$$

This representation makes the model focus on relative movement rather than the absolute level of the series.

3 Core Intuition

ECAPR is built on a simple idea: a forecast path should be both structurally informed and historically plausible.

The structural part comes from supervised learning. The model uses the current state of the time series to estimate future properties such as terminal return, future volatility, roughness, and the approximate location of the future high and low.

The plausibility part comes from historical analogs. Instead of creating a completely artificial curve, ECAPR retrieves a historical path whose future behavior resembles the predicted structural profile. This path provides the texture of the forecast.

The final forecast is created by deforming the selected historical path. The deformation process adjusts the selected path so that it matches the predicted terminal return and volatility while preserving the relative shape of the original analog.

The framework therefore has three main layers:

1. estimate the future structure of the path;
2. retrieve a plausible historical path texture;
3. deform that texture to match the predicted structure.

4 Feature Representation

At each forecast origin t, the model computes a feature vector:

$$X_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,m}),$$

where m is the number of engineered features.

The feature vector summarizes recent behavior in the time series. In a financial implementation, the feature set may include:

- lagged log returns;
- rolling volatility over several windows;
- volatility ratios;
- momentum over several windows;
- local trend slopes;
- distance from recent highs and lows;
- RSI-style momentum measures;
- ATR-style range features when high and low observations are available;
- return autocorrelation;
- skewness and kurtosis;
- recent drawdown and runup;
- path geometry measures such as sign changes and efficiency ratio;
- volatility compression;
- optional volume features when volume data is available.

The essential rule is that X_t must only use information available at or before time t:

$$X_t = f(P_1, P_2, \ldots, P_t).$$

This prevents look-ahead bias in the feature construction process.

5 Future Path Targets

For each historical origin i, the future path over horizon h is:

$$Y_i = (P_{i+1}, P_{i+2}, \ldots, P_{i+h}).$$

The corresponding cumulative log-return path is:

$$R_i = \left(\log\left(\frac{P_{i+1}}{P_i}\right), \log\left(\frac{P_{i+2}}{P_i}\right), \ldots, \log\left(\frac{P_{i+h}}{P_i}\right)\right).$$

ECAPR also computes structural targets from each future path.
The terminal return is:

$$r_i^{\text{term}} = R_{i,h}.$$

The step-by-step log return is:

$$s_{i,j} = \log\left(\frac{P_{i+j}}{P_{i+j-1}}\right), \qquad j = 1, 2, \ldots, h.$$

The future realized volatility is:

$$\sigma_i^{\text{future}} = \sqrt{\frac{1}{h-1}\sum_{j=1}^{h}(s_{i,j}-\bar{s}_i)^2},$$

where $\bar{s}_i$ is the average of the future step returns.

The high step is:

$$\tau_i^{\text{high}} = \arg\max_{1\le j\le h} R_{i,j}.$$

The low step is:

$$\tau_i^{\text{low}} = \arg\min_{1\le j\le h} R_{i,j}.$$

The roughness of the future path is defined as:

$$\kappa_i = 1 - \frac{\left|\sum_{j=1}^{h} s_{i,j}\right|}{\sum_{j=1}^{h}|s_{i,j}| + \epsilon},$$

where ϵ is a small positive constant used to avoid division by zero.

This roughness measure compares the net movement of the path with the total absolute movement. A smooth directional path has lower roughness. A jagged path with frequent reversals has higher roughness.

6 Model Architecture

ECAPR uses two supervised learning components and one nearest-neighbor component.

6.1 Path Model

The path model estimates the full future cumulative return path:

$$\widehat{R}_t = g_{\text{path}}(X_t),$$

where

$$\widehat{R}_t = \left(\widehat{R}_{t,1}, \widehat{R}_{t,2}, \ldots, \widehat{R}_{t,h}\right).$$

This is a multi-output regression problem. The model maps the current feature vector to an estimated future return trajectory.

6.2 Structural Model

The structural model estimates key properties of the future path:

$$\widehat{S}_t = g_{\text{struct}}(X_t),$$

where

$$\widehat{S}_t = \left(\widehat{r}_t^{\text{term}}, \widehat{\sigma}_t^{\text{future}}, \widehat{\kappa}_t, \widehat{\tau}_t^{\text{high}}, \widehat{\tau}_t^{\text{low}}\right).$$

These values describe the expected endpoint, volatility, roughness, and approximate location of the future maximum and minimum.

