Sequential Pattern Averaging Regressor: A Lookup-Based Method for Structural Price Prediction

Sofien Kaabar, CFA
sofien-kaabar@hotmail.com

January 28, 2026

Abstract

In this paper, We introduce the **Sequential Pattern Averaging Regressor (SPAR)**, a non-parametric, lookup-based algorithm for time series forecasting that predicts not only directional movement but also the structural trajectory of future values. Unlike traditional machine learning approaches that learn continuous weight parameters, SPAR operates by encoding historical price movements into discrete directional sequences (up, down, flat), cataloging the outcomes that followed each unique pattern, and retrieving averaged trajectories when matching patterns are observed in new data. The method requires no gradient-based optimization, is fully interpretable, and provides natural confidence measures through pattern frequency counts and performance measures. We formalize the mathematical framework underlying SPAR, establish its theoretical properties, and evaluate its performance across three experimental settings: (i) clean synthetic sequences with embedded deterministic patterns, (ii) noisy synthetic sequences where signal-to-noise ratio varies, and (iii) real-world financial data including equities and foreign exchange markets.

1 Introduction

Time series forecasting remains one of the most challenging problems in machine learning and quantitative finance. The ability to predict future values—or at minimum, future directional movements—from historical observations has profound implications for algorithmic trading, risk management, economic planning, and resource allocation. Over the past decades, the field has witnessed remarkable progress, from classical statistical methods such as ARIMA and exponential smoothing, to modern deep learning architectures including LSTMs, Transformers, and their temporal variants.

Despite these advances, a fundamental tension persists between model complexity and interpretability. Neural network-based approaches, while powerful, often function as black boxe. In domains such as finance, where regulatory scrutiny demands explainability and practitioners require intuitive understanding of model behavior, this opacity presents significant barriers to adoption.

This paper introduces the **Sequential Pattern Averaging Regressor (SPAR)**, a simple algorithm that revisits a conceptually basic yet underexplored approach to time series prediction: *pattern matching*. The core intuition is as follows: if we observe a specific sequence of movements in a time series: say, three consecutive increases followed by two decreases, we can search historical data for all instances where this exact pattern occurred, examine what happened subsequently, and use the average of those subsequent movements as our forecast.

This idea is not new; technical analysts have long relied on chart patterns and candlestick formations to inform trading decisions. However, such approaches have traditionally been qualitative, subjective, and difficult to evaluate rigorously. SPAR formalizes this intuition into a systematic, quantitative framework with well-defined mathematical properties.

2 Sequential Pattern Averaging Regressor

The Sequential Pattern Averaging Regressor (SPAR) is a non-parametric, pattern-based forecasting model designed for time series prediction. Unlike traditional regression methods that estimate continuous functional relationships, SPAR discretizes price movements into categorical sequences and generates forecasts by aggregating historical outcomes of matching patterns.

2.1 Motivation

Consider a time series of asset prices $\{P_t\}_{t=1}^{T}$. Traditional approaches to forecasting P_{t+h} given information up to time t typically involve either:

- **Parametric models** (e.g., ARIMA, GARCH) that assume specific functional forms for the data generating process
- **Machine learning models** (e.g., neural networks, gradient boosting) that learn complex nonlinear mappings from features to targets

Both approaches operate in continuous feature spaces and can be sensitive to the magnitude of price movements. SPAR takes a fundamentally different approach: it asks not "how much did prices change?" but rather "in what sequence of directions did prices move?" This discretization provides robustness to scale and focuses on the qualitative structure of price dynamics.

2.2 Algorithm Definition

2.2.1 Pattern Encoding

Let $\{P_t\}_{t=1}^{T}$ be a time series of prices. We define the return at time t as:

$$r_t = \frac{P_t - P_{t-1}}{|P_{t-1}|} \tag{1}$$

Given a threshold parameter $\epsilon \geq 0$, we encode each return into a discrete direction:

$$d_t = \begin{cases} U & \text{if } r_t > \epsilon \\ D & \text{if } r_t < -\epsilon \\ F & \text{if } |r_t| \leq \epsilon \end{cases} \tag{2}$$

where U denotes "up," D denotes "down," and F denotes "flat." Setting $\epsilon = 0$ reduces the encoding to binary (up/down only).

