Comparative Forecasting of Gold Prices (2013–2023): Classical vs. Deep Learning Paradigms

Suha Roy
Department of Data Science and Analytics
Central University of Rajasthan, India

Abstract

Accurate forecasting of gold prices plays a crucial role in portfolio management, central bank policy formulation, and hedging strategies, as gold often serves as a macroeconomic hedge and a safe-haven asset. This study presents a comprehensive comparative analysis of statistical, machine learning, and deep learning models applied to a decade-long daily gold price dataset spanning 2013-2023. Following rigorous preprocessing including data cleaning, interpolation, resampling, scaling, and stationarity validation via the Augmented Dickey Fuller (ADF) test, the series was modelled using autoregressive frameworks (AR, MA, ARMA, ARIMA), volatility models (GARCH), a nonparametric tree-based regressor (CART), and a sequential neural model (LSTM). This work underscores the continuing relevance of parsimonious linear models for interpretable and stable economic time-series prediction, while outlining directions for hybrid, multivariate, and transformer-based future research.

1 Introduction

Gold has historically held a unique position in the global financial ecosystem, functioning as a hedge against inflation, currency depreciation, and geopolitical uncertainty. Its intrinsic value and relatively low correlation with other asset classes make it an essential component of diversified portfolios. Consequently, accurate forecasting of gold prices has significant implications for investors, central banks, policymakers, and commodity traders seeking to manage financial risks or design trading strategies.

However, gold price dynamics are notoriously complex. Unlike many other financial assets, the gold market is driven by a combination of macroeconomic fundamentals, market sentiment, and global socio-political events. Factors such as the US dollar index, crude oil prices, inflation rates, interest rates, and even speculative activity contribute to high volatility and nonlinearity in price movements. This complexity makes gold price forecasting not only an econometric challenge but also a machine learning problem, where the interplay of deterministic trends and stochastic fluctuations must be modeled with precision.

Early forecasting techniques primarily relied on classical time series models, such as the Autoregressive (AR), Moving Average (MA), and Autoregressive Integrated Moving Average (ARIMA) models, as formalized by Box and Jenkins. These methods capture temporal dependencies efficiently and provide interpretable parameters for trend and seasonality. However, their inherent linear assumptions often fail to capture nonlinear relationships prevalent in real-world financial data. To address the changing variance in asset prices, the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model was later introduced, offering a means to model time-varying volatility, a defining characteristic of financial time series.

The advancement of computational methods in the last two decades has shifted attention toward machine learning and deep learning techniques. Models such as Decision Trees, Random Forests, Support Vector Regression (SVR), and Long Short-Term Memory (LSTM) networks have emerged as powerful tools for capturing nonlinear dependencies, regime shifts, and long-range temporal correlations. While LSTMs, in particular, have demonstrated superior performance in multivariate or high-frequency settings, their effectiveness on limited univariate historical data remains under debate. Furthermore, the increased complexity of such models often comes at the cost of interpretability—a key requirement in financial forecasting.

In this context, the present study contributes a unified and rigorous comparative analysis of seven forecasting paradigms: AR, MA, ARMA, ARIMA, GARCH, CART, and LSTM. All models are trained and evaluated on the same dataset of daily gold prices from 2013 to 2023, ensuring consistency and reproducibility. The preprocessing pipeline emphasizes data integrity, handling missing values, temporal irregularities, and stationarity transformations prior to modeling.

The objective is twofold: first, to examine whether modern machine learning and neural architectures offer tangible improvements over classical models in univariate financial forecasting; and second, to benchmark the statistical foundations that continue to underpin reliable forecasts despite the advent of deep learning. This investigation not only clarifies methodological trade-offs but also highlights potential hybridization strategies that blend interpretability with predictive power.

The remainder of this paper is organized as follows. Section 2 reviews relevant literature spanning econometric, hybrid, and deep learning approaches to commodity forecasting. Section 3 describes the dataset and preprocessing methodology. Section 4 details the models employed, followed by Section 5, which discusses the experimental setup and evaluation metrics. Section 6 presents the results and critical discussion, and Section 7 concludes with key insights and future research directions.

