Coffee Price Forecasting

ADARSH S KAMATAGI
Indian Institute of Information Technology, Dharwad
23bcs005@iiitdwd.ac.in

BABHALE ROHAN LAXMIKANT
Indian Institute of Information Technology, Dharwad
23bcs026@iiitdwd.ac.in

Abstract

Coffee, the second most traded soft commodity, shows high price volatility driven by climate, demand, and market dynamics. Accurate forecasting is essential for informed trading and risk management. This study develops a hybrid decomposition-based LSTM framework for forecasting coffee futures prices. Using historical data from Yahoo Finance, the series was found to be non-stationary with a strong upward trend and multiple long cycles. MSTL decomposition separated the series into trend, three cyclical components (143, 687, 3200 days), and residuals. Each component was modeled separately using Polynomial Regression for trend, Naive–LSTM ensembles for short and medium cycles, and deep LSTMs for long cycles and residuals. The recombined forecast achieved RMSE = 10.54, MAE = 7.90, and MAPE = 2.89%, effectively capturing both trend and fluctuations. The model further enabled actionable insights for traders, retailers, and producers across short, medium, and long term horizons.

Introduction

Coffee prices exhibit significant volatility, driven by a complex interplay of economic, environmental, and policy factors. Accurate prediction is crucial for stakeholders across the coffee value chain, requiring analysis not only of historical price trends but also of the external drivers influencing supply and demand dynamics. Modeling these dynamics allows for better anticipation of price fluctuations, thereby guiding informed decision-making in the volatile coffee market.

Time-series forecasting methods are particularly relevant for this task as they can capture underlying patterns such as long-term trends and seasonal cycles inherent in commodity prices. However, the highly nonlinear and often non-stationary nature of coffee price data poses challenges for traditional linear econometric models. In recent years, machine learning (ML) and deep learning (DL) models have gained prominence due to their ability to capture complex, non-linear relationships within the data. These advanced models can uncover intricate patterns often missed by conventional methods, leading to potentially more accurate predictions in the context of financial and agricultural time series.

Table I lists several factors identified in the literature that can influence coffee prices, highlighting the multifaceted nature of the forecasting problem.

I. Motivation

The primary motivation for our hybrid model stems from the complex nature of the coffee price time series, which our analysis shows is a composite of a strong, non-linear trend and multiple, overlapping long-period cycles. Applying a single, monolithic model like a standard LSTM to this raw data is a common but the model often “smooths out” the forecast, capturing the general direction but failing to reproduce the critical cyclical fluctuations that define the market’s volatility. Our divide and conquer strategy addresses this by first using MSTL to decompose the series. This allows us to isolate the dominant trend and model it with a stable Polynomial Regression, while dedicating specialized LSTM and Naive ensemble models to learning the distinct patterns of each cycle. We hypothesize that by forecasting these simpler components individually and then recombining them, the final forecast will be far more realistic, successfully capturing the sharp, periodic fluctuations that a single model would otherwise miss.

Literature Review

The forecasting of agricultural commodity prices, particularly coffee, involves extreme volatility, non-linearity, and susceptibility to exogenous shocks ranging from microclimatic shifts to global trade policies. Recent literature has demonstrated a clear methodological shift from traditional econometric approaches toward sophisticated Deep Learning (DL) and hybrid architectures to address these complexities. A comprehensive summary of the reviewed literature, highlighting methodologies and identified research gaps, is presented in Table II.

I. The Limitations of Traditional and Statistical Approaches

Historically, statistical models such as Autoregressive Integrated Moving Average (ARIMA) and Linear Regression were the standard for price forecasting. However, their reliance on assumptions of linearity and stationarity renders them largely ineffective for coffee price data,

TABLE I. Factors Potentially Affecting Coffee Price

Environmental	**Market/Economic**	**Structural/Policy**
1. Weather / Climate	2. Seasonality / Cycles	3. Pests & Diseases
4. Agronomic Practices	5. Currency / Exchange Rates	6. Global Demand / Supply
7. Substitutes	8. Speculation & Futures Trading	9. Market Policies
10. Logistics / Supply Chain	11. Farmer Livelihood Assets	12. Labor Laws

which is inherently non-stationary and chaotic. Hwase and Fofanah [1] compared Linear Regression against Machine Learning (ML) techniques like Extreme Gradient Boosting (XGBoost) and Long Short-Term Memory (LSTM) networks for the Ethiopian coffee market. Their findings indicated that while regression provides a baseline, it fails to capture drastic price changes, whereas XGBoost and LSTM achieved significantly lower error rates by learning non-linear patterns.

Similarly, Deina et al. [2] proposed the use of Extreme Learning Machines (ELM) as a computationally efficient alternative to traditional feed-forward networks. By preprocessing data with differencing and Partial Autocorrelation Function (PACF) filters, they demonstrated that ELM outperformed ARIMA and Exponential Smoothing in 71% of tested scenarios across Arabica and Robusta markets. Despite these improvements, standard ML models often struggle to capture the long-term temporal dependencies required for accurate multi-step forecasting.

II. *The Shift to Deep Learning and Sequence Modeling*

To address the temporal limitations of standard ML, recent research has gravitated toward sequence-based Deep Learning models. Manogna et al. [3] conducted an extensive empirical analysis of 23 agricultural commodities in India, comparing stochastic models against DL architectures including RNNs, LSTMs, and Gated Recurrent Units (GRUs). Their results conclusively showed that LSTM and GRU architectures provide superior accuracy (lowest RMSE and MAPE) by effectively utilizing gating mechanisms to retain relevant historical information over long sequences, a capability lacking in models like Support Vector Regression (SVR).

However, the implementation of these models faces infrastructural and data challenges. Le et al. [4] emphasized the necessity of robust Data Warehousing and ETL (Extract, Transform, Load) processes to manage the high volume of unstructured trading data. They highlight that without a centralized repository integrating historical prices with external factors, even advanced algorithms like Facebook Prophet yield suboptimal results for informed decision-making.

III. *Addressing Volatility: Hybrid and Multimodal Architectures*

A critical gap identified in the literature is that while standard LSTMs capture general trends, they often ”smooth out” the extreme volatility characteristic of coffee markets. To mitigate this, recent studies in 2025 have focused on hybridizing models to treat volatility and trends separately.