6.3 Nearest-Neighbor Retrieval

The nearest-neighbor component retrieves historical observations with feature profiles similar to the current forecast origin. This creates a pool of candidate future paths that actually occurred in the past.

The training set available at forecast origin t is:

$$\mathcal{D}_t = \{(X_i, R_i, S_i) : i + h \leq t\}.$$

The condition $i + h \leq t$ ensures that every future target in the training set is fully known before the current forecast origin.

7 Extremum Constrained Analog Selection

After retrieving nearby historical observations, ECAPR selects the analog path whose structure is closest to the predicted structure.

For each candidate neighbor i, define the structural distance:

$$D_i = w_1 \left| r_i^{\text{term}} - \widehat{r}_t^{\text{term}} \right| + w_2 \left| \sigma_i^{\text{future}} - \widehat{\sigma}_t^{\text{future}} \right| + w_3 \left| \kappa_i - \widehat{\kappa}_t \right| + w_4 \left| \tau_i^{\text{high}} - \widehat{\tau}_t^{\text{high}} \right| + w_5 \left| \tau_i^{\text{low}} - \widehat{\tau}_t^{\text{low}} \right|.$$

The selected analog is:

$$i^* = \arg \min_{i \in \mathcal{N}(t)} D_i,$$

where $\mathcal{N}(t)$ is the set of nearest neighbors of the current feature vector.

The selected historical step-return path is:

$$q_t = (s_{i^*,1}, s_{i^*,2}, \ldots, s_{i^*,h}).$$

This path provides the historical texture of the forecast.

8 Path Deformation

The selected analog path is historical, but it may not match the predicted terminal return or volatility. ECAPR therefore deforms the path.

Let

$$q_t = (q_{t,1}, q_{t,2}, \ldots, q_{t,h})$$

be the selected step-return path.

First, rescale the path to match the predicted future volatility:

$$q'_{t,j} = q_{t,j} \cdot \frac{\widehat{\sigma}_t^{\text{future}}}{\sigma(q_t)}, \qquad j = 1, 2, \ldots, h.$$

Second, compute the drift correction required to match the predicted terminal return:

$$c_t = \frac{\widehat{r}_t^{\text{term}} - \sum_{j=1}^{h} q'_{t,j}}{h}.$$

Then apply the correction to each step:

$$\widetilde{q}_{t,j} = q'_{t,j} + c_t.$$

The final forecast path is reconstructed from the last observed value:

$$\widehat{P}_{t+j} = P_t \exp\left(\sum_{k=1}^{j} \widetilde{q}_{t,k}\right), \qquad j = 1, 2, \ldots, h.$$

This produces a path that preserves the relative texture of the selected analog while matching the predicted drift and volatility profile.

9 Implementation Logic

The implementation follows a walk-forward structure. At each forecast origin t, the following steps are performed:

1. Build the training set using only historical samples whose future paths are already known.
2. Compute feature vectors for all valid historical origins.
3. Compute future path targets and structural targets.
4. Standardize the feature matrix using the training set only.
5. Fit the path model.
6. Fit the structural model.
7. Fit a nearest-neighbor model on the standardized training features.
8. Compute and standardize the current feature vector.
9. Predict the future structural profile.
10. Retrieve candidate analog paths.
11. Select the analog path whose structure is closest to the predicted structure.
12. Rescale and shift the selected path.
13. Reconstruct the final forecast path.

A simplified implementation outline is:

```
For each forecast origin t:

    Build training samples i such that i + h <= t

    Compute X_train
    Compute future path targets
    Compute structural targets

    Fit scaler on X_train
    Fit path model
    Fit structural model
```

```
Fit nearest-neighbor model

Compute current feature vector X_t
Predict structural future profile

Retrieve nearest historical feature vectors
Select the best analog path using structural distance

Rescale analog path to predicted volatility
Shift analog path to predicted terminal return

Reconstruct final forecast path from the last observed value
```

10 Illustrative Prediction Section

This section is reserved for visual examples of ECAPR forecasts. The goal of these figures is to show how the predicted path behaves relative to the realized path after the forecast origin.