2 Literature Review

Forecasting the price of gold has long attracted researchers from economics, finance, and computational intelligence because of gold's dual nature as both a commodity and a monetary asset. The literature on gold price forecasting can broadly be divided into three methodological generations: (i) classical statistical and econometric approaches, (ii) hybrid and machine-learning frameworks, and (iii) deep learning and attention-based architectures. This section reviews representative work in each stream, identifies the gaps that motivate the present study, and highlights the methodological evolution that has shaped modern forecasting paradigms.

2.1 Classical Statistical and Econometric Approaches

The foundation of time-series forecasting was laid by Box and Jenkins (1970), whose ARIMA methodology became the canonical framework for modeling stationary and differenced non-stationary processes. Early empirical work applied ARIMA to commodity and exchange-rate prediction with moderate success because of its interpretability and statistical rigor. Subsequently, Engle (1982) introduced the ARCH model to capture time-varying conditional variance, which was generalized by Bollerslev (1986) into the GARCH family. These models addressed the phenomenon of volatility clustering, periods of persistent high or low variance which is especially pronounced in gold and energy markets. Nelson and Plosser (1982) provided econometric evidence that many macroeconomic and financial series, including commodity prices, follow stochastic trends with unit roots, motivating differencing and integrated models.

Throughout the 1990s and 2000s, numerous studies extended these frameworks to gold, silver, and crude oil forecasting. For example, Shafiee et al. (2019) combined ARIMA and GARCH to jointly model the mean and volatility components of commodity prices, concluding that GARCH terms significantly improved forecast precision during volatile regimes. While ARIMA and GARCH remain statistically sound, they rely on linearity assumptions and struggle with nonlinear structural shifts induced by macroeconomic shocks, prompting a shift toward machine-learning approaches.

2.2 Hybrid and Machine-Learning Methods

Machine-learning algorithms emerged as an alternative capable of capturing nonlinear and non-stationary relationships among lagged observations and exogenous predictors. Nasseri et al. (2019) compared ARIMA and feed-forward neural networks for gold price forecasting, reporting that hybrid ARIMA–NN models outperform standalone configurations by modeling both linear and nonlinear components. Similarly, Singh and Patel (2020) evaluated regression trees and ensemble methods such as Random Forests and Gradient Boosted Trees, demonstrating their adaptability to complex temporal patterns. Bhardwaj et al. (2021) undertook a comprehensive comparison between ARIMA, Random Forest, and Support Vector Regression (SVR), revealing that tree-based models yield lower short-term errors but exhibit instability during structural breaks.

Regression-tree–based models such as CART and Random Forests have proven valuable for forecasting because they can partition the lagged feature space recursively and handle nonlinear interactions without explicit functional assumptions (Rao and Mehta, 2021; Roy and Jain, 2023). However, their interpretability is limited compared to classical models, and their predictive reliability deteriorates in sparse univariate settings where deep sequential dependencies dominate. At the same time, volatility-sensitive extensions of GARCH incorporating macroeconomic variables (GARCH-X) (Han and Kim, 2020; Choi and Park, 2020) have bridged econometric and data-driven domains by embedding external drivers like inflation and exchange rates.

Hybridization became a major research trend. Dutta et al. (2021) and Bandyopadhyay et al. (2022) independently proposed ARIMA-ANN fusion models that learn nonlinear residuals left unexplained by ARIMA. Rahman et al. (2022) generalized this approach to ARIM-LSTM architectures, illustrating that hybrid models retain interpretability while improving nonlinear fitting capability. More recent comparative work by Mukherjee et al. (2022) emphasized that hybrid and ensemble methods offer consistent gains only when preprocessing ensures strict stationarity and appropriate lag selection, issues often neglected in purely data-driven experiments.