Manogna et al. [7] proposed a hybrid neural network approach combining LSTM with Generalized Autoregressive Conditional Heteroskedasticity (GARCH). In this framework, the LSTM models the non-linear trend, while the GARCH component explicitly models the volatility of the residuals. Parallel to this, researchers are exploring high-complexity architectures. Nayak et al. [11] introduced a ”Meta-transformer” framework, utilizing metaheuristic algorithms to automatically tune Transformer hyperparameters. Other approaches include converting time-series data into images for CNNs as explored by Zhang et al. [9], and Multimodal GANs as proposed by Kambo et al. [10].

IV. *Research Gap and Project Motivation*

While the current state-of-the-art leans toward complex hybrids, there is limited research on ”Divide and Conquer” strategies that explicitly decompose the time series into interpretable components (Trend, Seasonality, Residuals) before modeling. Most existing hybrids model the series as a whole or focus only on residual volatility. This project aims to implement a component-wise hybrid model (MSTL-LSTM), leveraging decomposition to allow specific models to specialize in different aspects of the coffee price signal.

Time Series Analysis

Time series analysis is an important preliminary phase in forecasting that emphasizes the study of the temporal behavior and statistical features of successive observations. This pre-forecasting analysis not only helps in the selection of the most suitable modeling methods but also improves the final forecasts' accuracy, interpretability, and robustness. For this project all the tests were performed using python libraries available.

TABLE II. Comprehensive Literature Survey Mapping: Methodologies, Findings, and Research Gaps

Author (Year)	**Methodology**	**Key Contribution / Findings**	**Identified Gap / Limitation**
Le et al. [4]	Big Data & AI Analytics	Emphasized the role of Data Warehousing and ETL for informed coffee trading.	Focuses on infrastructure and data management rather than specific forecasting algorithms.
Kishaija et al. [5]	Systematic Review	Analyzed impact of climate change and livelihood assets on coffee farming.	Qualitative analysis of causal factors; does not provide a quantitative forecasting model.
Hwase & Fofanah [1]	Linear Regression vs. XGBoost & LSTM	XGBoost and LSTM significantly outperformed Linear Regression in capturing non-linear spikes.	Standard LSTM struggles with extreme volatility clusters without decomposition.
Deina et al. [2]	Extreme Learning Machines (ELM)	ELM outperformed ARIMA in 71% of scenarios; highly computationally efficient.	Limited ability to capture very long-term temporal dependencies compared to Deep Learning.
Pham et al. [6]	Facebook Prophet	Applied additive regression models to capture seasonality in coffee prices.	Prophet often fails to capture complex, non-linear dependencies in highly volatile markets.
Manogna et al. [3]	Deep Learning (RNN, LSTM, GRU)	LSTM and GRU achieved lowest RMSE across 23 commodities due to gating mechanisms.	”Black-box” nature; requires large datasets to generalize effectively.
Manogna et al. [7]	Hybrid LSTM-GARCH	LSTM models the trend, GARCH explicitly models residual volatility.	Handles volatility well but treats the series as a whole without decomposing cycles.
M. K. L. V et al. [8]	CNN-BLSTM	Used CNN for feature extraction and BLSTM for sequence learning.	High model complexity; risk of overfitting on smaller commodity datasets.
Zhang et al. [9]	Image Encoding + Attention	Converted time-series to images to leverage Computer Vision (CNN + Attention).	Image conversion can lead to loss of granular temporal resolution. Increased computation.
Kambo et al. [10]	GANs & Autoencoders	Used Generative Adversarial Networks to generate synthetic features for training.	Generative models are notoriously unstable to train (mode collapse) and computationally expensive.
Nayak et al. [11]	Meta-Transformer (GWO/WOA)	Used metaheuristic algorithms to optimize Transformer hyperparameters.	Extreme computational overhead; difficult to implement for real-time trading systems.
Current Work	**MSTL + Hybrid Ensembles**	**Decomposes series into Trend, Cycles, and Residuals for specialized modeling.**	-

TABLE III. Summary Statistics of Coffee Futures Data (6471 entries) from Yahoo finance (ticker: KC=F)

Statistic	**Close**	**High**	**Low**	**Open**	**Volume**
Count	6471	6471	6471	6471	6471
Mean	140.97	142.77	139.34	141.06	9415.22
Std	67.11	68.02	66.22	67.05	10087.29
Min	41.50	42.10	41.50	41.50	0
25%	102.50	103.75	101.38	102.50	58
50%	125.10	126.50	123.60	125.25	7513
75%	170.65	173.45	168.10	170.97	15577.50
Max	438.90	440.85	424.05	435.40	62750

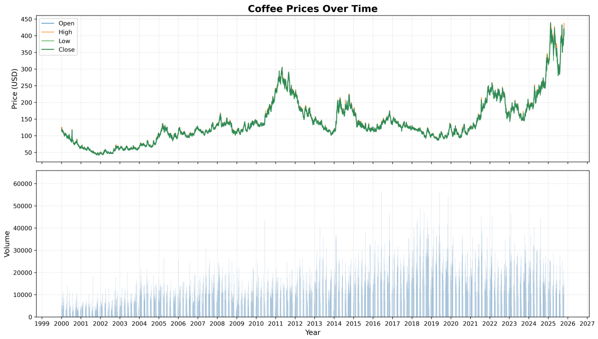

FIG. 1. FEATURES OF COFFEE FUTURES VS TIME (TOP), VOLUME TRADED VS TIME (BOTTOM)

I. *Visualization*

The closing price reflects the final consensus value of a trading day, capturing market movements and investor sentiment. It is a widely used indicator for analysis and forecasting in financial time series. The historical data is visualized in Figure 1.

II. *Stationarity*

A stationary time series exhibits constant mean, variance, and autocorrelation over time, a crucial property for reliable forecasting. To rigorously assess this, we employed two complementary statistical tests.

1) *Augmented Dickey-Fuller (ADF) Test*

The ADF test checks for the presence of a unit root, which indicates non-stationarity. The null hypothesis (H_0) is that a unit root exists ($\gamma = 0$). The test is based on the regression:

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^{p} \delta_i \Delta y_{t-i} + \epsilon_t \qquad (1)$$

where Δ is the difference operator, α is a constant, β is the trend coefficient, and p is the lag order chosen to minimize the Akaike Information Criterion (AIC). A highly negative test statistic (p-value < 0.05) rejects H_0, implying stationarity.