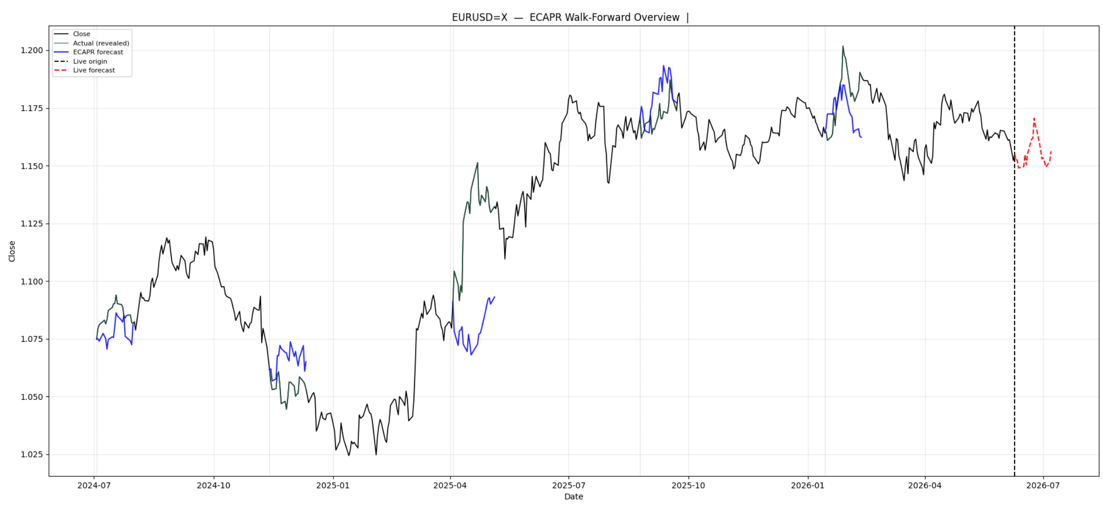

Figure 1: Example ECAPR forecast path. The historical time series is shown before the forecast origin, while the predicted path and realized path are shown over the forecast horizon.

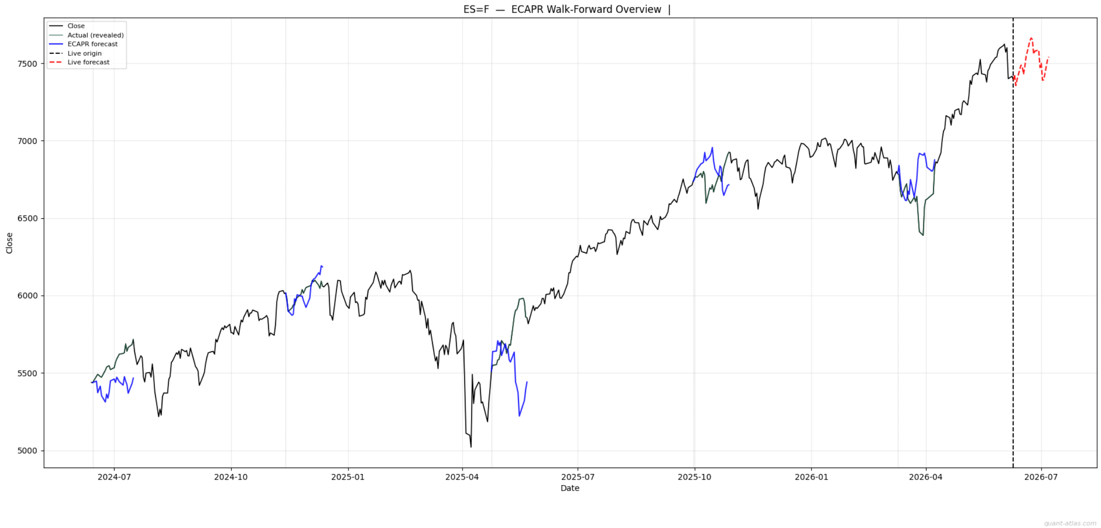

Figure 2: Example ECAPR forecast path. The historical time series is shown before the forecast origin, while the predicted path and realized path are shown over the forecast horizon.

