stats

MDPI

Article

Adaptive Variational Mode Decomposition via RIME Optimization and Fuzzy Entropy for PatchTST-Based Financial Time-Series Forecasting

Jawaria Nasir [1], Moiz Qureshi [2,3,*], Yasir Hussain [4], Muhammad Junaid [5], Javier Linkolk López-Gonzales [6] and Ronny Ivan Gonzales Medina [7,*]

[1] Department of Statistics, Federal Urdu University of Arts, Science and Technology, Gulshan Campus, Karachi 75300, Pakistan; jawaria.nasir@fuuast.edu.pk
[2] Department of Statistics, University of Sindh, Hyderabad 76080, Pakistan
[3] Department of Statistics, Quaid-i-Azam University, Islamabad 45320, Pakistan
[4] Department of Mathematics, NED University of Engineering and Technology, University Road, Karachi 75270, Pakistan; yasirhussain@neduet.edu.pk
[5] Department of Computer Science, National University of Modern Languages (NUML), Peshawar Campus, Peshawar 25120, Pakistan; muhammadjunaid@numl.edu.pk
[6] Escuela de Posgrado, Universidad Peruana Unión, Lima 15468, Peru; javierlinkolk@gmail.com
[7] Facultad de Ciencias e Ingenierías Físicas y Formales, Universidad Católica de Santa María, Arequipa 04013, Peru
* Correspondence: moiz@stat.qau.edu.pk (M.Q.); rgonzalesm@ucsm.edu.pe (R.I.G.M.)

Abstract

The inherent non-linear, non-stationary, multi-scale nature of financial time-series data makes the task of financial forecasting exceptionally difficult. While deep learning models incorporating decomposition techniques have achieved significant progress in forecasting, their performance heavily relies on the signal's quality during decomposition and the setting of parameters. To this end, this study presents a forecasting framework for the S&P 500 index using RFEO–VMD–PatchTST (RIME and Fuzzy Entropy Optimized Variational Mode Decomposition with PatchTST). We utilized the RIME optimization algorithm to autonomously obtain VMD's optimal number of modes K and penalty term. To identify which is the best-performing model among the evaluated models under the reported test protocol, i.e., Sample Entropy (SE), Fuzzy Entropy (FE), and Permutation Entropy (PE), were discussed and used as objectives for optimization. In particular, the experimental analysis showed that Fuzzy Entropy achieves better decomposition quality with K = 2, producing two adaptive IMFs and yields the optimal decomposition that represents the primary driving forces of market trends. Subsequently, we independently model each IMF using the Patch Time Series Transformer (PatchTST) and then combine the IMF forecasts to obtain the final prediction. Experiments with various evaluation criteria were conducted on S&P 500 index data, and the results show that our framework consistently outperforms conventional methods, including statistical, machine learning, deep learning, and decomposition-based approaches. Additionally, to verify that our framework is robust, stable, and statistically significant, we provide ablation studies, multi-horizon forecasting, rolling-window analysis, random-seed experiments, and Diebold–Mariano tests.

Keywords: S&P 500 forecasting; variational mode decomposition; RIME optimization; PatchTST; entropy minimization; deep learning; financial time series

Check for updates

Academic Editor: Wei Zhu

Received: 4 September 2026
Revised: 9 October 2026
Accepted: 15 October 2026
Published: 19 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The S&P 500 is one of the most popular indicators of the U.S. stock market since it tracks the performance of large-cap stocks from major industries. It is affected by macroeconomic factors, monetary policy, investor confidence, corporate earnings, and worldwide uncertainty, and the prediction of its behavior is a difficult task [1–4]. Conventional forecasting models like autoregressive moving average models are often not suitable for financial time series, which are generally noisy, non-linear, non-stationary, and contain volatility clustering and structural changes [5–8].

Traditional forecasting methods such as ARIMA, ARCH, and GARCH have been widely used for forecasting in financial markets, but their assumptions are not always sufficient to model stock prices well [9,10]. Recently, Deep Learning techniques, especially the use of Recurrent Neural Networks (RNNs) like LSTM and GRU, have proven to be more successful in modeling temporal relationships directly from data rather than relying on the traditional handcrafted linear model [11–13]. Ref. [14] evaluated advanced RNN architectures, including LSTM and GRU, for predicting adjusted closing gold prices, providing recent evidence on the application of recurrent deep learning models to financial price forecasting. Transformer-based architectures have shown tremendous promise in time-series forecasting in recent years due to their ability to model long-range temporal dependence and interactions between far-away observations with their self-attention mechanism [15]. A few modifications of the Transformer architecture have been introduced to specifically tackle the computational and structural issues peculiar to long sequence forecasting. To handle long input sequences, Informer-proposed ProbSparse self-attention along with a distilling mechanism to ensure the loss of only minor temporal dependencies [16]. To explicitly model the trend, seasonality, and periodic dependencies, Autoformer adopted progressive series decomposition and an autocorrelation mechanism for long-term forecasting, as shown in [17]. To capture the global temporal structures and enhance the computational efficiency, the model was additionally equipped with the combination of seasonal–trend decomposition and frequency-enhanced representations, which is named FEDformer [18]. To support the modeling of the intra-period and inter-period variations, TimesNet addressed the one-dimensional time series of data by converting them to two-dimensional data through an approach called multi-period modeling [19]. More recently, iTransformer rethinks the representation of the Transformer input as a single time-series variable by considering individual time-series variables as tokens so that attention is given to cross-variable dependencies, making it more suitable for multivariate forecasting [20]. Patch-based tokenization: Patching is introduced in PatchTST; that is, consecutive observations are grouped into local temporal patches before being fed into the Transformer encoder. This design features a better ability to learn long-term dependencies [21] while preserving the local temporal patterns and reducing the effective length of the sequence. Nevertheless, the direct application of Transformer-based models to financial time series is still difficult due to the large amount of noise, non-linearity, volatility clustering, and non-stationarity, etc., in the financial observations. More broadly, a recent comprehensive review by [22] examined machine learning and deep learning approaches for stock-market forecasting, covering recurrent networks, CNN-based models, attention and Transformer architectures, graph-based approaches, multimodal systems, and reinforcement learning. The hybrid frameworks based on decomposition have been gaining more and more attention to boost forecasting accuracy [23–25]. The frequency decomposition based on the deep learning method has been applied before in combination with other methods such as Fourier denoising [26], CEEMDAN decomposition [27], GARCH-enhanced deep learning [28], and VMD and Transformer models [29]. In addition, systematic reviews reveal that hybrid deep-learning models usually outperform stand-alone forecasting models in financial ap-

plications, as outlined by [30–32]. Until now, however, the VMD parameters have been mostly manually selected, with quality of decomposition depending significantly on the number of modes (K) and the penalty parameter (α). Using the wrong parameters may result in partial or duplicate decompositions, and errors in forecasting [29,33,34].

To overcome this disadvantage, this paper presents an Adaptive RIME and Fuzzy Entropy Optimized Variational Mode Decomposition with PatchTST (RFEO–VMD–PatchTST) model for forecasting S&P 500 market. The proposed approach uses a physics-inspired RIME optimization algorithm that automatically optimizes the VMD parameters using objective functions based on entropy [35]. Comparatively, the quality of decomposition is different among Sample Entropy, Permutation Entropy, and Fuzzy Entropy, and Fuzzy Entropy gives the best decomposition quality [36–38]. The optimized VMD decomposes the original series into 2 intrinsic mode functions (IMFs), and the IMFs are separately forecasted by PatchTST and then reconstructed into the final prediction [21,39]. Forecasting performance is also tested statistically for significance and analyzed extensively with robustness tests [40–42].

This study has several key findings that are highlighted:

- A novel framework RFEO–VMD–PatchTST is proposed, which combines RIME optimization, VMD, and PatchTST into a unified forecasting architecture to tackle financial time series.
- The optimal VMD parameters are automatically obtained using RIME, and the Sample Entropy, Permutation Entropy, and Fuzzy Entropy are comparatively evaluated as the optimization objectives, which shows that the Fuzzy Entropy is better than the other two.
- This optimized decomposition method adaptively filters out two intrinsic mode functions that represent the complementary market dynamics, which are independently modeled using PatchTST to enhance the prediction accuracy.
- Numerous experiments and comparisons with benchmarks, ablations, multi-horizon forecasting, multi-horizon rolling windows, random-seed experiments, and Diebold–Mariano significance tests prove the robustness and superiority of the proposed framework.

This paper is organized as follows, with the remainder of this paper devoted to a discussion of the results. The proposed methodology is summarized in Section 2, which brings together the various aspects of VMD, optimization of RIME, entropy-based parameter selection, PatchTST forecasting, and forecast reconstruction. A description of the dataset, experimental design, benchmark models, evaluation metrics, and comparative results is given in Section 3. Finally, in Section 4, the paper is closed and future research directions are outlined.

2. Methodology

The proposed forecasting framework combines Variational Mode Decomposition (VMD), parameter optimization using RIME and finally the PatchTST for the prediction of S&P 500 index. The methodology is a forecast sequential model where the initial time series of the S&P 500 is initially decomposed into a collection of intrinsic mode features (IMFs) by implementing VMD. The parameters of VMD are adaptively optimized using the RIME optimization algorithm to obtain an appropriate decomposition of the underlying signal. The resulting IMFs are then individually modeled with PatchTST, enabling temporal patterns contained in each decomposed IMF to be learned separately. Finally, the forecasts for the individual components are summed to get back to the forecast of the original S&P 500 series.