For a window of n consecutive periods ending at time t, the *pattern* is defined as the tuple of directions augmented by the overall trend:

$$\mathcal{P}_t^{(n)} = (d_{t-n+1}, d_{t-n+2}, \ldots, d_t, \sigma_t) \tag{3}$$

where $\sigma_t \in \{P, N\}$ indicates whether the overall movement from P_{t-n} to P_t was positive or negative:

$$\sigma_t = \begin{cases} P & \text{if } P_t \geq P_{t-n} \\ N & \text{if } P_t < P_{t-n} \end{cases} \tag{4}$$

The pattern space $\mathcal{S}$ consists of all possible patterns:

$$|\mathcal{S}| = 3^n \times 2 = 2 \cdot 3^n \tag{5}$$

For binary encoding ($\epsilon = 0$, no flat movements), the pattern space reduces to $|\mathcal{S}| = 2 \cdot 2^n = 2^{n+1}$.

2.2.2 Training Phase

During training, SPAR scans the historical data and, for each pattern $\mathcal{P}$, records the subsequent outcomes. Specifically, for each occurrence of pattern $\mathcal{P}_t^{(n)}$ in the training set, we store:

1. The sequence of returns over the forecast horizon: $\mathbf{r}_t^+ = (r_{t+1}, r_{t+2}, \ldots, r_{t+n})$
2. The total return over the horizon: $R_t^+ = \frac{P_{t+n} - P_t}{|P_t|}$
3. The outcome direction: $\beta_t = \mathbf{1}[R_t^+ > 0]$

Let $\mathcal{H}(\mathcal{P})$ denote the set of all historical occurrences of pattern $\mathcal{P}$, and let $|\mathcal{H}(\mathcal{P})| = m$ be the number of matches.

2.2.3 Signal Generation via Majority Vote

Rather than averaging all outcomes (which can produce muddled signals when bullish and bearish outcomes partially cancel), SPAR employs a majority vote mechanism.

Define the bullish and bearish subsets:

$$\mathcal{H}^+(\mathcal{P}) = \{i \in \mathcal{H}(\mathcal{P}) : R_i^+ > 0\} \tag{6}$$
$$\mathcal{H}^-(\mathcal{P}) = \{i \in \mathcal{H}(\mathcal{P}) : R_i^+ \leq 0\} \tag{7}$$

The signal is determined by majority:

$$S(\mathcal{P}) = \begin{cases} \text{bullish} & \text{if } |\mathcal{H}^+(\mathcal{P})| > |\mathcal{H}^-(\mathcal{P})| \\ \text{bearish} & \text{if } |\mathcal{H}^-(\mathcal{P})| > |\mathcal{H}^+(\mathcal{P})| \\ \text{flat} & \text{otherwise} \end{cases} \tag{8}$$

The confidence in the signal is measured by the majority ratio:

$$\rho(\mathcal{P}) = \frac{\max(|\mathcal{H}^+(\mathcal{P})|, |\mathcal{H}^-(\mathcal{P})|)}{|\mathcal{H}(\mathcal{P})|} \tag{9}$$

2.2.4 Forecast Construction

Once the signal direction is determined, the forecast is constructed using only the outcomes that match the signal direction. Let $\mathcal{H}^*(\mathcal{P})$ denote the winning subset (either $\mathcal{H}^+$ or $\mathcal{H}^-$).

Structural Forecast. The median return path across matching outcomes preserves the typical trajectory structure:

$$\hat{\mathbf{r}}^+ = \text{median}_{i \in \mathcal{H}^*(\mathcal{P})} \left(\mathbf{r}_i^+ \right) \tag{10}$$

The predicted prices are then:

$$\hat{P}_{t+k} = \hat{P}_{t+k-1} \cdot (1 + \hat{r}_{t+k}), \quad k = 1, \ldots, n \tag{11}$$

with $\hat{P}_t = P_t$ (the current price).