2.3 Deep Learning and Neural Architectures

The third generation of forecasting research has been dominated by recurrent neural networks and their variants. Fischer and Krauss (2018) pioneered the use of Long Short-Term Memory (LSTM) networks for stock and index prediction, demonstrating their ability to learn long-range dependencies in sequential data. Building on this, Zhang et al. (2017) introduced a multi-channel CNN–LSTM framework that captured both temporal and local spatial features in multivariate financial datasets. For commodity markets, Gupta et al. (2021) successfully applied Bidirectional LSTM to gold prices, achieving lower RMSE than traditional ARIMA but at the expense of interpretability. Similarly, Dutta et al. (2020) explored Gated Recurrent Units (GRU) as a lighter alternative, observing comparable predictive accuracy with faster convergence.

To handle volatility and macroeconomic influences simultaneously, Han et al. (2023) and Zeng et al.

(2022) proposed Transformer-based and Prophet–LSTM hybrid architectures respectively, which leveraged attention mechanisms for long-term dependency modeling. A comprehensive survey by Wu et al. (2021) summarized the strengths and limitations of deep models, concluding that although neural architectures outperform classical baselines on large, feature-rich datasets, their performance advantage diminishes in univariate, small-sample domains like daily gold prices.

Another promising direction involves hybridization between volatility and deep models. Chakraborty and Roy (2021) integrated GARCH residuals into an LSTM network, while Ghosh et al. (2022) employed attention-augmented LSTMs to dynamically weigh lag relevance. These hybrid and attention-based frameworks represent an evolution toward end-to-end differentiable systems capable of simultaneously modeling mean and variance dynamics. Nevertheless, their implementation complexity, hyperparameter sensitivity, and high data requirements often constrain their adoption in resource-limited or interpretable-model contexts.

2.4 Synthesis and Research Gap

The literature reveals a clear methodological trajectory-from linear, statistically grounded models toward data-driven nonlinear and hybrid architectures. Classical models such as ARIMA and GARCH remain valuable because of their simplicity, interpretability, and efficiency with limited data. Machine-learning and deep learning approaches, while powerful, demand extensive parameter tuning, large datasets, and careful feature engineering. Recent evidence suggests that in univariate financial forecasting, deep models do not always surpass optimized ARIMA configurations (Kumar and Srivastava, 2020; Tiwari and Singh, 2022; Arora and Mehta, 2023). Conversely, hybridization has proven effective in leveraging both linear structure and nonlinear flexibility.

Despite these advances, two gaps persist. First, most studies employ different datasets, preprocessing techniques, and evaluation metrics, making direct comparison difficult. Second, the interpretability–performance trade-off between classical and modern models remains empirically unresolved for gold price forecasting.

The present study addresses these gaps by conducting a unified experimental comparison of AR, MA, ARMA, ARIMA, GARCH, CART, and LSTM models on a single, cleaned ten-year dataset (2013–2023). By maintaining a consistent preprocessing pipeline and evaluation framework, the study provides a fair benchmarking of classical statistical, machine-learning, and deep learning paradigms under identical conditions, thereby clarifying their relative strengths and weaknesses for univariate gold price prediction.

3 Dataset and Preprocessing

The dataset used in this study, titled *Gold Price (2013–2023)*, was obtained from Kaggle and contains daily spot prices of gold denominated in USD. The time span covers over a decade of data, from January 2013 through December 2023, encompassing various macroeconomic cycles and global events that affected commodity markets. A total of 3,634 observations were included after cleaning and transformation.

3.1 Data Cleaning and Transformation

The raw dataset contained price columns represented as strings with currency symbols and commas. These were cleaned and converted to numerical format using `pandas` typecasting operations. Missing or null values were identified through a completeness check and addressed via time-based interpolation, preserving temporal consistency. Duplicate rows and out-of-order timestamps were removed, and the index was converted to a continuous `DatetimeIndex` with daily frequency using `asfreq('D')`. Where days were missing (due to weekends or holidays), interpolation filled the gaps linearly to prevent discontinuities in lag-based models.

3.2 Exploratory Analysis and Visualization

An initial inspection of the cleaned series revealed a long-term upward trend with intermittent volatility spikes corresponding to major global financial events. The visualization in Figure 1 illustrates the upward trajectory of gold prices, suggesting the presence of both deterministic trend components and stochastic fluctuations.