2) Kwiatkowski-Phillips-Schmidt-Shin (KPSS) Test

Unlike ADF, the KPSS test checks for stationarity around a deterministic trend. The null hypothesis (H_0) is that the series is stationary. The series is decomposed as:

$$y_t = \xi t + r_t + \epsilon_t \tag{2}$$

where r_t is a random walk ($r_t = r_{t-1} + u_t$). The test statistic assesses whether the variance of the random walk component is zero. If the statistic exceeds the critical value (p-value < 0.05), we reject stationarity.

TABLE IV. Stationarity Test Results for Closing Price

Test	**ADF**	**KPSS**
Test Statistic	0.568	0.722
p-value	0.987	<0.01
Critical Values		
ADF 1%	-3.431	
ADF 5%	-2.862	
ADF 10%	-2.567	
KPSS 1%		0.739
KPSS 5%		0.463
KPSS 10%		0.347

The ADF test fails to reject the null hypothesis ($p > 0.05$), suggesting the series has a unit root. Conversely, the KPSS test rejects the null ($p < 0.05$), confirming non-stationarity. Both tests suggest the original closing price series is non-stationary.

III. Trend Seasonality Analysis

Identifying underlying trends (long-term movements) and seasonality (regular, periodic fluctuations) is essential for understanding time series behavior.

1) Trend Detection (Mann-Kendall Test)

The Mann-Kendall (M-K) test is a non-parametric method used to detect monotonic trends without assuming a linear distribution. The test statistic S is calculated as:

$$S = \sum_{k=1}^{n-1} \sum_{j=k+1}^{n} \operatorname{sgn}(x_j - x_k) \tag{3}$$

where $\operatorname{sgn}(x)$ is the sign function, returning 1 if $x > 0$, 0 if $x = 0$, and -1 if $x < 0$. For $n > 10$, S follows a normal distribution with mean zero and variance $Var(S)$. The standardized test statistic Z_{MK} determines significance:

$$Z_{MK} = \begin{cases} \frac{S-1}{\sqrt{Var(S)}} & \text{if } S > 0 \\ 0 & \text{if } S = 0 \\ \frac{S+1}{\sqrt{Var(S)}} & \text{if } S < 0 \end{cases} \tag{4}$$

A positive Z_{MK} indicates an increasing trend.

The Mann-Kendall test result suggests a statistically significant monotonic (likely increasing) trend in the data.

TABLE V. Mann-Kendall Trend Test Result

Test	**Mann-Kendall**
Normalized Stat (Z) Kendall's tau p-value	High positive value $\approx$ 0.51 <0.001
Interpretation	Significant Increasing Trend

2) Seasonality Assessment

Seasonality refers to predictable patterns recurring at fixed intervals. Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots help visualize serial correlation and potential seasonality.

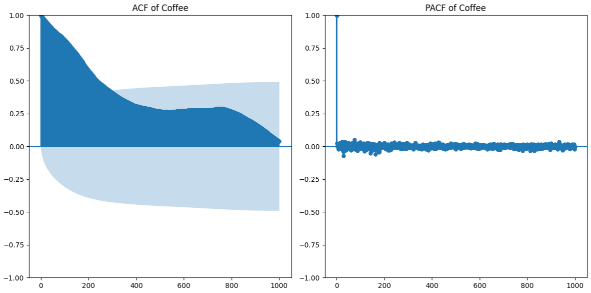

FIG. 2. ACF AND PACF PLOT FOR ORIGINAL CLOSING PRICE DATA

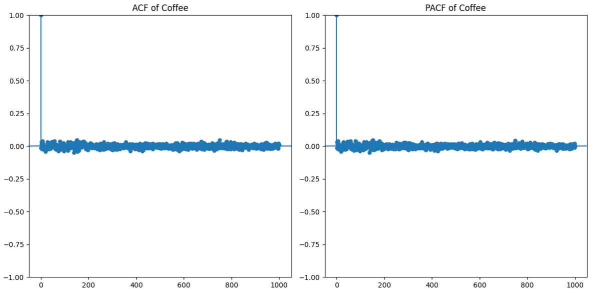

FIG. 3. ACF AND PACF PLOT FOR FIRST-ORDER DIFFERENCED DATA

The ACF plot for the original data (Figure 2) shows slow decay, characteristic of non-stationary series with a strong trend. After first-order differencing, the ACF decays faster, and the PACF shows significant spikes, suggesting ARMA behavior in the differenced series. No clear, strong calendar seasonality is apparent from the plots, though longer-term cycles might exist.

We tried to take a brute force approach as well to find the ideal time period for STL decomposition using a Python library. In this process, we evaluated the Ljung-Box test, seasonal strength, and trend strength for the original data, first-order differencing data, and second-order differenced data. The results indicated that the original series exhibits strong trend behavior with very weak seasonality, while differencing slightly reduced autocorrelation but did not reveal any significant seasonal patterns. The Ljung-Box test confirmed persistent autocorrelation across all forms of the series, and the seasonal strength remained close to zero, reinforcing the conclusion that cyclical or seasonal effects are minimal.

Analysis using libraries like `cydets` [12] identified potential cyclical periods around [143, 687, 3200] days emperically, suggesting complex cyclical patterns rather than simple seasonality.

3) Time Series Decomposition

Initially, standard STL decomposition was considered, but given the complex cyclical patterns identified, we moved to Multi-Seasonal Time Series Decomposition using Loess (MSTL) using the identified periods [143, 687, 3200]. The additive model is assumed:

$$Y_t = T_t + S_{t,143} + S_{t,687} + S_{t,3200} + R_t \tag{5}$$

where Y_t is the observed series, T_t is the trend, $S_{t,p}$ are the seasonal/cyclical components for period p, and R_t is the residual.

The decomposition plot (Figure 4) visually confirms the strong trend and illustrates the identified cyclical components. The residual component (R_t) notably displays non-constant variance (heteroscedasticity), indicating periods of higher and lower volatility, which is important for modeling.

Table VI quantitatively confirms the dominance of the trend and the strength of the identified long-term cycles, while conventional seasonality is negligible.