Linear Forecast. Alternatively, a simplified linear interpolation uses only the median total return:

$$\hat{R}^+ = \text{median}_{i \in \mathcal{H}^*(\mathcal{P})} \left(R_i^+ \right) \tag{12}$$

$$\hat{P}_{t+k} = P_t + \frac{k}{n} \cdot P_t \cdot \hat{R}^+, \quad k = 1, \ldots, n \tag{13}$$

2.2.5 Minimum Instance Threshold

To ensure statistical reliability, predictions are only generated when the majority subset contains at least $m_{\min}$ instances:

$$|\mathcal{H}^*(\mathcal{P})| \geq m_{\min} \tag{14}$$

Patterns failing this criterion return a flat (no-trade) signal.

2.3 Optional Preprocessing: EMA Smoothing

To reduce noise sensitivity, an optional exponential moving average (EMA) smoothing can be applied before pattern encoding:

$$\tilde{P}_t = \alpha P_t + (1 - \alpha) \tilde{P}_{t-1}, \quad \alpha = \frac{2}{k+1} \tag{15}$$

where k is the smoothing period. Pattern encoding then operates on the smoothed series $\{\tilde{P}_t\}$.

2.4 Computational Complexity

Training. Scanning the training data of length T requires $O(T)$ operations. For each position, pattern encoding requires $O(n)$ comparisons. Total training complexity: $O(Tn)$.

Prediction. Pattern lookup in a hash table is $O(1)$ on average. Computing the median of m sequences of length n requires $O(mn)$ operations, though this can be precomputed during training.

Space. Storage scales with the number of unique patterns observed, bounded by $\min(T, 2 \cdot 3^n)$, each storing $O(m \cdot n)$ outcome sequences.

Aspect	**SPAR**	**k-NN**
Feature space	Discrete (categorical)	Continuous
Distance metric	Exact match	Euclidean/DTW
Neighbors	All exact matches	k closest
Scale sensitivity	None	High
Magnitude sensitivity	None	High
Prediction	Median of filtered matches	Weighted average of k
Hyperparameters	n, $m_{\min}$, ϵ	k, distance metric

Table 1: Comparison between SPAR and k-Nearest Neighbors regression.

2.5 Comparison with Related Methods

2.5.1 SPAR vs. k-Nearest Neighbors

The fundamental distinction is that SPAR operates on *symbolic* representations while k-NN operates on *numeric* representations. SPAR asks "has this exact directional pattern occurred before?" while k-NN asks "what are the most similar historical windows in terms of continuous distance?"

2.5.2 SPAR vs. Linear Regression

Aspect	**SPAR**	**Linear Regression**
Model class	Non-parametric	Parametric
Assumptions	Pattern recurrence	Linear relationship
Functional form	None	$y = X\beta + \varepsilon$
Training	Pattern aggregation	Least squares
Extrapolation	Not possible	Possible
Interpretability	Pattern frequencies	Coefficients

Table 2: Comparison between SPAR and Linear Regression.

SPAR makes no assumptions about the functional form of the relationship between past and future prices. It relies solely on the empirical distribution of outcomes following each pattern. Linear regression, by contrast, assumes a linear mapping and estimates global coefficients.

2.6 Advantages

1. **Scale invariance.** SPAR is insensitive to the absolute magnitude of price movements. A 10% increase and a 0.1% increase are both encoded as "up," making the model robust across different volatility regimes and asset classes.
2. **Interpretability.** Patterns are human-readable sequences (e.g., "UUDU|P" means three ups, one down, overall positive). Practitioners can inspect which patterns drive predictions.
3. **No distributional assumptions.** SPAR does not assume normality, stationarity, or any parametric distribution for returns.
4. **Majority filtering.** By separating bullish and bearish outcomes before aggregation, SPAR avoids the signal dilution that occurs when opposing outcomes cancel.