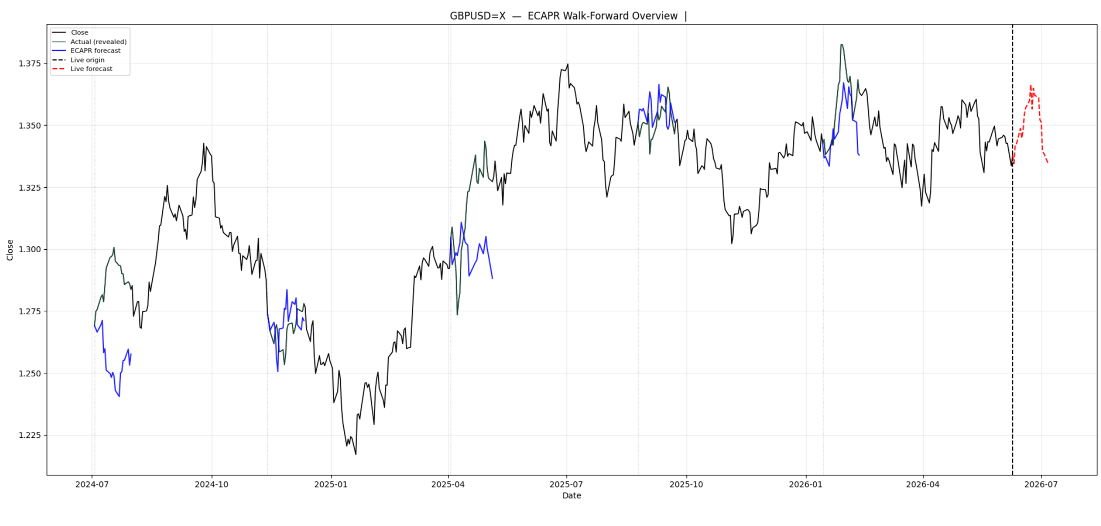

Figure 3: Example ECAPR forecast path. The historical time series is shown before the forecast origin, while the predicted path and realized path are shown over the forecast horizon.

These examples are illustrative only. They are not intended to represent a statistical validation of the algorithm.

11 Pathwise Correlation for Multi-Step Forecast Evaluation

Because ECAPR produces a multi-step forecast, it should be evaluated using metrics that can compare entire paths. A suitable metric is pathwise correlation.

At forecast origin t, define the predicted cumulative return path as:

$$\widehat{R}_t = \left(\widehat{R}_{t,1}, \widehat{R}_{t,2}, \ldots, \widehat{R}_{t,h}\right),$$

and the realized cumulative return path as:

$$R_t = (R_{t,1}, R_{t,2}, \ldots, R_{t,h}).$$

The pathwise Pearson correlation is:

$$\rho_t = \frac{\sum_{j=1}^{h} \left(\widehat{R}_{t,j} - \overline{\widehat{R}}_t\right)\left(R_{t,j} - \overline{R}_t\right)}{\sqrt{\sum_{j=1}^{h} \left(\widehat{R}_{t,j} - \overline{\widehat{R}}_t\right)^2} \sqrt{\sum_{j=1}^{h} \left(R_{t,j} - \overline{R}_t\right)^2}}.$$

Here,

$$\overline{\widehat{R}}_t = \frac{1}{h} \sum_{j=1}^{h} \widehat{R}_{t,j}$$

and

$$\overline{R}_t = \frac{1}{h} \sum_{j=1}^{h} R_{t,j}.$$

This metric measures whether the predicted path and the realized path move together over the forecast horizon. It is more suitable for multi-step forecasts than a single endpoint directional hit because it evaluates the shape of the forecast trajectory.