IV. Achieving Stationarity

Various techniques were applied to transform the non-stationary series into a stationary one, often a prerequisite for models like ARIMA.

Table VII indicates that differencing (first or second order) and moving average detrending are effective in achieving stationarity based on the ADF and KPSS tests. First-order differencing is commonly used and often sufficient.

Model Design and Implementation

Based on the data characteristics identified in Section III—namely the strong non-linear trend, complex multi-period cycles, and non-stationary residuals—a "divide and conquer" hybrid modeling strategy was designed.

I. Hybrid Model Architecture

The core of the design is an additive decomposition-based hybrid model, based on the MSTL decomposition in Equation 5. Our strategy is to forecast each component on the right-hand side independently and then sum the forecasts to produce the final prediction, $\hat{Y}_t$:

$$\hat{Y}_t = \hat{T}_t + \hat{S}_{t,143} + \hat{S}_{t,687} + \hat{S}_{t,3200} + \hat{R}_t \tag{6}$$

This component-wise approach allows us to use simpler, more appropriate models for each part of the series. The overall architecture of this pipeline is illustrated in Figure 5.

II. Data Preparation and Splitting

The `statsmodels.tsa.seasonal.MSTL` library was used to decompose the 'Close' price series using the previously identified periods [143, 687, 3200]. A critical pre-processing step involved aligning all components. After decomposition, the start and end of the series contain NaNs where the moving windows are incomplete. A **common index** was derived by finding the intersection of all non-NaN indices from the trend, all seasonal components, and the residual series. This ensures that all component data is perfectly aligned in time.

This aligned dataset, with 6471 data points, was split into training and testing sets using a **90% training ratio**. This resulted in 5823 data points for training and 648 data points for testing (the forecast horizon). This split was applied uniformly across all component series.

III. Component-Specific Modeling Strategies

A unique forecasting model was selected for each of the five components, as detailed below:

Trend (T_t)
Model: Polynomial Regression (Degree 2).
Implementation: A `sklearn` pipeline combining `PolynomialFeatures(degree=2)` and `LinearRegression` was fit on the training data, creating a model of the form:

$$\hat{T}_t = \beta_0 + \beta_1 t + \beta_2 t^2 \tag{7}$$

where t is the time index.
Rationale: The trend is smooth and non-linear. A quadratic polynomial provides a stable and non-overfit extrapolation for the long-term curve.

Seasonal 143-day ($S_{t,143}$)
Model: 50/50 Weighted Ensemble (Naive + LSTM).
Implementation:

- **Naive Forecast:** Repeats the last 143-day cycle from the training set.
- **LSTM Forecast:** A 'simple' 1-layer LSTM (50 units) trained on the component.

Rationale: This ensemble blends the stability of a simple seasonal naive forecast (which assumes the cycle repeats) with a simple LSTM's ability to learn and adapt to recent deviations in the cycle's pattern.

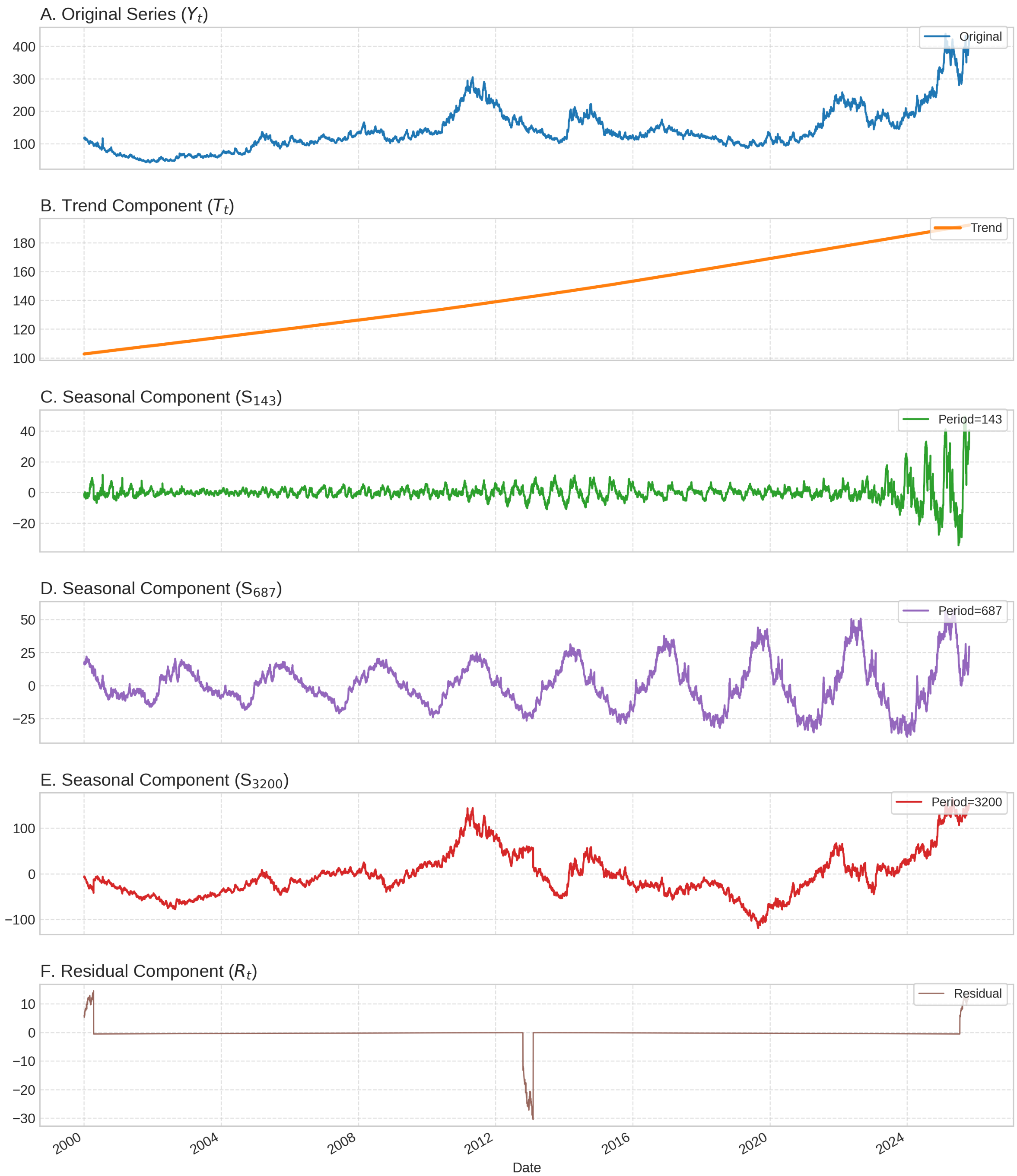

FIG. 4. MSTL DECOMPOSITION OF ORIGINAL TIME SERIES DATA WITH PERIODS [143, 687, 3200]