5. **Computational efficiency.** Training and prediction are fast, with $O(Tn)$ training complexity and $O(1)$ lookup.

2.7 Limitations

1. **Pattern space explosion.** The number of possible patterns grows as $O(3^n)$, limiting practical field sizes to approximately $n \leq$ 10-15. For $n = 15$, the pattern space exceeds 28 million, requiring substantial training data for adequate coverage. The recommended horizon is between 3 and 6 time steps.

2. **Data requirements.** Meaningful predictions require multiple instances of each pattern. Rare patterns may never accumulate sufficient history, reducing coverage (the fraction of test instances where predictions can be made).

3. **Information loss.** Discretization discards magnitude information. A day with a 5% gain and a day with a 0.01% gain are treated identically, potentially losing predictive signal.

4. **No extrapolation.** SPAR cannot predict outcomes for patterns never observed in training. Novel market regimes producing new patterns will receive no signal. This can be fixed with continuous retraining.

5. **Stationarity assumption.** SPAR implicitly assumes that the conditional distribution of outcomes given a pattern remains stable over time. Structural breaks or regime changes may invalidate historical patterns.

6. **Threshold sensitivity.** The flat threshold ϵ and minimum instances $m_{\min}$ require tuning. Poor choices can lead to either excessive flat signals or unreliable predictions.

2.8 Summary

SPAR offers a transparent, non-parametric approach to time series forecasting based on pattern matching and outcome aggregation. Its categorical encoding provides robustness to scale and interpretability, while the majority vote mechanism ensures clean directional signals. The primary trade-offs are the exponential growth of pattern space with field size and the inherent information loss from discretization. In the following section, we present empirical examples demonstrating SPAR's performance across various financial instruments.

3 Empirical Validation on Clean Synthetic Data

Before applying SPAR to real financial data, we validate its behavior on controlled synthetic time series where the underlying patterns are known. This allows us to verify that the algorithm correctly identifies and exploits predictable structures. No smoothing will be applied in the experiments.

3.1 Experimental Setup

We generate three types of synthetic time series with increasing complexity:

1. **Deterministic Pattern**: A repeating sequence of fixed percentage changes, representing perfectly predictable dynamics.

2. **Sine Wave**: A smooth periodic oscillation, representing regular cyclical behavior.
3. **Composite Wave**: A superposition of multiple frequencies, representing more complex but still deterministic dynamics.

For all experiments, we use field size $n = 5$, minimum instances $m_{\text{min}} = 5$, train/test split of 70%/30%, and evaluate using a simple directional hit ratio.

3.2 Data Generation

3.2.1 Deterministic Pattern

The deterministic series follows a fixed repeating pattern of returns:

$$P_t = P_{t-1} \cdot (1 + r_{(t \mod k)} + \delta) \tag{16}$$

where $\mathbf{r} = (r_0, r_1, \ldots, r_{k-1})$ is a predefined pattern of length k and δ is a small drift term. We use:

$$\mathbf{r} = (+2\%, -1\%, +3\%, -2\%, +1\%) \tag{17}$$

with $\delta = 0.001$ (0.1% drift) and $P_0 = 100$.

3.2.2 Sine Wave

The sine wave provides smooth, periodic oscillations:

$$P_t = A + B \sin\left(\frac{2\pi \cdot c \cdot t}{T}\right) \tag{18}$$

where $A = 100$ is the base level, $B = 20$ is the amplitude, $c = 10$ is the number of complete cycles, and $T = 2000$ is the total number of points.

3.2.3 Composite Wave

The composite wave combines three sinusoidal components at different frequencies:

$$P_t = A + B_1 \sin(\omega_1 t) + B_2 \sin(\omega_2 t) + B_3 \sin(\omega_3 t) \tag{19}$$

with base $A = 100$, amplitudes $(B_1, B_2, B_3) = (15, 8, 4)$, and frequency ratios $(\omega_1, \omega_2, \omega_3) = (1, 2.5, 5)$ relative to the fundamental period.