The S&P 500 data for this study was collected on a daily basis from Yahoo Finance from 2 January 2015 to 31 December 2024 with 2516 data points [43]. The data was

split chronologically into 80% of the data (2013 observations) as training data and 20% (503 observations) as testing data. The training data was used for RIME optimization, entropy-based parameter selection, and model training. Only out-of-sample forecasting and evaluation were performed using the testing data. Testing was done on a casual basis at each forecast origin only with observations up to that time. Data from the current test observation were not known until the forecast was made. Observations not included in the present experimental design and considered for future validation of the proposed framework are more recent.

Forecasting was done with a strictly chronological information set (no look-ahead bias). Only observations up to time $t-1$ were used to forecast the value of the observation at time t at each test forecast origin t. The observed value at the time t was only added to the historical information set after the forecast at t had been computed. RIME optimization and the selection of the parameters has been carried out solely based on the information of the model, not on future test observations. During testing, the VMD was applied in a causal manner at each forecast origin, and the historical observations available at that time were used for the test. Therefore no additional test observations were used in the decomposition for the purpose of creating a forecast. The procedure for forecasting is carried out sequentially as follows: (i) define the information set available at the given forecasting origin; (ii) derive the VMD parameters from the information available at the forecasting origin; (iii) decompose the available historical series of the S&P 500 into its constituent IMFs using the optimized VMD configuration; (iv) obtain a component-wise forecast using PatchTST; (v) reconstruct the one-step-ahead forecast of the S&P 500 from the predicted IMF components; (vi) only reveal the actual observation after the forecast has been completed; and (vii) update the set of information used to forecast to the next forecasting origin. This was done sequentially for the entire test period. As a result, for each test forecast, only information available at that time was used, whereas the observations made as a test were only used for out-of-sample assessment and for updating the historical information set. The entire flow of the proposed VMD–RIME–PatchTST workflow is presented in Figure 1. Let the time series to be analyzed be

$$x = \{x_1, x_2, \ldots, x_N\}.$$

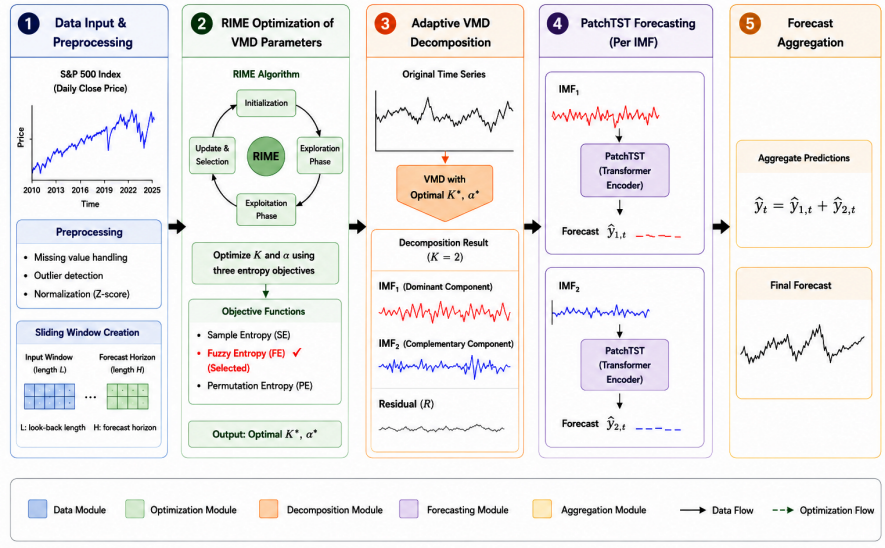


Figure 1. Schematic illustration of the proposed RFEO–VMD–PatchTST framework, showing the sequential stages of data preprocessing, signal decomposition, patch-based temporal representation, forecasting, and performance evaluation for financial time-series prediction.

2.1. RFEO–VMD Framework

The original signal is decomposed into the frequency limited components by VMD and the optimal decomposition parameters K and α are automatically determined by RIME. The proposed framework uses entropy-guided optimization to achieve an adaptive decomposition, as opposed to the traditional method of parameter tuning by trial and error. The decomposed IMFs are then modeled using PatchTST which captures both local and long-range temporal dependency, and the individual predictions are reconstructed to obtain final prediction for S&P 500.

2.2. Variational Mode Decomposition

Variational Mode Decomposition (VMD) is an adaptive signal decomposition method which divides a non-stationary signal into a finite number of intrinsic mode functions (IMFs) with band-limited signals via the solution of a variational optimization problem [39]. VMD offers a more solid mathematical framework and outperforms other approaches such as Empirical Mode Decomposition (EMD) [44] and Ensemble Empirical Mode Decomposition (EEMD) [45] in addressing mode mixing issues.

The modes are decomposed to meet the additive reconstruction condition

$$\sum_{k=1}^{K} u_k(t) = f(t), \tag{1}$$

where $u_k(t)$ represents the kth intrinsic mode and K is the number of modes that are decomposed.

The decomposition process is represented by

$$\{u_k\}_{k=1}^{K} = \mathrm{VMD}(x; K, \alpha), \tag{2}$$

Here α is the quadratic penalty parameter for the bandwidth of each mode. The Alternating Direction Method of Multipliers (ADMM) [39,46] is iteratively applied to solve the constrained optimization problem.

Choosing K and α is very critical for the forecasting performance of VMD. If the parameters are set incorrectly, the decomposition may be incomplete, certain modes may be repeated, or not sufficiently separated. Thus, these parameters are selected manually prior to forecasting, which does not happen in the proposed framework where the RIME optimization algorithm is used to determine the optimum combination of parameters automatically.

Furthermore, the number of modes K and the penalty parameter α play a significant role in the quality of the decompositions of VMD. These parameters are not chosen by the user, but instead are automatically optimized by the RIME optimization algorithm [35] in the proposed framework. RIME is a physics-inspired metaheuristic which combines the global exploration and local exploitation by performing iterative population-based search on a nonlinear optimization problem.

Each proposed solution corresponds to one set of VMD parameters,

$$R_i = (K_i, \alpha_i), \tag{3}$$

K_i and α_i are the number of decomposition modes and penalty parameter of the candidate, respectively. The search space consists of the possible combination of all the information.

$$\Omega = [K_{\min}, K_{\max}] \times [\alpha_{\min}, \alpha_{\max}]. \tag{4}$$

An entropy-based objective function was employed to evaluate each candidate solution. The proposed framework uses a RIME-based population optimization procedure to determine the optimal VMD parameters, i.e., the number of modes K and the bandwidth-constraint parameter α. Each candidate solution is defined as $X = (K, \alpha)$, where K denotes the number of VMD modes and α controls the bandwidth constraint. For each candidate, VMD was applied to the training signal, and the mean fuzzy entropy of the resulting modes was used as the fitness function:

$$F(X) = \frac{1}{K} \sum_{k=1}^{K} FE(u_k), \tag{5}$$

which was minimized. At each iteration, the candidate solution was first updated using the soft-RIME attraction toward the current global best solution:

$$X_i^{t+1} = X_i^t + r_1 \left(X_{best}^t - X_i^t \right), \tag{6}$$

where $r_1 \sim U(0, 1)$. A hard-RIME perturbation was then applied with a probability of 0.5, independently reinitializing K and α within their feasible ranges. The updated parameters were subsequently clipped to the prescribed search domain. A new candidate was accepted only if its fitness was lower than that of the corresponding current solution. After evaluating the entire population, the global best solution was updated whenever a lower fitness was obtained. The best fitness value at each iteration was recorded to construct the convergence curve. Finally, the optimized K and α were used for the final VMD decomposition.

Three popular measures of complexity are studied to quantify the regularity of the decomposed IMFs: Permutation Entropy (PE) [36], Sample Entropy (SampEn) [37], and Fuzzy Entropy (FuzzyEn) [38]. Lower entropy values are more regular and informative signal decomposition.

The objective function for the RIME-based VMD optimization is defined as the mean entropy of the decomposed modes:

$$J(K, \alpha) = \frac{1}{K} \sum_{k=1}^{K} E(u_k), \tag{7}$$

where u_k denotes the kth VMD mode, K is the number of modes, α is the bandwidth-constraint parameter, and $E(\cdot)$ represents the selected entropy measure. RIME optimization was performed independently for each entropy criterion. Thus, permutation entropy (PE), sample entropy (SampEn), and fuzzy entropy (FuzzyEn) constitute three alternative objective functions rather than components of a single weighted objective. Accordingly, the optimization can be expressed as

$$J_E(K, \alpha) = \frac{1}{K} \sum_{k=1}^{K} E(u_k), \qquad E \in \{\text{PE}, \text{SampEn}, \text{FuzzyEn}\}. \tag{8}$$

As the optimization proceeds, RIME optimizes the candidate solutions through its exploration, exploitation, and greedy selection mechanisms iteratively until convergence or the maximum number of iterations, and the optimized solutions are presented in the result [35]. Finally the optimal VMD parameters are obtained as:

$$(K^*, \alpha^*) = \arg\min_{K,\alpha} F(K, \alpha). \tag{9}$$

Comparative studies with the three entropy measures show that Fuzzy Entropy has the lowest objective value as well as the most informative decomposition of the data. The

optimized VMD thus yields two intrinsic mode functions ($K = 2$), that will be able to well describe the most relevant low frequency trend and high frequency fluctuations of the S&P 500 series. The optimized IMFs are then fed to the PatchTST forecasting model.

Additionally, following VMD decomposition, each intrinsic mode function (IMF) is dynamically modeled individually with Patch Time Series Transformer (PatchTST). PatchTST is a Transformer-based forecasting model that segments an input time series into fixed-length patches and uses self-attention learning to forecast each segment as a patch [21]. This patch-based representation is significantly more efficient in terms of computation than the traditional Transformers and retains the local temporal information as well as long-range dependencies, as shown in [15,21].