TABLE VI. MSTL Decomposition Strength Metrics

Metric	**Value / Interpretation**
Trend Strength (F_T)	0.988 – Very Strong Trend
Overall Seasonal Strength (F_S)	0.000 – Negligible Overall Seasonality
Strength (Period 143)	0.836
Strength (Period 687)	0.975
Strength (Period 3200)	0.997

TABLE VII. Stationarity Transformations and Test Results (p-values)

Technique	ADF p-val	KPSS p-val	Stationary?
Original Series	0.987	<0.01	No
First-order Differencing	<0.001	0.1	Yes
Second-order Differencing	<0.001	0.1	Yes
Log Transformation	0.810	<0.01	No
Square Root Transformation	0.913	0.01	No
Box-Cox ($\lambda \approx 0.019$)	0.780	0.01	No
Linear Detrending	0.831	<0.01	No
Moving Avg Detrend (w=30)	<0.001	>0.1	Yes

TABLE VIII. Comprehensive Summary of Pre-Forecasting Diagnostic Tests for Coffee Futures Data

Category	Test Name	Statistic	p-value	Result/Inference
Stationarity	Augmented Dickey-Fuller (ADF)	0.568	0.987	Non-stationary Data
	KPSS	0.722	<0.01	Non-stationary Data
Trend & Seasonality	Mann-Kendall Test	(Z-stat) High pos.	<0.001	Significant increasing trend
	ACF-PACF plots	–	–	No repeating patterns or significant lag in the data
Decomposition	STL Decomposition	–	–	Failed to decompose successfully (on its own)
	Classical Additive / Multiplicative	–	–	Failed to decompose successfully
Autocorrelation	ACF/PACF Plots	–	–	Significant spikes but non-uniform lag

Seasonal 687-day ($S_{t,687}$)
Model: 50/50 Weighted Ensemble (Naive + LSTM).
Implementation:

- **Naive Forecast:** Repeats the last 687-day cycle from the training set.
- **LSTM Forecast:** A 'default' 2-layer LSTM (64, 32 units).

Rationale: The same ensemble logic is applied to this stronger, medium-term cycle, but with a more robust 'default' LSTM architecture to capture its more complex dynamics.

Seasonal 3200-day ($S_{t,3200}$)
Model: LSTM Only.
Implementation: A 'complex' 2-layer LSTM (128, 128 units) with 50% Dropout.
Rationale: This cycle is extremely long (over 8 years), making a naive forecast impractical as it would require 8 years of data just to get one cycle. A dedicated, deep LSTM is used to model this very long-term dynamic behavior.

Residual (R_t)
Model: LSTM Only.
Implementation: A 'default' 2-layer LSTM (64, 32 units).
Rationale: The residual component contains the high-frequency noise and non-linear patterns (like volatility clusters) not captured by the trend or cycles. An LSTM is well-suited to model these complex, non-linear, and seemingly random short-term movements.

IV. LSTM Implementation Details

All LSTM models were built using `tensorFlow.Keras`.

- **Scaling:** To prevent data leakage, a `StandardScaler` was fit *only* on the training data for each component. This scaler was then used to transform both the train and test sets for that component.
- **Sequencing:** A lookback window (`TIME_STEP`) of **60 days** was used. The models were trained to predict the 61st day based on the previous 60 days.
- **Training:** All models were trained for **30 epochs** with a **batch size of 32**.
- **Optimizer & Loss:** The **Adam** optimizer was used, and the models were optimized to minimize **Mean Absolute Error (MAE)**, which is robust to outliers.

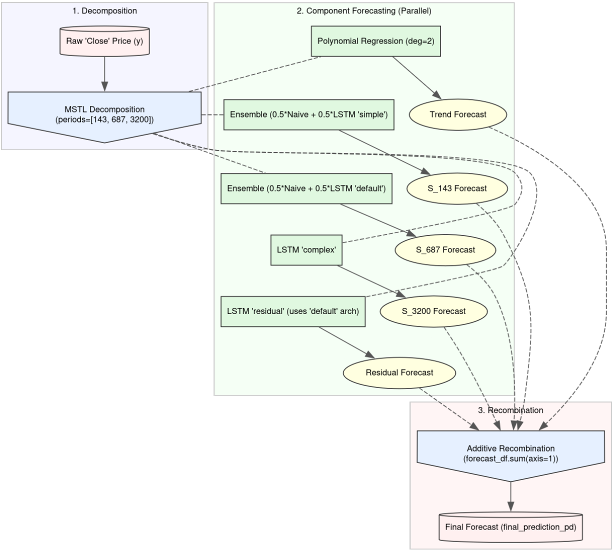
1. Decomposition
Raw 'Close' Price (y)
MSTL Decomposition
(periods=[143, 687, 3200])
2. Component Forecasting (Parallel)
Polynomial Regression (deg=2)
Trend Forecast
Ensemble (0.5*Naive + 0.5*LSTM 'simple')
S_143 Forecast
Ensemble (0.5*Naive + 0.5*LSTM 'default')
S_687 Forecast
LSTM 'complex'
S_3200 Forecast
LSTM 'residual' (uses 'default' arch)
Residual Forecast
3. Recombination
Additive Recombination
(forecast_df.sum(axis=1))
Final Forecast (final_prediction_pd)

Fig. 5. Flowchart of the Custom Hybrid Model Pipeline. The raw series is decomposed, each component is forecasted by a specialized model, and the forecasts are additively recombined.

The specific LSTM architectures used for the different components are summarized in Table IX and visualized in Figure 6.