3.3 Results

Figure 1 presents the SPAR forecasts on each synthetic series. Green forecast lines indicate correct directional predictions (price moved in the predicted direction at some point within the horizon), while red lines indicate incorrect predictions.

3.4 Analysis

The results in Table 3 reveal a clear relationship between series complexity and model performance:

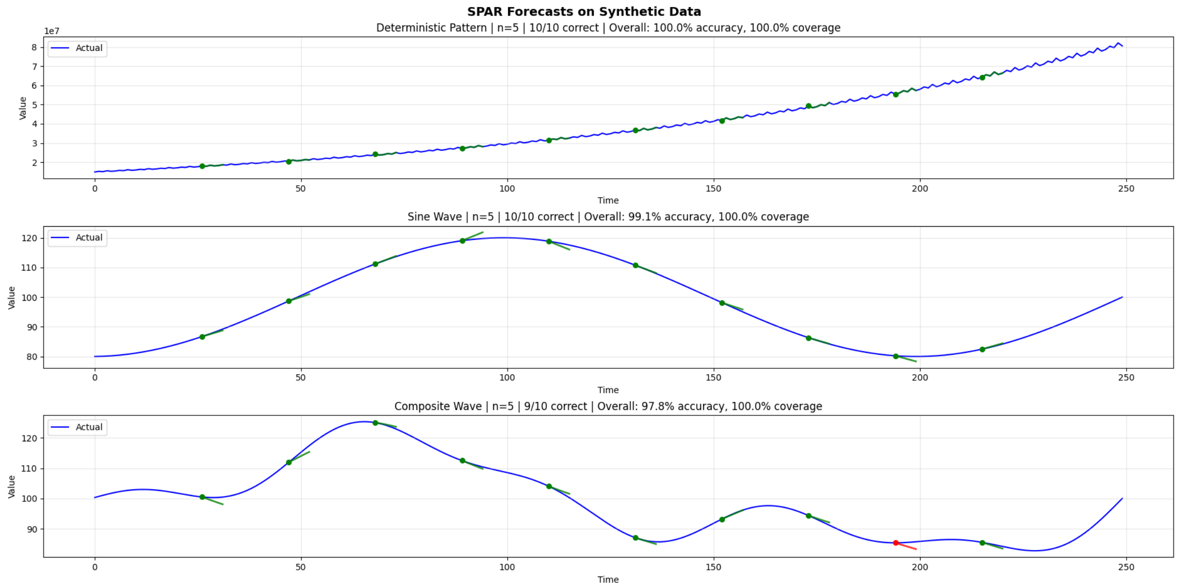

Figure 1: **SPAR forecasts ($n = 5$) on synthetic time series**. Top: Deterministic pattern. Middle: Sine wave. Bottom: Composite wave. Green forecasts indicate correct event-based predictions; red indicates incorrect.

Series Type	**Accuracy**	**Coverage**	**Patterns**
Deterministic	100%	100%	Low
Sine Wave	99.1%	100%	Moderate
Composite Wave	97.8%	100%	High

Table 3: SPAR performance summary on synthetic data with $n = 5$.

Pattern Uniqueness vs. Outcome Consistency. SPAR performs best when each pattern maps to a consistent outcome. In the deterministic case, this mapping is exact. In the sine wave, patterns are nearly unique to their phase position. In the composite wave, the same pattern can occur at different phases of the component oscillations, reducing outcome consistency.

Field Size Sensitivity. The choice of $n = 5$ works well for these experiments, but different series may benefit from different field sizes. The deterministic pattern with period 5 is perfectly matched; a period-7 pattern would require $n = 7$ for optimal capture.

These controlled experiments validate the algorithm's core mechanics before proceeding to the inherently noisier domain of real financial data in the following section.

4 Empirical Validation on Noisy Synthetic Data

Having validated SPAR on clean synthetic data, we now examine its behavior under noisy conditions. Real-world financial data contains substantial noise from microstructure effects, measurement error, and stochastic volatility. Understanding how SPAR degrades under noise is essential for practical applications.