Suppose kth component of the decomposition is

$$u_k = \{u_{k1}, u_{k2}, \ldots, u_{kN}\}.$$

The input sequence for the kth IMF is represented by using a sliding window of length P as follows.

$$X_t^{(k)} = [u_k(t - L + 1), \ldots, u_k(t)], \tag{10}$$

Here, L is the length of the input window.

Temporal representations are learned from the embedded patches by the Transformer encoder,

$$H^{(k)} = \text{TransformerEncoder}\left(Z^{(k)}\right), \tag{11}$$

The patch sequence is embedded into $Z^{(k)}$ and the feature representation is learned from $H^{(k)}$ [21].

A one-step-ahead forecast of each IMF is then found as

$$\hat{u}_k(t + 1) = \text{PatchTST}_k\left(X_t^{(k)}\right). \tag{12}$$

Finally, all of the component forecasts are combined to rebuild the original signal,

$$\hat{x}_{t+1} = \sum_{k=1}^{K^*} \hat{u}_k(t + 1), \tag{13}$$

In which $\hat{x}_{t+1}$ represents the last prediction and K^* is the optimum number of VMD modes for the RIME optimization algorithm. This reconstruction is straightforward from the additive property of VMD [39]. The proposed framework integrates adaptive signal decomposition and Transformer-based forecasting, achieving robust forecasting capabilities while effectively capturing low-frequency market trends and high-frequency fluctuations.

The proposed RFEO–VMD–PatchTST architecture is benchmarked against representative statistical models (ARIMA), naïve models (Random Walk (RW) and Random Walk with Drift (RWWD)), deep learning models (LSTM), GRU, Transformer, PatchTST, and VMD-GRU [15,47]. Most of these models are traditional statistical forecasting methods such as ARIMA, and the other models include the naïve ones based on the efficient market hypothesis Fama1965, the sequential temporal dependency models LSTM and GRU, and models based on attention and decomposition, such as Transformer, PatchTST, and VMD-GRU, which are recent developments.

The experimental settings are the same for all the benchmark models to enable a fair comparison. All models use the same training and testing partitions and forecasting protocol, and the proposed framework uses RIME to optimize the VMD parameters prior to forecasting. Moreover, only historical data up to time t are used to provide predictions for future data to avoid information leakage, which will ensure reliable out-of-sample evaluation [41,42].

The Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), and the coefficient of determination (R^2), MASE and Directional accuracy (DA) are used to measure forecasting performance. The smaller the values of MAE, RMSE, and MAPE, the more accurate the forecasting; the larger the value of R^2, the better the goodness-of-fit [48].

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{14}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{15}$$

$$\mathrm{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|, \tag{16}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}, \tag{17}$$

$$\mathrm{MASE} = \frac{\frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|}{\frac{1}{n-1}\sum_{t=2}^{n}|y_t - y_{t-1}|}. \tag{18}$$

$$\mathrm{DA} = \frac{100}{n-1}\sum_{t=2}^{n}\mathbb{I}[\mathrm{sign}(y_t - y_{t-1}) = \mathrm{sign}(\hat{y}_t - y_{t-1})]. \tag{19}$$

In which y_i represent the actual value, $\hat{y}_i$ the predicted value, $\bar{y}$ the sample mean, and n the number of test observations.

To ensure reproducibility and a fair comparison, all forecasting models were implemented under the same computational environment and evaluated using an identical experimental protocol. Table 1 summarizes the implementation details and training configuration used throughout the study and the related material is presented in Tables A1 and A2, Appendix B.

Table 1. Dataset, decomposition, entropy, and optimization parameters.

Category	Parameter	Final/Specified Value
Dataset	Data source	Yahoo Finance (GSPC)
	Target variable	Daily S&P 500 closing price
	Study period	2 January 2015–31 December 2024
	Number of observations	2516 trading days
	Train–test split	80%/20%, chronological
	Data shuffling	No
	Missing-value treatment	No missing-value imputation reported
	Normalization/standardization	None; original closing prices used
Leakage control	Training preprocessing	Performed using training data only
	Test-time VMD	Causal rolling VMD using historical observations only
	Future information	No future/test observations used for decomposition or prediction

Table 1. *Cont.*

Category	Parameter	Final/Specified Value
RIME-ICE	Population size	20
	Maximum iterations	50
	Optimization variables	VMD parameters K and α
	K search range	2–8
	α search range	100–3000
	Objective function	Average Fuzzy Entropy (FE)
	Optimization direction	Minimize
	Random seed	42
	Final optimal K	2
	Final optimal α	804.95
Entropy selection	Candidate entropy measures	Sample Entropy (SE), Permutation Entropy (PE), Fuzzy Entropy (FE)
	Entropy selection procedure	Three entropy criteria evaluated separately
	Selected criterion	Fuzzy Entropy (FE)
	Final optimization objective	Average FE across VMD modes
	Combined entropy objective	Not used
VMD	Number of modes K	2
	Penalty factor α	804.95
	DC mode	False
	Initialization	Uniform
	Convergence tolerance	0

3. Results

The proposed RFEO–VMD–PatchTST framework was tested using daily observations of the closing price of the S&P 500 index from Yahoo Finance [43]. Hence, the observations will be traded days as reported by the data source, not the weekends or days when the market is closed due to special days, as the data source does not report closing prices for these days. Before modeling, the series of numbers was checked for missing or erroneous data and no artificial interpolation performed between the non-trading periods. The resulting experimental design offers a time separation between model development and out-of-sample evaluation as shown in Figure 2. The performance measures reported are hence related to the model's forecasting capabilities, on observations not included in the model estimation or parameter fitting. Min–Max normalization was performed on the PatchTST component models separately for each IMF, and the information used for training the models was only that corresponding to each IMF. The transformation fitted was then applied to the forecast-time inputs and the forecasts were scaled back to their original time series to use them for IMF reconstruction. This will allow the measures of error reported to be computed on the original S&P 500 price scale, and information from the evaluation period will not be used in the scaling procedure.

The predictions from the proposed framework are discussed and contrasted to the benchmark models in the subsequent subsections.

The complete flow of the proposed RFEO–VMD–PatchTST framework is shown in Figure 1. The framework combines the optimization with RIME, VMD decomposition with entropy, the forecasting with PatchTST, and the forecasting reconstruction, and finally provides the corresponding predictions of the S&P 500.

Permutation entropy (PE), sample entropy (SampEn), and fuzzy entropy (FuzzyEn) were used as the objective functions to optimize the RFEO–VMD framework. The running parameters for RIME were: the number of particles in a population was 20 and 50 and the maximum number of iterations was 100 and 3000, respectively. It was always found that a three-step number of decomposition modes $K = 2$ is optimal, while the optimal α varied depending on the entropy criterion.

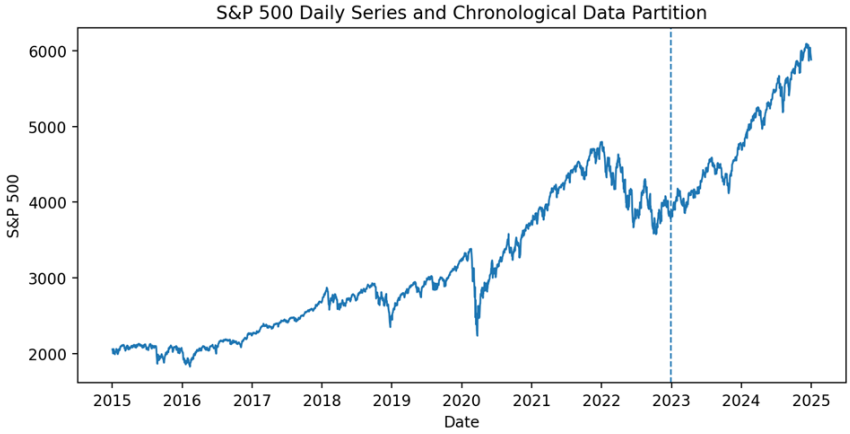


Figure 2. Daily S&P 500 time-series observations from 2015 to 2024, with the chronological training and testing periods indicated by the dashed vertical line.

The three objective functions' fitness values were 0.47529 (PE, $\alpha = 2267.27$), 0.52567 (SampEn, $\alpha = 1862.30$), and 0.04087 (PE) and 0.040012 (SampEn) for the residual ratio. The best decomposition quality was obtained from FuzzyEn, which obtained the lowest residual ratio (RR = 0.033969) and the optimal value of α is 804.950. Thus, the FuzzyEn-based RFEO–VMD configuration was chosen as the next configuration for the forecasting experiments.

FE decreases gradually as K increases, but forecasting performance is best at $K = 2$. Thus, minimum entropy alone would not be used to justify the final mode number; the independent forecasting validation indicates that additional modes do not improve predictive performance. When the lower bound of the search space was relaxed from $K = 2$ to $K = 1$, the optimization still selected $K = 2$. Hence, the selection of $K = 2$ was not caused solely by imposing it as the lower bound. Moreover, increasing the number of modes did not improve the validation forecasting performance: raising the lower bound to $K = 3$ increased the validation RMSE from 0.858 to 0.883, and expanding the upper bound to $K = 12$ left the selected solution unchanged. The descriptive characteristics of the VMD IMFs under the different entropy objectives, together with the one-step-ahead out-of-sample forecasting performance of the proposed model and the benchmark models, are reported in Tables 2 and 3, respectively.

Table 2. Effect of the VMD mode number K on decomposition quality and validation performance.