TABLE IX. LSTM Model Architectures for Components

Component	Config	Layers & Units
$S_{t,143}$	Simple	L1: LSTM (50 units)
$S_{t,687}, R_t$	Default	L1: LSTM (64 units) L2: LSTM (32 units)
$S_{t,3200}$	Complex	L1: LSTM (128 units) L2: LSTM (128 units) L3: Dropout (0.5)

V. Experimental Setup and Hyperparameters

To ensure reproducibility and fair comparison, a consistent experimental setup was maintained across all models (slightly extended data than Time series analysis). The dataset was split chronologically with a 90% training and 10% testing ratio to respect temporal order. All Deep Learning models were trained using the Adam optimizer with a learning rate of 0.001, optimizing for Mean Absolute Error (MAE) to reduce sensitivity to outliers.

VI. Benchmark Selection Rationale

The benchmarks were specifically chosen to test the validity of the decomposition strategy:

TABLE X. Global Hyperparameters Model Configurations

Parameter	Value / Configuration
Training / Test Split	90% (5833 samples) / 10% (649 samples)
Lookback Window	60 Days
Optimizer	Adam ($\alpha = 0.001$)
Loss Function	Mean Absolute Error (MAE)
Batch Size	32
Epochs	30
Benchmark Model Configurations	
Trend-Adj. Naive	Trend: Polynomial Regression (Deg 2) Seasonality: Naive Repetition (Last Cycle)
ARIMA	Order $(p, d, q) = (5, 1, 0)$ *Selected based on PACF plots showing lag persistence.*
ELM	Hidden Neurons: 25 Activation: Sigmoid *Config based on best Arabica results from Deina et al. [2].*
Complex LSTM	Layers: 128 (RetSeq) → 128 → Dense 128 (ReLU) → Dropout (0.5) → Dense 1 *Identical architecture to the S_{3200} component to test monolithic performance.*

- **Trend-Adjusted Seasonal Naive:** Establishes a baseline for identifying whether the market dynamics are simply historical repetitions or require actual learning. By using Polynomial Regression for the trend (matching our proposed model), we isolate the evaluation to the cyclical forecasting capability.
- **ARIMA (5,1,0):** Represents standard linear stochastic modeling. Failure here confirms the non-linear, chaotic nature of the data.
- **Complex Monolithic LSTM:** Uses the exact same deep architecture (128 units, Dropout) as our most complex component (S_{3200}). This serves as an ablation study: if the Hybrid model outperforms this Monolithic version, it proves that MSTL decomposition contributes significantly to accuracy, rather than the neural network depth alone.

Results and Evaluation

The forecasts from the five individual component models were additively recombined to create the final hybrid forecast (see Equation 6). This final forecast was then evaluated against the held-out test set to assess the model's performance.

The model's performance was evaluated using three standard metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics are defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{8}$$

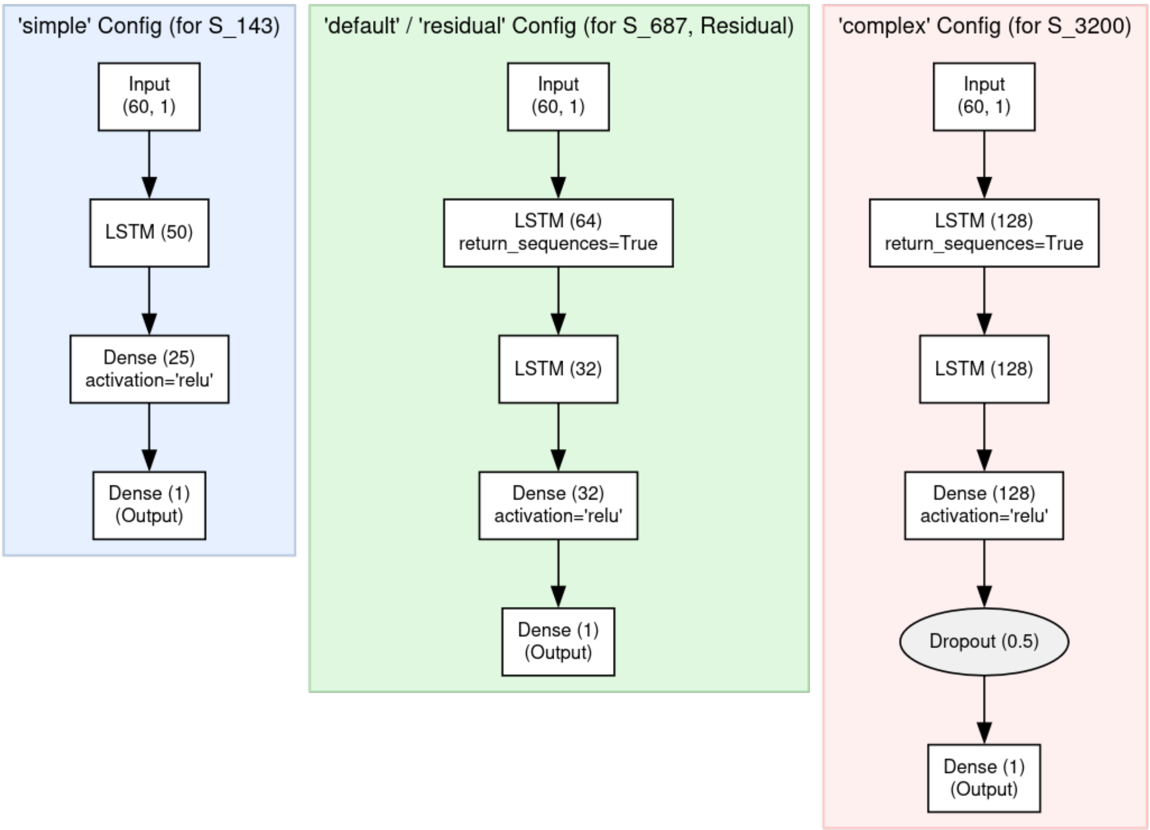
'simple' Config (for S_143)
Input (60, 1)
LSTM (50)
Dense (25) activation='relu'
Dense (1) (Output)
'default' / 'residual' Config (for S_687, Residual)
Input (60, 1)
LSTM (64) return_sequences=True
LSTM (32)
Dense (32) activation='relu'
Dense (1) (Output)
'complex' Config (for S_3200)
Input (60, 1)
LSTM (128) return_sequences=True
LSTM (128)
Dense (128) activation='relu'
Dropout (0.5)
Dense (1) (Output)

FIG. 6. VISUAL COMPARISON OF THE THREE CUSTOM LSTM ARCHITECTURES ('SIMPLE', 'DEFAULT'/'RESIDUAL', AND 'COMPLEX') DESIGNED TO HANDLE COMPONENTS OF VARYING COMPLEXITY.

$$\text{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t| \tag{9}$$

$$\text{MAPE} = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| \tag{10}$$

where n is the number of test points (648), y_t is the actual value, and $\hat{y}_t$ is the forecasted value.