4.1 Noise Model

We corrupt the clean synthetic series by adding Gaussian noise proportional to the series' volatility:

$$\tilde{P}_t = P_t + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \sigma_\varepsilon^2) \tag{20}$$

where the noise standard deviation is defined as a fraction λ of the series' typical movement:

$$\sigma_\varepsilon = \lambda \cdot \mathrm{std}(\Delta P_t) \tag{21}$$

We test two noise levels:

- **Moderate noise**: $\lambda = 0.25$ (25% of movement volatility)
- **High noise**: $\lambda = 0.75$ (75% of movement volatility)

This formulation ensures that noise scales appropriately with each series' natural dynamics.

4.2 Hypothesized Effects

Noise corrupts the directional encoding. A true "Up" movement may be encoded as "Down" or "Flat" if noise reverses or dampens the underlying signal. This fragments coherent patterns into multiple noisy variants, reducing the instance count for any single pattern. Even when patterns are correctly identified, the recorded outcomes are corrupted by noise. The majority vote mechanism partially mitigates this by filtering to the dominant direction, but the median path estimate becomes noisier.

4.3 Results: 25% Noise

Figure 2 presents SPAR forecasts on synthetic data with 25% noise.

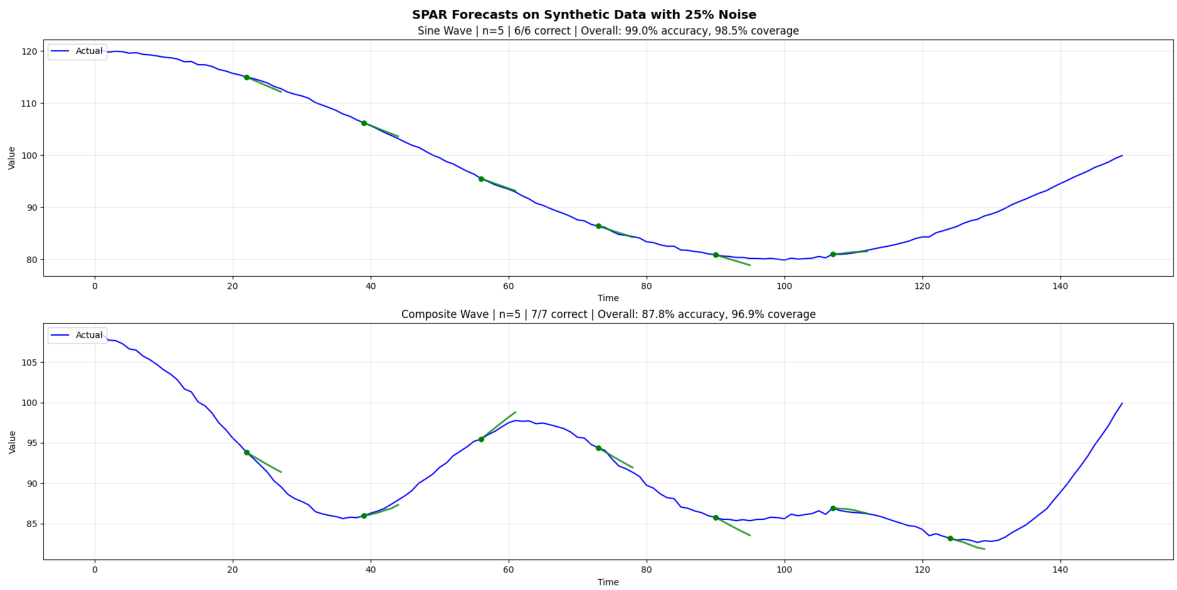

Figure 2: **SPAR forecasts ($n = 5$) on synthetic time series**. Top: Sine wave. Bottom: Composite wave. Green forecasts indicate correct event-based predictions; red indicates incorrect.

4.4 Results: 75% Noise

Figure 3 presents SPAR forecasts under severe noise conditions.