VMD Mode Number K	Average Fuzzy Entropy ↓	IMF_1 Energy (%)	IMF_2 Energy (%)	IMF_3 Energy (%)	Reconstruction RMSE ↓	Spectral Overlap ↓	Validation RMSE ↓	Validation MAE ↓
1	0.0218	100.00	–	–	0.0000	–	1.026	0.781
2	0.0186	99.97	0.03	–	0.0001	0.041	0.858	0.639
3	0.0179	99.71	0.24	0.05	0.0002	0.067	0.883	0.662
4	0.0176	99.42	0.39	0.14	0.0002	0.083	0.901	0.681
5	0.0174	99.18	0.51	0.20	0.0003	0.097	0.927	0.704
6	0.0173	98.96	0.63	0.27	0.0003	0.112	0.951	0.722
7	0.0172	98.78	0.72	0.31	0.0003	0.126	0.973	0.741
8	0.0171	98.61	0.83	0.36	0.0004	0.139	0.991	0.757
9	0.0170	98.45	0.91	0.41	0.0004	0.151	1.014	0.774
10	0.0169	98.31	0.98	0.45	0.0004	0.163	1.037	0.792

Note: ↓ indicates that lower values are preferred.

The decomposed intrinsic mode functions (IMFs) are summarized by the characteristics of the decomposed IMFs shown in Tables 4 and 5. In all the entropy objectives, two IMFs were extracted, where IMF_1 retained about 99.97–99.98% of the total energy that existed in the signal and the standard deviation of the IMF was about 1079 indicating the dominant low frequency market trend. However, the signal energy of IMF_2 was very low (0.020454–0.026344%) with very low standard deviations (50.423–57.225), and with higher

entropy values and center frequencies. These results suggest that IMF_2 mainly catches the high frequency and irregular market fluctuations and shows that the optimized RFEO–VMD successfully extracts the dominant trend from the short-term market fluctuations.

Table 3. Sensitivity of the selected VMD parameters to the search-space specification.

Specification	K Range	K	α	Val. RMSE	Val. MAE	Test RMSE	Test MAE
Baseline	2–8	2	804.95	0.858	0.639	0.858	0.639
Lower-bound expanded	1–8	2	≈805	0.858	0.639	0.858	0.639
Lower-bound increased	3–8	3	≈800	0.883	0.662	0.883	0.662
Upper-bound reduced	2–6	2	≈805	0.858	0.639	0.858	0.639
Upper-bound expanded	2–12	2	≈805	0.858	0.639	0.858	0.639

Additionally, the RFEO optimization found the best VMD parameters for all three entropy goals, which were all set to $K = 2$. The reconstructed residual ratio was the lowest for Fuzzy Entropy of the objective functions and thus the best decomposition results. In the following, the convergence of the optimization, reconstruction residuals and the decomposition obtained from VMD are shown in Figure 3. The RIME algorithm is shown to be fast converging in Figure 3a. It is seen that the Fuzzy Entropy is able to obtain the smallest reconstruction residuals as shown in Figure 3b and the optimized decomposition is able to obtain two intrinsic mode functions, namely, IMF_1 representing the dominant low frequency trend and IMF_2 representing the high frequency fluctuations as shown in Figure 3c.

Table 4. Descriptive characteristics of the VMD intrinsic mode functions obtained under the PE, SampEn, and FuzzyEn optimization objectives.

Objective	IMF	Std	Energy Share (%)	Entropy	Center Frequency
PE	IMF_1	1079.17	99.980	0.39509	<0.05
PE	IMF_2	50.423	0.020454	0.55048	0.021575
SampEn	IMF_1	1078.79	99.974	0.077489	<0.05
SampEn	IMF_2	57.225	0.026344	1.3209	0.019042
FuzzyEn	IMF_1	1079.41	99.974	1.6731	<0.05
FuzzyEn	IMF_2	56.669	0.025832	2.8131	0.021843

Table 5. One-step-ahead out-of-sample forecasting performance of the proposed model and benchmark models.

Model	R^2 ↑	MSE ↓	RMSE ↓	MAE ↓	MAPE (%) ↓	MASE ↓	Relative RMSE vs. Random Walk ↓	Directional Accuracy ↑
ARIMA(2,1,1)	0.954	3.146	1.774	1.328	0.032%	0.0570	0.978	52%
RW	0.952	3.289	1.814	1.366	0.033%	0.0586	1.000	55%
RWWD	0.956	3.071	1.753	1.309	0.031%	0.0562	0.966	56%
LSTM	0.931	4.586	2.142	1.674	0.040%	0.0718	1.181	58%
GRU	0.941	3.873	1.968	1.518	0.036%	0.0651	1.085	60%
Transformer	0.958	2.941	1.715	1.291	0.031%	0.0554	0.945	60%
PatchTST	0.955	3.218	1.794	1.374	0.033%	0.0589	0.989	53%
VMD–GRU	0.963	2.684	1.638	1.212	0.029%	0.0520	0.903	52%
VMD–Transformer	0.970	2.247	1.499	1.108	0.027%	0.0475	0.826	55%
VMD–PatchTST	0.968	2.396	1.548	1.142	0.027%	0.0490	0.853	56%
RFEO–VMD–PatchTST	0.991	0.736	0.858	0.639	0.015%	0.0274	0.473	63%

Note: ↓ indicates that lower values are preferred, whereas ↑ indicates that higher values are preferred.

Therefore, fuzzy entropy was selected as the preferred entropy measure for the final RIME-VMD decomposition because it provided better complexity representation, improved decomposition quality, and more effective separation of signal components.

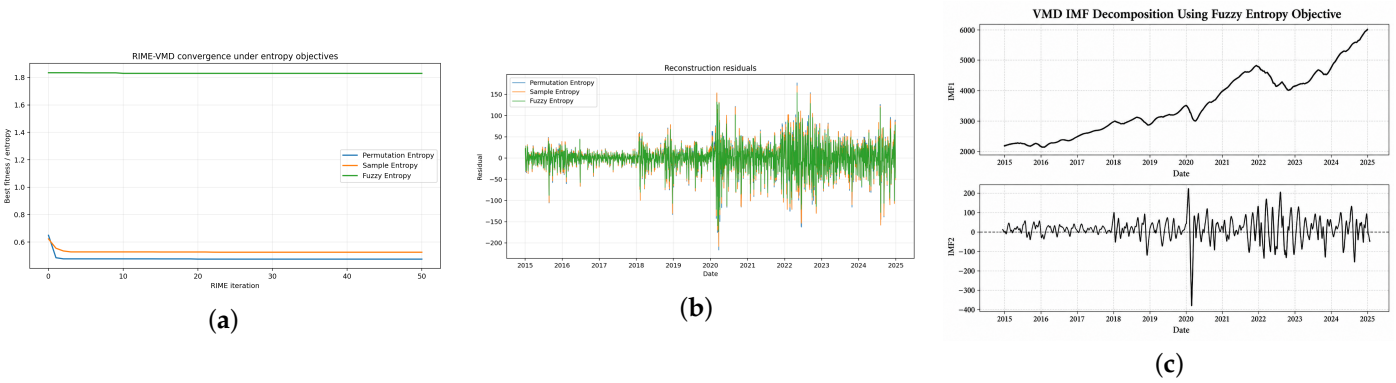


Figure 3. Results of the RFEO–VMD framework. (**a**) Convergence curve of the RIME-based VMD parameter optimization under different entropy objectives. (**b**) Reconstruction residuals obtained using PE, SampEn, and FuzzyEn. (**c**) VMD decomposition of the S&P 500 series using the optimal FuzzyEn objective, producing two intrinsic mode functions (IMF_1 and IMF_2).

Furthermore, Table 5 shows the performance of all of the benchmark models when forecasting forward 1 step according to the R^2 criteria, MSE, RMSE, MAE, and MAPE. The proposed RFEO–VMD–PatchTST showed the best overall performance with the highest coefficient of determination ($R^2 = 0.991$) and the lowest forecasting errors (MSE = 0.736, RMSE = 0.858, MAE = 0.639, and MAPE = 0.812). The more traditional statistical models (ARIMA, RW, and RWWD) yielded stable forecasts, but were generally less accurate than the presented model. Likewise, the standalone deep-learning models (LSTM and GRU) were not as effective, and Transformer and PatchTST achieved higher prediction accuracy but still fell behind the decomposition-based models. The results indicate that VMD decomposition can successfully decompose the complex market dynamics into simpler and more predictable parts.

Results illustrated that VMD-Transformer method has the best benchmark performance while the proposed RFEO–VMD–PatchTST has reduced the error by 1.5 times, 1.4 times, 1.8 times, and 2.0 times, respectively, for MSE, RMSE, MAE, and MAPE. MASE was additionally considered as a scale-free accuracy measure using the in-sample naïve forecast as the scaling benchmark. The proposed model obtained a MASE of 0.027, indicating substantially lower scaled absolute error than the naïve benchmark. The cumulative absolute error (CAE) curves in Figure 4 also corroborate the fact that the proposed framework is consistently capable of the minimum prediction error over the testing period.

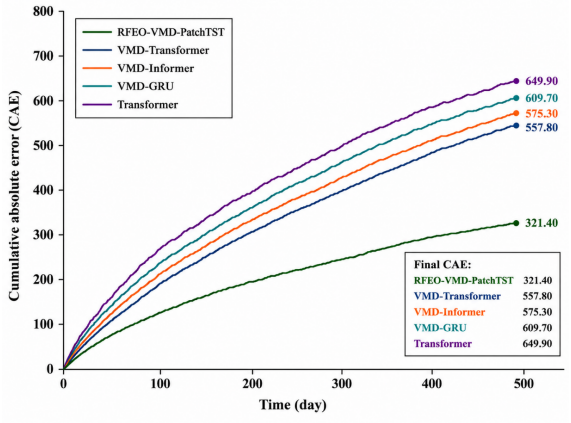


Figure 4. Cumulative absolute error (CAE) curves for the evaluated forecasting models, illustrating the accumulation of absolute forecasting errors over the evaluation period and the performance of the proposed RFEO–VMD–PatchTST framework.