Across N forecast origins, a simple average pathwise correlation is:

$$\overline{\rho} = \frac{1}{N} \sum_{t=1}^{N} \rho_t.$$

A Fisher-transformed average may also be used:

$$z_t = \frac{1}{2} \log\left(\frac{1 + \rho_t}{1 - \rho_t}\right),$$

$$\overline{\rho}_F = \tanh\left(\frac{1}{N} \sum_{t=1}^{N} z_t\right).$$

Forecasts generated by ECAPR can therefore be evaluated using pathwise correlation. A positive value means the predicted and realized paths tend to move together. A value near zero suggests little path similarity. A negative value suggests that the predicted path often moves in the opposite shape to the realized path.

Pathwise correlation should not be treated as a complete performance measure. It evaluates shape similarity, but it does not measure trading value, economic usefulness, transaction costs, or risk-adjusted performance.

12 Extrema Precision Index

Since ECAPR explicitly models the approximate timing of future highs and lows, another useful descriptive metric is the Extrema Precision Index.

Define the predicted high step as:

$$\widehat{\tau}_t^{\text{high}} = \arg \max_{1 \le j \le h} \widehat{P}_{t+j},$$

and the predicted low step as:

$$\widehat{\tau}_t^{\text{low}} = \arg \min_{1 \le j \le h} \widehat{P}_{t+j}.$$

The realized high and low steps are:

$$\tau_t^{\text{high}} = \arg \max_{1 \le j \le h} P_{t+j},$$

$$\tau_t^{\text{low}} = \arg \min_{1 \le j \le h} P_{t+j}.$$

Given a tolerance margin m, the high-step hit is:

$$I_t^{\text{high}} = \mathbb{1} \left(\left| \widehat{\tau}_t^{\text{high}} - \tau_t^{\text{high}} \right| \le m \right),$$

and the low-step hit is:

$$I_t^{\text{low}} = \mathbb{1} \left(\left| \widehat{\tau}_t^{\text{low}} - \tau_t^{\text{low}} \right| \le m \right).$$

The Extrema Precision Index is:

$$EPI_t = \frac{1}{2} \left(I_t^{\text{high}} + I_t^{\text{low}} \right).$$

This metric is useful when the forecast is expected to describe not only the direction of the path, but also the approximate timing of important local extrema.

13 Limitations

ECAPR has several limitations. First, the model depends on the quality of the engineered features. If the feature vector fails to describe the relevant state of the time series, the forecast will be weak. Second, the model relies on historical analog paths. If the current environment differs materially from historical conditions, the selected analog may be misleading. Third, deformation can preserve historical texture while still producing an incorrect forecast. Matching terminal return and volatility does not guarantee that the future sequence will unfold in the same way. Fourth, pathwise correlation and EPI are descriptive evaluation tools. They do not prove that the model has economic value or practical decision value. Finally, ECAPR should be understood as a forecasting framework rather than a complete trading system. Turning it into a trading algorithm would require additional layers such as signal conversion, position sizing, transaction cost modeling, robustness analysis, and risk management.

14 Descriptive Scope

This paper is descriptive in nature rather than empirical research-oriented. Its objective is to explain ECAPR from the basics to the mathematical and implementation logic. For that reason, the paper does not include a complete back-test across time series. The absence of a back-test is intentional. The goal is to present the structure of the algorithm, clarify its mechanics, and define evaluation tools suited to multi-step path forecasting. Future work could test ECAPR across several datasets, horizons, frequencies, regimes, and benchmark models. It could also compare the framework against random walk forecasts, autoregressive models, simple nearest-neighbor analog models, tree-based models, recurrent networks, and transformer-based time series models.

15 Conclusion

This paper introduced the Extremum Constrained Anchor Path Regressor as a framework for multi-step time series forecasting. The model is built around the idea that a useful path forecast should describe both destination and trajectory. ECAPR combines feature-based learning, structural target prediction, nearest-neighbor analog retrieval, extremum-aware path selection, and volatility-adjusted path deformation. The framework is not presented as a complete trading system or as an empirically validated predictive engine. It is presented as a basic structure upon which more sophisticated algorithms can be built. Possible extensions include probabilistic path ensembles, regime conditioning, confidence bands, asset-specific calibration, dynamic feature selection, multi-series training, and integration into larger decision systems. The main contribution of ECAPR is therefore conceptual and architectural: it provides a structured way to think about multi-step forecasting when the shape of the future path matters as much as the final predicted value.