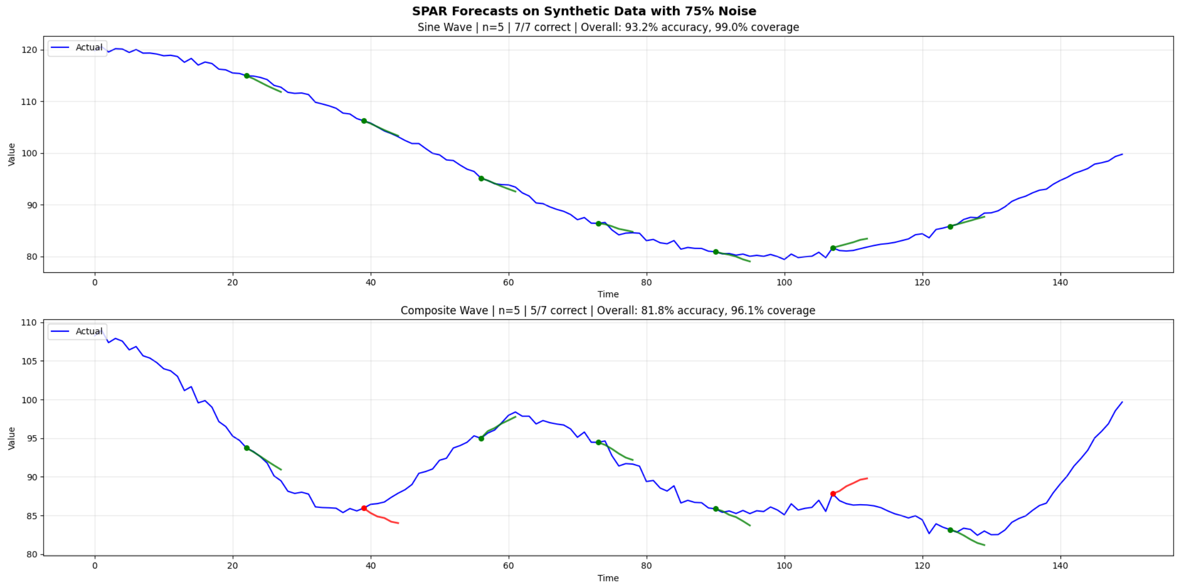

Figure 3: **SPAR forecasts ($n = 5$) on synthetic time series with 75% noise**. Top: Sine wave. Bottom: Composite wave.

4.5 Summary of Noise Effects

Series Type	Accuracy	Coverage	Patterns
Sine Wave (25% noise)	99.0%	98.5%	Moderate
Sine Wave (75% noise)	93.2%	99.0%	Moderate
Noisy Composite Wave (25% noise)	87.7%	96.9%	High
Noisy Composite Wave (75% noise)	81.8%	96.1%	High

Table 4: SPAR performance summary on noisy synthetic data with $n = 5$.

SPAR exhibits predictable degradation under noise, with performance primarily determined by the signal-to-noise ratio of the underlying process. The majority vote mechanism provides partial robustness by filtering inconsistent outcomes, but cannot recover signal that is lost during pattern encoding.

In the following section, we apply SPAR to real financial data to assess its performance in authentic market conditions.

5 Application to Real Financial Data

Having established SPAR's behavior on synthetic data with known properties, we now evaluate its performance on real financial instruments. This section examines four diverse assets: two currency pairs (EUR/USD, USD/JPY), one equity (Tesla), and one commodity (Gold).

5.1 Data Description

We obtain daily closing prices from Yahoo Finance spanning approximately 20 years of history. These assets represent different market microstructures and return distributions:

- **EUR/USD**: The most liquid currency pair globally, characterized by relatively small daily moves and mean-reverting behavior over medium horizons.
- **USD/JPY**: A major currency pair influenced by Bank of Japan interventions and carry trade dynamics, exhibiting occasional strong trends.
- **Tesla**: A high-beta equity with substantial idiosyncratic volatility, prone to momentum and sentiment-driven moves.
- **Gold**: A commodity and safe-haven asset with regime-dependent behavior, trending during risk-off periods and ranging otherwise.