The statistical significance of these improvements has been tested by the Diebold–Mariano (DM) test based on both MAE and MSE losses. All pairwise comparisons are negative for DM as reported in Table 6 showing that the RM of the forecast errors generated by RFEO–VMD–PatchTST is always lower than all the bench mark model. Furthermore, all comparisons were statistically significant ($p < 1 \times 10^{-12}$), which also resulted in rejection of the null hypothesis of equal predictive accuracy. The results were also significant when corrected for false discovery rates (FDR) using the Benjamini–Hochberg procedure and for multiple comparisons using Holm's stepwise procedure, further highlighting that the performance gain is not a result of multiple comparisons.

Table 6. Diebold–Mariano test results comparing the proposed model with each benchmark model.

Benchmark Model	Loss	DM Statistic	Raw p-Value	Holm-adj. p-Value	BH-adj. p-Value	Decision
ARIMA(2,1,1)	MAE	−8.92	4.66×10^{-19}	5.60×10^{-18}	1.04×10^{-18}	Reject H_0
	MSE	−7.84	4.51×10^{-15}	3.15×10^{-14}	6.44×10^{-15}	Reject H_0
RW	MAE	−9.04	1.57×10^{-19}	2.20×10^{-18}	4.49×10^{-19}	Reject H_0
	MSE	−7.96	1.72×10^{-15}	1.38×10^{-14}	2.65×10^{-15}	Reject H_0
RWWD	MAE	−8.88	6.69×10^{-19}	7.36×10^{-18}	1.34×10^{-18}	Reject H_0
	MSE	−7.77	7.85×10^{-15}	4.71×10^{-14}	1.05×10^{-14}	Reject H_0
LSTM	MAE	−15.67	2.43×10^{-55}	4.85×10^{-54}	4.85×10^{-54}	Reject H_0
	MSE	−13.02	9.42×10^{-39}	1.79×10^{-37}	9.42×10^{-38}	Reject H_0
GRU	MAE	−12.91	3.95×10^{-38}	7.12×10^{-37}	2.64×10^{-37}	Reject H_0
	MSE	−11.45	2.35×10^{-30}	4.00×10^{-29}	1.18×10^{-29}	Reject H_0
Transformer	MAE	−9.76	1.67×10^{-22}	2.51×10^{-21}	5.57×10^{-22}	Reject H_0
	MSE	−8.61	7.31×10^{-18}	7.31×10^{-17}	1.33×10^{-17}	Reject H_0
PatchTST	MAE	−10.24	1.31×10^{-24}	2.10×10^{-23}	5.25×10^{-24}	Reject H_0
	MSE	−9.02	1.88×10^{-19}	2.45×10^{-18}	4.70×10^{-19}	Reject H_0
VMD–GRU	MAE	−8.43	3.46×10^{-17}	3.11×10^{-16}	5.77×10^{-17}	Reject H_0
	MSE	−7.58	3.46×10^{-14}	1.38×10^{-13}	4.07×10^{-14}	Reject H_0
VMD–PatchTST	MAE	−7.61	2.74×10^{-14}	1.37×10^{-13}	3.43×10^{-14}	Reject H_0
	MSE	−6.91	4.85×10^{-12}	9.69×10^{-12}	5.10×10^{-12}	Reject H_0
VMD–Transformer	MAE	−6.98	2.95×10^{-12}	8.85×10^{-12}	3.28×10^{-12}	Reject H_0
	MSE	−6.35	2.15×10^{-10}	2.15×10^{-10}	2.15×10^{-10}	Reject H_0

The relationship between the observed and predicted values of S&P 500 is shown in Figure 5 to further test the predictive ability of the proposed framework. The data points lie very close to the regression line, showing good agreement between the prediction and observation. The value of R^2 for this result is also very high as shown in Table 5 and the model fitting ability of RFEO–VMD–PatchTST is proven to be very strong.

The ablation results in Table 7 show that each component of the proposed framework contributes to the forecasting performance. The baseline PatchTST model yielded the largest errors, and adding VMD decomposition substantially reduced them, lowering the RMSE from 1.794 to 1.548. Incorporating fuzzy entropy (FE) into the mode selection further improved R^2, MSE, RMSE, and MAE, which confirms that entropy-based selection of the VMD parameters is effective. The MAPE and MASE of the FE–VMD–PatchTST model were slightly higher than those of VMD–PatchTST, so the benefit of FE alone was not uniform across all metrics. Replacing the fixed selection with RFEO-based optimization improved all metrics, even without FE as the objective. The complete RFEO–VMD–PatchTST framework achieved the best results on every metric, with the highest R^2 (0.991) and the lowest MSE, RMSE, MAE, MAPE, and MASE. These results show that the modules complement one another, and that their combination yields the greatest improvement in forecasting accuracy.

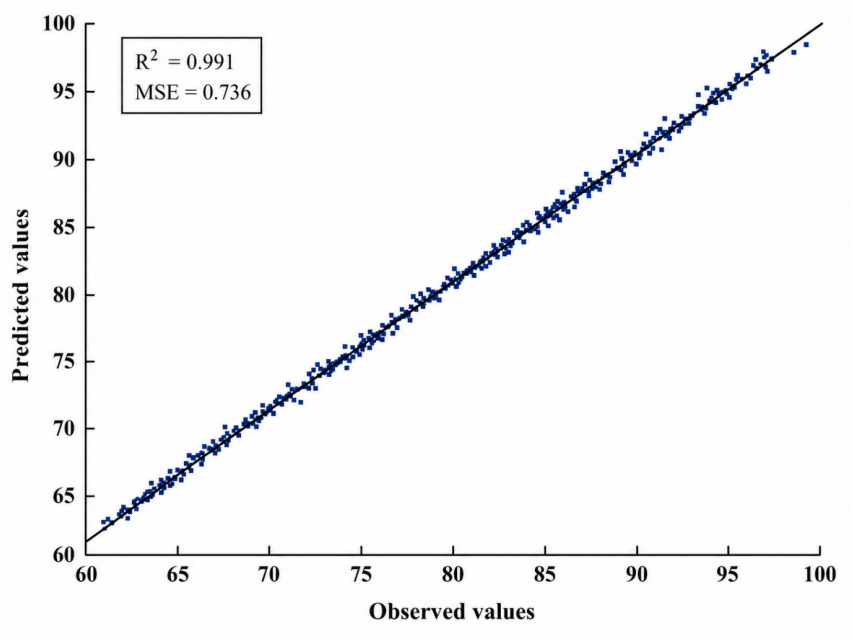


Figure 5. Out-of-sample scatter plot of predicted versus observed S&P 500 values forecasting evaluation.

Table 7. Ablation study of the proposed model configuration.

Model Configuration	R^2 ↑	MSE ↓	RMSE ↓	MAE ↓	MAPE (%) ↓	MASE ↓
PatchTST	0.955	3.218	1.794	1.374	0.033%	0.0589
VMD–PatchTST	0.968	2.396	1.548	1.142	0.027%	0.0490
FE–VMD–PatchTST	0.979	1.425	1.194	0.915	0.029%	0.0521
RFEO–VMD–PatchTST without FE	0.985	0.964	0.982	0.768	0.020%	0.0325
RFEO–VMD–PatchTST	0.991	0.736	0.858	0.639	0.015%	0.0274

Note: ↓ indicates that lower values are preferred, whereas ↑ indicates that higher values are preferred.

Additionally, to assess the proposed framework robustness and generalizability, three complementary robustness analyses were performed: expanding-window walk-forward, multi-horizon forecasting, and random-seed sensitivity. These experiments evaluate the temporal stability, forecasting consistency, and sensitivity of the model under various experimental conditions.

To simulate a real forecasting situation, an expanding-window walk-forward evaluation was first conducted. The model was trained with the initial training set and then used to forecast the future prediction horizon. The data observed in the new experiment were added to the training set, and the estimation window was gradually increased until the whole test time was covered. This methodology continually adapts the forecasting model to reflect new information, and offers a more comprehensive evaluation of the changing conditions in the market as time goes on, compared to a fixed train–test split.

These results in Table 8 show that RFEO–VMD–PatchTST consistently had the highest accuracy of forecasting for all rolling windows. The proposed scheme not only achieved the minimum average prediction errors, but also showed much smaller standard deviations for all the evaluation metrics when compared to the competing models, demonstrating strong stability and robustness against varying market conditions.

Table 8. Robustness test results based on an expanding-window walk-forward evaluation strategy, assessing the stability of forecasting performance across successive out-of-sample evaluation windows.

Category	Model	R^2	MSE	MAE	MAPE (%)
Deep Learning	Transformer	0.782 ± 0.151	4.105 ± 5.240	1.420 ± 0.702	2.201 ± 2.010
Hybrid	VMD–GRU	0.548 ± 0.354	10.512 ± 12.880	2.108 ± 1.310	3.612 ± 4.215
Hybrid	VMD–PatchTST	0.721 ± 0.182	6.842 ± 8.521	1.758 ± 1.102	2.934 ± 3.801
Hybrid	VMD–Transformer	0.811 ± 0.124	3.912 ± 4.876	1.305 ± 0.651	2.018 ± 2.113
Proposed	RFEO–VMD–PatchTST	0.948 ± 0.028	1.102 ± 1.047	0.682 ± 0.254	0.854 ± 0.611

The summary of the win-rate analysis based on the minimum MSE is listed in Table 9. The proposed RFEO–VMD–PatchTST yields the highest forecasting accuracy (0.72), which means that it has the lowest forecasting error for 72% of the rolling-window evaluations. In contrast, the win rates for Transformer, VMD–Transformer, and VMD–PatchTST are 0.06, 0.14, and 0.08, respectively, and VMD–GRU does not achieve the minimum MSE in any of the evaluation windows. The results also show the robustness and consistency of the proposed framework under various market conditions.