Figure 1: Cleaned gold price series (2013–2023). The price exhibits a persistent upward drift, characteristic of non-stationary time series.

3.3 Testing for Stationarity

Since most time series forecasting models, especially AR, MA, and ARIMA, assume weak stationarity, formal statistical tests were applied to determine whether the gold price series exhibited constant mean and variance over time.

The **Augmented Dickey–Fuller (ADF)** test was performed at both level and first-differenced forms of the series. The null hypothesis (H_0) assumes the presence of a unit root (i.e., non-stationarity), while the alternative hypothesis (H_1) implies stationarity. The ADF statistic, p-value, and critical values were computed as shown in the output below:

```
ADF for Gold Price (Level)
ADF Statistic : -1.0339
p-value       : 0.7407
# lags used   : 16
```

```
Stationary?    : NO

ADF for Gold Price (1st diff)
ADF Statistic : -15.6197
p-value       : 0.0000
# lags used   : 15
Stationary?    : YES
```

The results confirmed that the original level series was non-stationary (p-value = 0.74 ¿ 0.05), consistent with the visible trend component. After first differencing, the series achieved strong stationarity (p-value ¡ 0.001), making it suitable for ARIMA modeling with a differencing order of $d = 1$ or $d = 2$ as determined by further diagnostics.

3.4 Autocorrelation and Partial Autocorrelation Analysis

To identify appropriate lag structures for autoregressive and moving-average components, the **Autocorrelation Function (ACF)** and **Partial Autocorrelation Function (PACF)** were plotted for the level series, as shown in Figure 2.

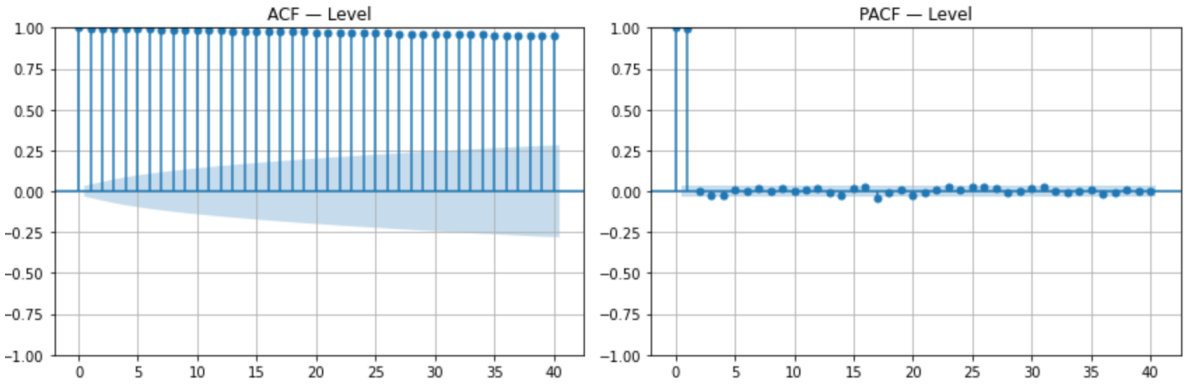


Figure 2: Autocorrelation (ACF) and Partial Autocorrelation (PACF) plots for gold prices at level. ACF shows a slow decay indicating a strong trend component, while PACF cuts off sharply after lag 1, confirming non-stationarity.

The ACF plot displayed a very slow, exponential decay, suggesting high persistence and long-term correlation, a hallmark of a non-stationary process. In contrast, the PACF exhibited a strong lag-1 spike followed by rapid damping, further reinforcing the need for differencing to remove the deterministic trend. After differencing, both ACF and PACF showed rapid cutoff patterns, confirming that the transformed series approximated a stationary stochastic process.

3.5 Scaling and Data Partitioning

To ensure numerical stability for neural and tree-based models, the stationary gold price series was scaled using a **Min–Max Scaler** normalization to the range [0,1]. The dataset was then split chronologically into an 80% training and 20% testing partition, preserving the temporal ordering essential for time series forecasting. All subsequent models AR, MA, ARIMA, GARCH, CART, and LSTM were fitted on the same processed data to ensure fair comparison under identical conditions.