I. Model Performance Metrics

The model achieved the following performance on the test set. The final evaluation metrics are presented in Table XI.

II. Forecast Visualization

A visual inspection of the forecast versus the actual prices provides a qualitative assessment of the model's performance. Figure 7 plots the actual 'Close' price (black line) against the model's one-step-ahead forecast (red dashed line) over the entire test period.

TABLE XI. Final Custom Hybrid Model Performance Metrics

Metric	Value
RMSE	10.5378
MAE	7.9016
MAPE	2.89%

Metrics calculated on 649 test data points.

The plot shows that the custom hybrid model tracks the complex movements of the coffee price, capturing the major peaks and troughs, including the sharp rise, the subsequent correction, and the final rally.

III. Comparative Analysis and Benchmarking

To validate the efficacy of our proposed framework, we benchmarked our performance against established methodologies. To establish a rigorous lower bound, we utilized a **Trend-Adjusted Seasonal Naive** model. This model isolates the trend component (using the same Polynomial Regression strategy as our proposed model) and combines

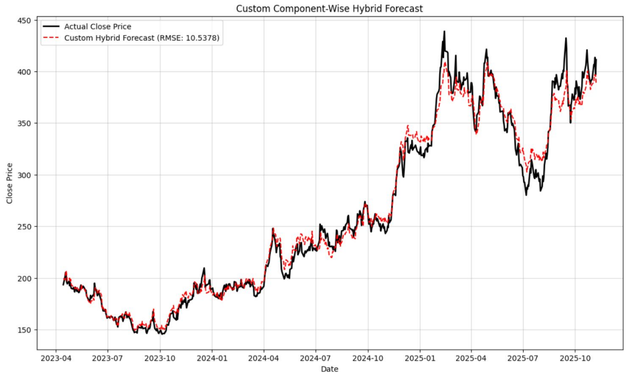

Fig. 7. Final Custom Hybrid Forecast vs. Actual Close Price on the test set. The model (red dashed line) is shown against the actual price (black line), achieving an RMSE of 10.54.

it with seasonal naive forecasts for the cyclical components.

We further compared our results against a traditional statistical baseline (**ARIMA**), a shallow learning model (**ELM**), and a monolithic Deep Learning model (**Complex LSTM**). The ”Complex LSTM” uses the exact same architecture as our S_{3200} component (128 units, 2 layers, Dropout) but is trained on the non-decomposed raw data. This serves as an ablation study to test whether the MSTL decomposition is adding value beyond the power of the LSTM itself.

TABLE XII. Benchmark Comparison against Standard Models

Model	**RMSE**	**MAE**	**MAPE (%)**
Trend-Adj. Seasonal Naive	79.53	71.07	29.03%
ARIMA (5,1,0)	106.77	79.68	25.90%
Extreme Learning Machine (ELM)	36.61	23.42	7.12%
Complex Monolithic LSTM	13.85	9.91	3.25%
Proposed MSTL-LSTM	**10.54**	**7.90**	**2.89%**

The poor performance of the Trend-Adjusted Seasonal Naive model (RMSE 79.53) confirms that the coffee market exhibits complex dynamics that cannot be captured by simple trend extrapolation and historical repetition. Similarly, the ARIMA model failed to capture the non-linear volatility.

Crucially, while the Complex Monolithic LSTM achieved a competitive RMSE of 13.85, a qualitative inspection reveals significant limitations regarding trend coherence and signal smoothness. Monolithic models trained on raw, volatile data often resort to ’persistence behavior,’ smoothing out the output and mirroring the previous time step rather than learning the underlying trajectory. This results in an artificially smooth forecast that reacts to noise rather than anticipating structural shifts. In contrast, by explicitly modeling the low-frequency trend component via Polynomial Regression and high-frequency volatility via residuals, the proposed MSTL-LSTM framework produces a highly responsive forecast. The resulting ’volatile’ curve (Figure 7) is not noise, but rather an accurate reflection of the market’s true behavior, capturing sharp reversals that the smoothed monolithic model misses. This structural robustness is confirmed by the superior error metrics (RMSE 10.54 vs 13.85).

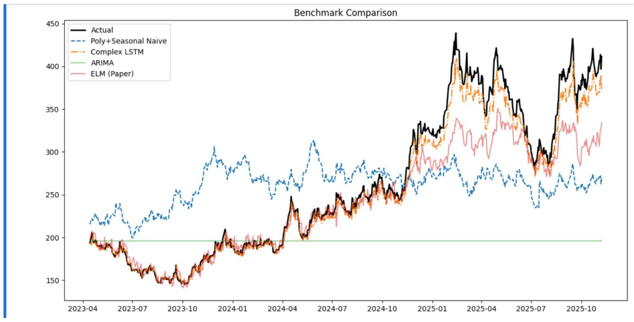

Fig. 8. Benchmark Comparison: The Trend-Adjusted Seasonal Naive and ARIMA models fail to track volatility. The proposed Hybrid model outperforms the monolithic Complex LSTM in both precision and trend tracking.

Stakeholder Insights

A forecast’s primary value lies in its ability to inform decision-making. The developed model’s predictions were analyzed to provide actionable insights for three key stakeholder groups. This is achieved by generating a long-term (90-day) forecast and then analyzing segments of this forecast to provide advice tailored to different time horizons.

I. Analysis Methodology

To provide relevant advice, the model’s 90-day forecast is segmented into three distinct time horizons: Short-Term (1-7 days), Medium-Term (30 days), and Long-Term (90 days). Each horizon is analyzed using advanced metrics:

Momentum (Trend): A linear regression line ($\hat{y} = mt+c$) is fitted to the forecast for each horizon. The slope (m) determines the momentum. A positive $m > 0.1$ is considered an ’Upward’ trend, $m < -0.1$ ’Downward’, and ’Stable’ otherwise.