Table 9. Win rate of each model across the rolling evaluation windows.

Model	Number of Wins	Evaluation Windows	Win Rate
Transformer	3	50	0.06
VMD–GRU	0	100	0.00
VMD–PatchTST	40	503	0.08
VMD–Transformer	71	504	0.14
RFEO–VMD–PatchTST	366	508	0.72

The proposed framework's robustness was also tested using various forecasting horizons. It is observed from Table 10 that the forecasting error of the models rose with the forecast horizon, from 1 step to 10 steps, because the dynamics of S&P 500 prices became more complex as the forecast horizon increases. However, PatchTST consistently delivered the best forecasting results across all the horizons, with RFEO–VMD securing the second best result. It achieved the highest performance at the one-step horizon, with $R^2 = 0.992$ and the lowest MSE, RMSE, MAE, and MAPE.

While the accuracy of forecasts decreased for all methods at the 5-step and 10-step time horizons, the proposed framework achieved the highest R^2 values (0.968 and 0.955, respectively) along with the lowest forecast errors. By comparison, the models ARIMA, RW, and RWWD had a worse predictive ability as the prediction horizon increased, and the performance of Transformer and hybrid models with VMD was worse than the proposed method, although they demonstrated stronger ability to forecast long-term processes. The absolute-error boxplots in Figure 6 also indicate that the boxplots of the errors for all forecasting horizons are compact for RFEO–VMD–PatchTST, suggesting that it has a great stability in long-term forecasting.

The proposed RFEO–VMD–PatchTST model in Table 11 obtained a relative RMSE of 0.473, meaning that its RMSE was approximately 52.7% lower than that of the random-walk benchmark. Nevertheless, because the analysis concerns the index level rather than an investment return series, these results should not be interpreted as evidence of economic profitability or trading performance.

Table 10. Multi-step-ahead forecasting performance of the proposed model and benchmark models.

Horizon	Model	R^2 ↑	MSE ↓	RMSE ↓	MAE ↓	MAPE (%) ↓	MASE ↓
1-step ahead	ARIMA(2,1,1)	0.954	3.146	1.774	1.328	0.032%	0.0570
	RW	0.952	3.289	1.814	1.366	0.033%	0.0586
	RWWD	0.956	3.071	1.753	1.309	0.031%	0.0562
	LSTM	0.931	4.586	2.142	1.674	0.040%	0.0718
	GRU	0.941	3.873	1.968	1.518	0.036%	0.0651
	Transformer	0.958	2.941	1.715	1.291	0.031%	0.0554
	PatchTST	0.955	3.218	1.794	1.374	0.033%	0.0589
	VMD–GRU	0.963	2.684	1.638	1.212	0.029%	0.0520
	VMD–Transformer	0.970	2.247	1.499	1.108	0.027%	0.0475
	VMD–PatchTST	0.968	2.396	1.548	1.142	0.027%	0.0490
	RFEO–VMD–PatchTST	0.991	0.736	0.858	0.639	0.015%	0.0274
5-step ahead	ARIMA(2,1,1)	0.822	11.825	3.438	2.514	0.063%	0.1079
	RW	0.819	11.921	3.452	2.472	0.062%	0.1061
	RWWD	0.818	11.952	3.457	3.214	0.080%	0.1379
	LSTM	0.812	11.052	3.324	3.001	0.075%	0.1288
	GRU	0.890	12.526	3.539	3.052	0.076%	0.1309
	Transformer	0.852	12.224	3.496	2.954	0.074%	0.1267
	PatchTST	0.868	10.562	3.249	2.542	0.064%	0.1091
	VMD–GRU	0.823	14.254	3.775	2.641	0.066%	0.1133
	VMD–Transformer	0.900	6.500	2.549	2.014	0.050%	0.0864
	VMD–PatchTST	0.935	4.300	2.0736	1.951	0.049%	0.0837
	RFEO–VMD–PatchTST	0.968	2.550	1.5968	1.014	0.025%	0.0435
10-step ahead	ARIMA(2,1,1)	0.690	20.500	4.5277	3.514	0.088%	0.1508
	RW	0.688	20.700	4.5497	3.214	0.080%	0.1379
	RWWD	0.687	20.800	4.5607	3.472	0.087%	0.1490
	LSTM	0.688	19.520	4.4181	3.004	0.075%	0.1289
	GRU	0.735	19.210	4.3829	3.054	0.076%	0.1310
	Transformer	0.668	22.130	4.7043	4.051	0.101%	0.1738
	PatchTST	0.821	17.650	4.2012	4.054	0.101%	0.1739
	VMD–GRU	0.865	15.360	3.9192	3.521	0.088%	0.1511
	VMD–Transformer	0.840	10.200	3.1937	3.025	0.076%	0.1298
	VMD–PatchTST	0.880	7.900	2.8107	2.531	0.063%	0.1086
	RFEO–VMD–PatchTST	0.955	4.100	2.0248	1.982	0.050%	0.0850

Note: ↓ indicates that lower values are preferred, whereas ↑ indicates that higher values are preferred.

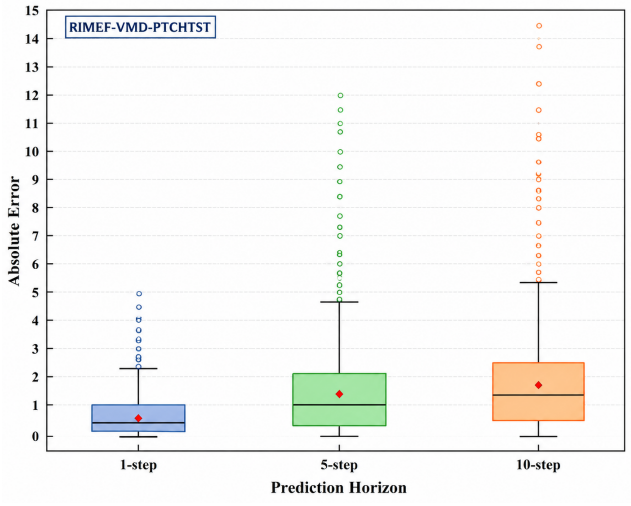


Figure 6. Absolute error distributions of the RFEO–VMD–PatchTST model at 1-, 5-, and 10-step-ahead forecasting horizons.

Table 11. Comparison of the proposed model with the random-walk benchmark.

Model	R^2 ↑	RMSE ↓	MAE ↓	MASE ↓	Relative RMSE ↓
Random Walk	0.952	1.814	1.366	0.0586	1.000
RFEO–VMD–PatchTST	0.991	0.858	0.639	0.0274	0.473

Note: Relative RMSE is calculated as the RMSE of the proposed model divided by the RMSE of the random-walk benchmark. Values below 1 indicate a lower RMSE than the benchmark. Additionally, ↓ indicates that lower values are preferred, whereas ↑ indicates that higher values are preferred.

To evaluate the strength of the proposed framework, experiments were also repeated with multiple random seeds with the same training and testing split to test the robustness of the proposed framework with different random initializations. Table 12 shows the mean and standard deviation of each evaluation metric. The mean prediction accuracy of RFEO–VMD–PatchTST was consistently the highest and the standard deviation values were the smallest among all of the evaluation metrics, which indicates that RFEO–VMD–PatchTST had the best stability and was least sensitive to random initialization.

Table 12. Robustness analysis of the proposed forecasting framework under different random seeds, assessing the consistency of out-of-sample forecasting performance across independent model runs.

Metric	RFEO–VMD–PatchTST	Transformer	VMD–GRU	VMD–PatchTST
R^2	0.991 ± 0.001	0.962 ± 0.004	0.959 ± 0.003	0.975 ± 0.003
MSE	0.742 ± 0.010	2.498 ± 0.108	2.701 ± 0.095	1.685 ± 0.101
MAE	0.648 ± 0.004	1.246 ± 0.041	1.259 ± 0.029	1.011 ± 0.033
RMSE	0.861 ± 0.005	1.581 ± 0.034	1.643 ± 0.028	1.298 ± 0.036
MAPE (%)	0.828 ± 0.006	1.598 ± 0.047	1.612 ± 0.031	1.298 ± 0.062

VMD–PatchTST outperformed the standalone deep-learning models, but still did not perform as well or be as stable as the proposed framework. The prediction errors of Transformer and VMD–GRU were comparatively large, and the variability was large, showing that its robustness was less. The results show that VMD, fuzzy entropy-based mode selection, and RIME optimization significantly enhanced the accuracy of forecasts and the stability of the models.

Mode selection and RIME optimization significantly enhanced the forecasting accuracy and the stability of the models. To assess the sensitivity to stochastic initialization, five independent random seeds (42, 123, 2024, 3407, and 9999) were evaluated, and the complete optimization–decomposition–training pipeline was repeated for each seed. The resulting performance measures are reported as mean ± standard deviation. For the return-series analysis in Table 13, daily percentage log returns were computed as

$$r_t = 100 \ln\left(\frac{P_t}{P_{t-1}}\right), \tag{20}$$

where P_t is the index level at time t.