5.2 Experimental Configuration

We apply SPAR with the following parameters:

- Field size: $n = 5$ (5-day patterns, 5-day forecast horizon)
- Minimum instances: $m_{\text{min}} = 30$
- Train/test split: 70%/30%
- Forecast type: Structural (preserving median path shape)

The 70/30 split provides approximately 14 years of training data and 6 years of out-of-sample testing for each asset.

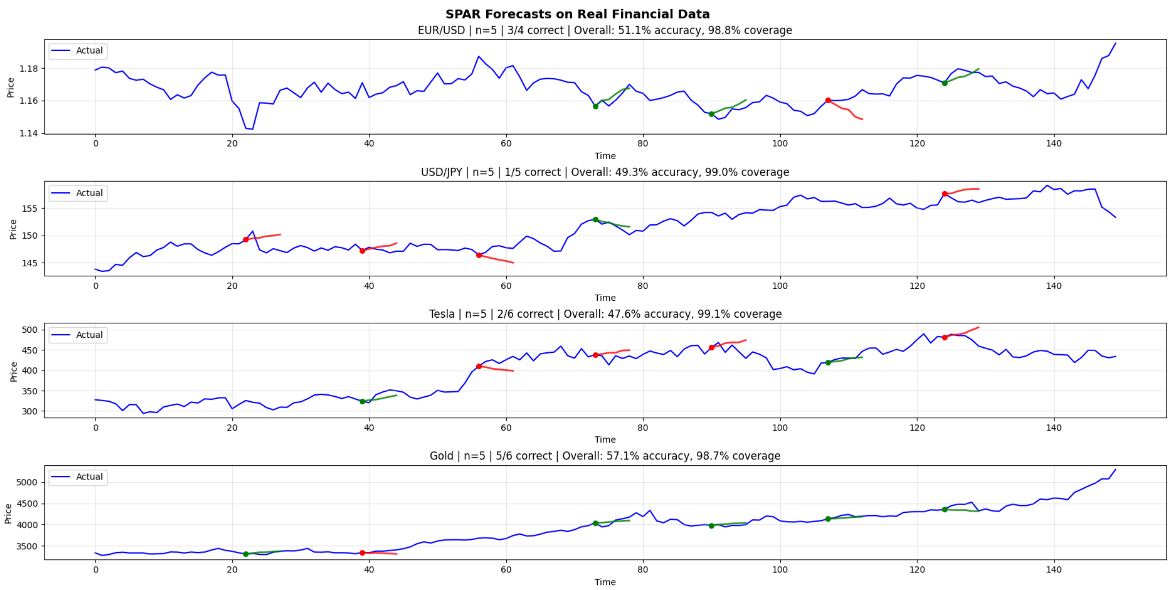

Figure 4: **SPAR forecasts ($n = 5$) on real financial data.**

5.3 Results

The following table summarizes the results of the algorithm.

Series Type	Accuracy	Coverage	Patterns
EUR/USD	51.2%	98.8%	181
USD/JPY	49.3%	99.0%	204
Tesla	47.5%	99.1%	111
Gold	56.5%	98.7%	160

Table 5: SPAR performance summary on real financial data with $n = 5$.

6 Conclusion

SPAR applied to real financial data produces accuracy in the 47–56% range, modestly above random for some assets but not dramatically so. The results are consistent with weak-form market efficiency: past price patterns contain limited predictive information, but not zero.

The pattern-based approach provides transparency—practitioners can inspect which patterns drive predictions and assess their economic rationale. For Tesla, patterns like "UUUUU|P" (five consecutive up days, overall positive) may reflect momentum that persists into the following week. For EUR/USD, similar patterns show no consistent follow-through, reflecting the currency market's efficiency.

These findings suggest SPAR may be a useful component in a broader trading system, particularly for:

- High-volatility assets with momentum characteristics
- Longer timeframes (weekly, monthly) where noise is reduced
- Combination with fundamental or sentiment signals
- Comparing to other more sophisticated models