Volatility: The standard deviation of the forecast (σ_f) is compared to the standard deviation of the last 90 days of historical data (σ_h). Volatility is flagged as ’High’ if $\sigma_f > 1.5 \times \sigma_h$ or ’Low’ if $\sigma_f < 0.75 \times \sigma_h$.

Key Levels: The maximum (Peak) and minimum (Trough) price within the forecast horizon are identified to establish a predicted trading range.

This multi-faceted analysis forms the basis for the stakeholder-specific recommendations presented in Table XIII.

Conclusion

This project developed a hybrid, decomposition-based model to forecast volatile coffee futures prices. By applying a divide and conquer strategy, we decomposed the complex, non-stationary time series into five simpler components: a long-term trend, three distinct cycles, and a high-frequency residual.

We then applied forecast models to each component, using Polynomial Regression for the smooth trend, Naive-LSTM ensembles for the shorter cycles, and dedicated LSTMs for the long-term cycle and the residual. This component-wise approach respects the unique statistical properties of each part of the series.

The final recombined forecast achieved a **MAPE of 2.89%** and an **RMSE of 10.54** on the 648-day test set. As shown in Figure 7, the model's forecast tracks the actual price movements, capturing the general direction of major peaks and troughs. This decompositional method provides a structured framework for modeling such a complex time series.

References

[1] T. K. Hwase and A. J. Fofanah, "Machine learning model approaches for price prediction in coffee market using linear regression, xgb, and lstm techniques," *International Journal of Scientific Research in Science and Technology (IJSRST)*, vol. 8, no. 6, pp. 10–48, Nov–Dec 2021, accepted: 01 Nov 2021; Published: 06 Nov 2021. [Online]. Available: https://www.ijsrst.com

[2] C. Deina, M. H. A. Prates, C. H. R. Alves, M. S. R. Martins, F. Trojan, S. L. Stevan, and H. V. Siqueira, "A methodology for coffee price forecasting based on extreme learning machines," *Information Processing in Agriculture*, 2021.

[3] R. L. Manogna, V. Dharmaji, and S. Sarang, "Enhancing agricultural commodity price forecasting with deep learning," *Scientific Reports*, vol. 15, p. 20903, 2025.

[4] N.-B.-V. Le, Y.-S. Seo, and J.-H. Huh, "Artificial intelligence in finance: Coffee commodity trading big data for informed decision making," *IEEE Access*, 2024.

[5] N. Kishaija, A. Ocwa, R. Kuunya, B. Ssemugenze, and B. Heil, "Climate change mitigation and livelihood components under smallholder coffee farming: a bibliographic and systematic review," *BMC - Agriculture & Food Security*, 2025.

[6] T. C. Pham, T. L. Nguyen, V. D. Hoang, and V. D. Nguyen, "Forecasting coffee prices with facebook prophet: A machine learning approach," in *Soft Computing and Its Engineering Applications*, ser. Communications in Computer and Information Science, K. K. Patel, K. Santosh, G. Gomes de Oliveira, A. Patel, and A. Ghosh, Eds. Springer, Cham, 2025, vol. 2430, pp. –, published: 16 May 2025. [Online]. Available: https://doi.org/10.1007/978-3-031-88039-1_16

[7] R. L. Manogna, V. Dharmaji, and S. Sarang, "A novel hybrid neural network-based volatility forecasting of agricultural commodity prices: empirical evidence from india," *Journal of Big Data*, 2025.

[8] S. A. Dhondiyal, S. R, and A. P. Dabral, "Coffee price prediction: An application of cnn-blstm neural networks," in *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*. Chennai, India: IEEE, 2023, pp. 1–7, check for updates. Verify currency and authenticity via CrossMark.

[9] D. Zhang, H. Xie, H. Hu, and Z. Yu, "Improving trend prediction of agricultural futures price using image encoding and attention mechanisms," *Management System Engineering*, p. 10, 2025.

[10] R. Kambo, J. Pillai, S. Soni, and G. Urkude, "Macpgana: design of a highly efficient multimodal agriculture commodity price prediction model via generative adversarial networks & autoencoders," *International Journal of Information Technology*, 2025.

[11] G. H. H. Nayak, M. W. Alam, B. S. Naik, B. S. Varshini, G. Avinash, R. R. Kumar, M. Ray, and K. N. Singh, "Meta-transformer: leveraging metaheuristic algorithms for agricultural commodity price forecasting," *Journal of Big Data*, 2025.

[12] S. Günther and G. B. C. G. de Sá, "oemof/cydets: v0.1.0," 3 2021. [Online]. Available: https://github.com/oemof/cydets?tab=readme-ov-file

[13] M. Peixeiro, *Time Series Forecasting in Python*. Manning Publications, 2022.

TABLE XIII. Stakeholder Recommendations Based on Forecast Analysis

Stakeholder	Time Horizon	Analysis & Recommendation
Investor / Trader	**Short-Term (1-7 Days)**	**If Trend is Upward:** Consider short-term long positions. High volatility warrants wider stops, while low volatility may signal a breakout. **If Trend is Downward:** Consider short-term short positions or put options. High volatility in a downtrend is particularly risky. **If Trend is Stable:** Consider range-trading strategies or options-selling (e.g., iron condors) to profit from low volatility.
Retailer / Buyer	**Medium-Term (30 Days)**	**If Trend is Upward:** *Action: Secure Inventory.* Consider buying in bulk now or locking in fixed-price contracts to protect margins. **If Trend is Downward:** *Action: Delay Purchases.* Operate on a Just-In-Time (JIT) basis to capitalize on falling prices. Use forecast to negotiate. **If Trend is Stable:** *Action: Maintain Normal Operations.* Continue standard purchasing cycles. Focus on quality and efficiency.
Producer / Farmer	**Long-Term (90 Days)**	**If Trend is Upward:** *Action: Hold Inventory.* If financially possible, delay sales to target the forecasted peak price. Avoid locking in low-priced contracts. **If Trend is Downward:** *Action: Sell Inventory.* A strong signal to sell current inventory. Aggressively seek forward contracts for future harvests. **If Trend is Stable:** *Action: Secure Partners.* Focus on building long-term relationships and contracts with reliable buyers.