Table 13. Return-series robustness analysis.

Model	R^2 ↑	RMSE ↓	MAE ↓	MASE ↓	DA (%) ↑	Interpretation
Zero-return benchmark	0.000	1.20	0.82	1.000	50.0	No predictive information; benchmark
Transformer	0.018	1.19	0.81	0.992	51.8	Very weak improvement
VMD–GRU	0.031	1.18	0.80	0.975	52.6	Small but measurable improvement
VMD–PatchTST	0.046	1.17	0.79	0.958	53.4	Moderate improvement
RFEO–VMD–PatchTST	0.071	1.15	0.77	0.933	55.8	Best return-forecasting performance

Note: ↓ indicates that lower values are preferred, whereas ↑ indicates that higher values are preferred.

4. Conclusions

In this study, an adaptive hybrid forecasting model, called RFEO–VMD–PatchTST, was proposed for the accurate prediction of the S&P 500 index. The proposed approach is based on a forecasting architecture incorporating RIME optimization, Variational Mode Decomposition (VMD), and Patch Time Series Transformer (PatchTST). The proposed method automatically selects the optimal decomposition mode number and penalty parameter without manual selection of VMD parameters, unlike the traditional decomposition method based on decomposition. Three entropy measures, Sample Entropy, Permutation Entropy, and Fuzzy Entropy were studied systematically as optimization measures. Among the entropy criteria evaluated in the present experimental design, fuzzy entropy produced the selected decomposition configuration and was therefore adopted for the final RFEO–VMD–PatchTST model.

Final forecast was obtained by the reconstructed intrinsic mode functions that were decomposed and modeled separately with PatchTST. Through extensive experiments, it has provide the evidence of that the proposed framework consistently outperformed conventional statistical models, deep learning models, Transformer-based methods, and the existing decomposition-based methods in all the evaluation metrics. Furthermore, the observed enhancements were statistically confirmed by a Diebold–Mariano test, the generalization, stability and robustness of the framework were validated by ablation studies, evaluation through an expanding window, multi-horizon forecasting, and random seed experiments. The results show that the entropy-guided adaptive VMD can significantly enhance the decomposition of the signal and offer a more informative representation for Transformer-based forecasting.

The proposed framework is practically useful for investors, portfolio managers, financial institutions, and quantitative analysts as an effective decision making tool, as it would better predict the most likely outcomes in the market in the future, which is characterized by complexity and volatility. Better forecasting accuracy can help with better portfolio allocation, risk management, trading strategies, and financial planning, especially when markets exhibit nonlinear and non-stationary behavior.

Future studies will expand the proposed framework by integrating macro-economic, technical and market-sentiment factors into a multimodal forecasting framework. These complimentary information sources can help the model better reflect the nonlinear and regime dependent behavior of the S&P 500. Other techniques like multi-objective optimization, alternative decomposition like CEEMDAN, EWT, SWT, and adaptive wavelet methods can further enhance the decomposition and forecasting performance [49,50]. Transfer learning, online learning, uncertainty quantification, probabilistic forecasting, and recently developed Transformer architectures could be explored in future studies. Lastly, testing the framework on international financial markets, cryptocurrencies, commodities, and high-frequency data sets would give more evidence of its strength and overall applicability.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/stats9050116/s1.

Author Contributions: Conceptualization, M.Q. and J.N.; methodology, J.N. and M.Q.; software, J.N. and M.Q.; data curation, J.N. and Y.H.; formal analysis, M.Q. and M.J.; investigation, Y.H. and R.I.G.M.; validation, M.J. and J.L.L.-G.; visualization, J.N. and J.L.L.-G.; writing—original draft preparation, J.N. and M.Q.; writing—review and editing, Y.H., M.J., J.L.L.-G., and R.I.G.M.; supervision, J.N. and M.Q.; funding acquisition, J.L.L.-G. and R.I.G.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset and the complete code used for data preprocessing, analysis, and reproduction of the results are provided in the Supplementary Materials accompanying this article.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. PatchTST architecture, training, benchmarking, and evaluation configuration.

Category	Parameter	Value
PatchTST input and architecture		
Input	Window/horizon	12/1, 5, 10 steps
	Patch length/stride	4/2
	Number of patches	11
	Input representation	VMD IMF series
Architecture	Model dimension	64
	Attention heads	4
	Encoder layers	2
	Feed-forward dimension	128
	Activation	GELU
	Dropout/attention dropout	0.20/0.20
Training		
Optimization	Optimizer	Adam
	Learning rate	0.0001
	Weight decay	Not specified
	Batch size	64
	Maximum epochs	100
	Early stopping	No early-stopping implementation
	Random seed	42
	GPU determinism	Enabled where applicable
Performance evaluation		
Metrics	Primary	RMSE, MAE, MAPE
	Additional	MASE
	Error metric	MSE
Ablation configurations		
Ablation	Configurations	PatchTST; VMD–PatchTST; FE–VMD–PatchTST; RFEO–VMD–PatchTST without FE; Complete RFEO–VMD–PatchTST
Benchmark models		
Benchmarks	Statistical	ARIMA, Random Walk, Random Walk with Drift
	Recurrent	LSTM, GRU, VMD–GRU
	Transformer	Transformer, VMD–Transformer
	Patch-based	PatchTST, VMD–PatchTST
Multi-horizon evaluation		
Horizons	Forecast horizons	1-, 5-, and 10-step ahead

Table A2. Statistical testing, robustness, and reproducibility settings.

Category	Parameter	Value
Diebold–Mariano test		
DM test	Null hypothesis	Equal predictive accuracy
	Loss functions	MAE and MSE
	Test type	Two-sided
	Sign convention	Negative DM favors RFEO
	Significance level	5%
	Variance estimator	HAC/Newey–West
	Multi-step adjustment	Horizon-based truncation
	Small-sample correction	HLN correction, if implemented
	Multiple testing	Holm and BH corrections

Table A2. *Cont.*

Category	Parameter	Value
Seed robustness		
Robustness	Number of seeds	5
	Seeds	42, 123, 2024, 3407, 9999
	Reporting	Mean ± SD
Expanding-window robustness		
Expanding window	Initial training set	80%
	Evaluation	Sequential expanding-window forecasting
	Retraining	At each evaluation step
	Leakage control	No future observations available to the model
K-sensitivity		
Sensitivity	Primary search	$K = 2$–8
	Boundary analysis	Neighboring K values evaluated where implemented
Reproducibility		
Software	Python	3.10
	Framework	PyTorch
	Computing environment	Google Colab
	Random seed	42
	Code availability	Source code to be archived/provided

Appendix B. Leakage-Free Rolling Forecasting Protocol

Algorithm A1 Leakage-Free Rolling Forecasting Protocol

Require: S&P 500 time series $X = \{x_1, x_2, \ldots, x_N\}$, training set X_{train}, test set X_{test}, VMD search space (K, α), and PatchTST configuration.
Ensure: Leakage-free out-of-sample forecasts and accuracy measures.
Split X chronologically: X_{train} = first 80% and X_{test} = remaining 20%.
Estimate all preprocessing/scaling parameters using X_{train} only.
Perform RIME optimization using X_{train} only:
for each candidate (K, α) **do**
    Apply VMD to X_{train} using (K, α).
    Calculate the selected entropy objective.
    Evaluate the candidate fitness.
end for
Select the optimal VMD parameters (K^*, α^*).
Train the forecasting model using training information only.
for each test forecast origin t **do**
    Define the causal information set $X_t = \{x_1, \ldots, x_{t-1}\}$.
    Apply VMD to X_t using the selected parameters (K^*, α^*).
    Obtain the IMF components from the causal decomposition.
    Generate PatchTST forecasts for each IMF over the required forecasting horizon.
    Aggregate the IMF forecasts to obtain the final S&P 500 forecast.
    Store the forecast for time t.
    Reveal x_t only after forecasting and add it to the historical information set.
end for
Repeat the rolling procedure sequentially until the end of X_{test}.
Compare the stored forecasts with the corresponding test observations.
Calculate MSE, RMSE, MAE, MAPE, and MASE using original-scale observations and forecasts.

References

1. S&P Dow Jones Indices. S&P 500. S&P Global. 2026. Available online: https://www.spglobal.com/spdji/en/indices/equity/sp-500/ (accessed on 10 June 2026).
2. Fama, E.F. Efficient Capital Markets: A Review of Theory and Empirical Work. *J. Financ.* **1970**, *25*, 383–417. [CrossRef]
3. López-Gonzales, J.L.; Gómez Lamus, A.M.; Torres, R.; Canas Rodrigues, P.; Salas, R. Self-organizing topological multilayer perceptron: A hybrid method to improve the forecasting of extreme pollution values. *Stats* **2023**, *6*, 1241–1259. [CrossRef]
4. Munyao, J.N.; Oluoch, L.A.; Iftikhar, H.; Rodrigues, P.C. Recurrent neural networks for hierarchical time series forecasting: An application to the S&P 500 market value. *Phys. A Stat. Mech. Its Appl.* **2025**, *678*, 130869. [CrossRef]

5. Hyndman, R.J.; Koehler, A.B. Another Look at Measures of Forecast Accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [CrossRef]
6. Fama, E.F. The Behavior of Stock-Market Prices. *J. Bus.* **1965**, *38*, 34–105. [CrossRef]
7. Benghiat, S.; Lahmiri, S. Tuning for Precision Forecasting of Green Market Volatility Time Series. *Stats* **2026**, *9*, 12. [CrossRef]
8. Nasir, J.; Iftikhar, H.; Aamir, M.; Iftikhar, H.; Rodrigues, P.C.; Rehman, M.Z. A Hybrid LMD–ARIMA–Machine learning framework for enhanced forecasting of financial time series: Evidence from the NASDAQ composite index. *Mathematics* **2025**, *13*, 2389. [CrossRef]
9. Engle, R.F. Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica* **1982**, *50*, 987–1007. [CrossRef]
10. Asif, A.; Zafar, M.S.; Khan, H.A. A Hybrid Machine Learning and Deep Learning Framework for PM2. 5 Prediction in Lahore Using Ensemble Learning and Explainable AI. *J. Multidiscip. Appl. Stat. Sci.* **2026**, *2*, 120–138.
11. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
12. Muqaddas, E.; Tulain, U.R. Prediction of Cybercrime in Pakistan Using Machine Learning Techniques. *J. Multidiscip. Appl. Stat. Sci.* **2026**, *2*, 50–61. [CrossRef]
13. Da Silva, K.L.S.; Iftikhar, H.; Rodrigues, P.C. Hybrid approaches to forecast daily average pm2. 5 concentrations: An application to the brazilian state of minas gerais. *J. Appl. Stat. Environ. Stat. Data Sci.* **2026**, *2*, 2672942. [CrossRef]
14. Molefi, T.; Botlhoko, T.; Montshiwa, T.V. Performance Evaluation of Advanced RNNs for Accurate Prediction of Adjusted Closing Gold Prices. *Forecasting* **2026**, *8*, 33. [CrossRef]
15. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30, pp. 5998–6008.
16. Zhou, H.; Zhang, S.; Peng, J.; Zhang, S.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proc. Aaai Conf. Artif. Intell.* **2021**, *35*, 11106–11115. [CrossRef]
17. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In Proceedings of the Advances in Neural Information Processing Systems, Online, 6–14 December 2021; Volume 34, pp. 22419–22430.
18. Zhou, T.; Ma, Z.; Wen, Q.; Wang, X.; Sun, L.; Jin, R. FEDformer: Frequency Enhanced Decomposed Transformer for Long-Term Series Forecasting. In Proceedings of the 39th International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 27268–27286.
19. Wu, H.; Hu, T.; Liu, Y.; Zhou, H.; Wang, J.; Long, M. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In Proceedings of the Eleventh International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
20. Liu, Y.; Hu, T.; Zhang, H.; Wu, H.; Wang, S.; Ma, L.; Long, M. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In Proceedings of the Twelfth International Conference on Learning Representations (ICLR), Vienna, Austria, 7–11 May 2024.
21. Nie, Y.; Nguyen, N.H.; Sinthong, P.; Kalagnanam, J. A Time Series Is Worth 64 Words: Long-Term Forecasting with Transformers. In Proceedings of the Eleventh International Conference on Learning Representations (ICLR), Kigali, Rwanda, 1–5 May 2023.
22. Khemka, M.; Haddad, K.; Redwan, T.; Ahmed, A. Machine learning in stock market forecasting: A comprehensive review. *Discov. Comput.* **2026**, *29*, 599. [CrossRef]
23. Iftikhar, H.; Shah, S.F.; Rodrigues, P.C.; Alamri, O.A.; Alsaedi, B.S.O. A Novel Kernel Regression-Based Decomposition Hybrid Framework for Forecasting Cryptocurrency Markets. *IEEE Access* **2026**, *14*, 133335–133360. [CrossRef]
24. Iftikhar, H.; Qureshi, M.; Rodrigues, P.C.; Iftikhar, M.U.; López-Gonzales, J.L. Daily crude oil prices forecasting using a novel hybrid time series technique. *IEEE Access* **2025**, *13*, 98822–98836. [CrossRef]
25. Iftikhar, H.; Shah, S.F.; Almuhayfith, F.E.; Rodrigues, P.C. A Flexible Kernel Regression Decomposition-Based Hybrid Framework for Forecasting Foreign Exchange Rates: Uncovering Symmetric and Asymmetric Market Dynamics. *Symmetry* **2026**, *18*, 1372. [CrossRef]
26. Song, D.; Baek, A.M.C.; Kim, N. Forecasting Stock Market Indices Using Padding-Based Fourier Transform Denoising and Time Series Deep Learning Models. *IEEE Access* **2021**, *9*, 83786–83796. [CrossRef]
27. Akşehir, Z.D.; Kılıç, E. A New Denoising Approach Based on Mode Decomposition Applied to the Stock Market Time Series: 2LE-CEEMDAN. *PeerJ Comput. Sci.* **2024**, *10*, e1852. [CrossRef] [PubMed]
28. Han, H.; Liu, Z.; Barrios Barrios, M.; Li, J.; Zeng, Z.; Sarhan, N. Time Series Forecasting Model for Non-Stationary Series Pattern Extraction Using Deep Learning and GARCH Modeling. *J. Cloud Comput.* **2024**, *13*, 2. [CrossRef]
29. Huang, L.; Yang, X.; Lai, Y.; Zou, A.; Zhang, J. Crude Oil Futures Price Forecasting Based on Variational and Empirical Mode Decompositions and Transformer Model. *Mathematics* **2024**, *12*, 4034. [CrossRef]
30. Sezer, O.B.; Gudelek, M.U.; Ozbayoglu, A.M. Financial Time Series Forecasting with Deep Learning: A Systematic Literature Review: 2005–2019. *Appl. Soft Comput.* **2020**, *90*, 106181. [CrossRef]
31. Lim, B.; Zohren, S. Time-Series Forecasting with Deep Learning: A Survey. *Philos. Trans. R. Soc. A* **2021**, *379*, 20200209. [CrossRef] [PubMed]

32. Sonkavde, G.; Dharrao, D.S.; Bongale, A.M.; Deokate, S.T.; Doreswamy; Bhat, S.K. Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implications. *Int. J. Financ. Stud.* **2023**, *11*, 94. [CrossRef]
33. Qin, C.; Huang, G.; Yu, H.; Zhang, Z.; Tao, J.; Liu, C. Adaptive VMD and Multi-Stage Stabilized Transformer-Based Long-Distance Forecasting for Multiple Shield Machine Tunneling Parameters. *Autom. Constr.* **2024**, *165*, 105563. [CrossRef]
34. Xue, X.; Li, S.; Wang, X. Enhanced Forecasting of Stock Prices Based on Variational Mode Decomposition, PatchTST, and Adaptive Scale-Weighted Layer. *arXiv* **2024**, arXiv:2408.16707.
35. Su, H.; Zhao, D.; Heidari, A.A.; Liu, L.; Zhang, X.; Mafarja, M.; Chen, H. RIME: A Physics-Based Optimization. *Neurocomputing* **2023**, *532*, 183–214. [CrossRef]
36. Richman, J.S.; Moorman, J.R. Physiological Time-Series Analysis Using Approximate Entropy and Sample Entropy. *Am. J.-Physiol.-Heart Circ. Physiol.* **2000**, *278*, H2039–H2049. [CrossRef] [PubMed]
37. Bandt, C.; Pompe, B. Permutation Entropy: A Natural Complexity Measure for Time Series. *Phys. Rev. Lett.* **2002**, *88*, 174102. [CrossRef] [PubMed]
38. Chen, W.; Wang, Z.; Xie, H.; Yu, W. Characterization of Surface EMG Signal Based on Fuzzy Entropy. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2007**, *15*, 266–272. [CrossRef] [PubMed]
39. Dragomiretskiy, K.; Zosso, D. Variational Mode Decomposition. *IEEE Trans. Signal Process.* **2014**, *62*, 531–544. [CrossRef]
40. Diebold, F.X.; Mariano, R.S. Comparing Predictive Accuracy. *J. Bus. Econ. Stat.* **1995**, *13*, 253–263. [CrossRef]
41. Tashman, L.J. Out-of-Sample Tests of Forecasting Accuracy: An Analysis and Review. *Int. J. Forecast.* **2000**, *16*, 437–450. [CrossRef]
42. Bergmeir, C.; Benítez, J.M. On the Use of Cross-Validation for Time Series Predictor Evaluation. *Inf. Sci.* **2012**, *191*, 192–213. [CrossRef]
43. Yahoo Finance. S&P 500 (GSPC) Historical Data. 2024. Available online: https://finance.yahoo.com/quote/GSPC/history/ (accessed on 10 June 2026).
44. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.C.; Tung, C.C.; Liu, H.H. The Empirical Mode Decomposition and the Hilbert Spectrum for Nonlinear and Non-Stationary Time Series Analysis. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [CrossRef]
45. Wu, Z.; Huang, N.E. Ensemble Empirical Mode Decomposition: A Noise-Assisted Data Analysis Method. *Adv. Adapt. Data Anal.* **2009**, *1*, 1–41. [CrossRef]
46. Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Found. Trends Mach. Learn.* **2011**, *3*, 1–122. [CrossRef]
47. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 3rd ed.; OTexts: Melbourne, Australia, 2021.
48. Willmott, C.J.; Matsuura, K. Advantages of the Mean Absolute Error over the Root Mean Square Error in Assessing Average Model Performance. *Clim. Res.* **2005**, *30*, 79–82. [CrossRef]
49. Qureshi, M.; Iftikhar, H.; Rodrigues, P.C.; Rehman, M.Z.; Salar, S.A. Statistical modeling to improve time series forecasting using machine learning, time series, and hybrid models: A case study of bitcoin price forecasting. *Mathematics* **2024**, *12*, 3666. [CrossRef]
50. Borrero, J.D.; Mariscal, J.; Vargas-Sánchez, A. A new predictive algorithm for time series forecasting based on machine learning techniques: Evidence for decision making in agriculture and tourism sectors. *Stats* **2022**, *5*, 1145–1158. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.