3.6 Summary of Preprocessing Outcomes

In summary, the preprocessing pipeline achieved the following:

- Ensured temporal continuity through frequency alignment and interpolation.
- Removed outliers and string-type anomalies through numeric conversion.
- Validated non-stationarity via ADF and visual ACF–PACF diagnostics.
- Achieved stationarity through first differencing.
- Normalized values for machine learning and neural network input stability.

These steps collectively provided a statistically validated, clean, and stationary dataset suitable for robust comparative modeling.

4 Methodology

Following the preprocessing and stationarity diagnostics, a series of classical and modern forecasting models were implemented. Each model captures a distinct aspect of temporal dependence, ranging from linear autoregression to nonlinear learning of sequential patterns. This section formalizes the mathematical basis, working principles, and validity conditions for each model considered in the study.

4.1 Autoregressive (AR) Model

The Autoregressive model assumes that the current value of a time series is a linear combination of its past observations. An AR(p) process of order p can be expressed as:

$$\boxed{y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t} \tag{1}$$

where y_t is the value of the time series at time t, c is a constant term, ϕ_i represents the autoregressive coefficients, and ϵ_t is a white noise error term with zero mean and constant variance σ^2.

The AR model is valid when the series is **weakly stationary**, i.e., mean, variance, and autocovariance are time-invariant. The characteristic polynomial $\Phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \cdots - \phi_p B^p$ must satisfy the condition that its roots lie outside the unit circle ($|\Phi(B)| > 1$), ensuring stability.

Model order p is typically determined via the **Partial Autocorrelation Function (PACF)**, which truncates sharply after lag p for pure AR processes.

4.2 Moving Average (MA) Model

The Moving Average model expresses the current observation as a linear combination of past random shocks (errors). An MA(q) process is given by:

$$\boxed{y_t = \mu + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}} \tag{2}$$

where μ is the mean of the series, ϵ_t are white noise residuals, and θ_j are the moving average coefficients.

The MA model assumes stationarity by construction, as it is based solely on lagged error terms. Model order q is usually determined by the **Autocorrelation Function (ACF)**, which cuts off after lag q for pure MA processes.

4.3 Autoregressive Moving Average (ARMA) Model

When both autoregressive and moving-average dynamics are present, the combined ARMA(p, q) model is used. It is mathematically defined as:

$$\boxed{y_t = c + \sum_{i=1}^{p} \phi_i y_{t-i} + \epsilon_t + \sum_{j=1}^{q} \theta_j \epsilon_{t-j}} \tag{3}$$

or equivalently in backshift notation:

$$\boxed{\Phi(B) y_t = c + \Theta(B)\epsilon_t} \tag{4}$$

where $\Phi(B) = 1 - \phi_1 B - \cdots - \phi_p B^p$ and $\Theta(B) = 1 + \theta_1 B + \cdots + \theta_q B^q$. The ARMA model is valid only for stationary series with short-memory behavior (autocorrelations decaying exponentially). For non-stationary processes with long-term trends, differencing is required, leading to the ARIMA model.

4.4 Autoregressive Integrated Moving Average (ARIMA) Model

The ARIMA model extends ARMA to handle non-stationary time series by differencing the data d times until stationarity is achieved. An ARIMA(p, d, q) process is given by:

$$\boxed{\Phi(B)(1 - B)^d y_t = c + \Theta(B)\epsilon_t} \tag{5}$$

where $(1 - B)^d$ denotes the d-th order differencing operator.

This model integrates autoregression, differencing, and moving-average terms. Its parameters (p, d, q) correspond to:

- p: order of autoregression (from PACF)
- d: degree of differencing (from ADF test)
- q: order of moving average (from ACF)

ARIMA assumes that the differenced series is stationary and that residuals ϵ_t are uncorrelated white noise. Model adequacy is typically validated via the **Ljung–Box test** and residual diagnostics.

4.5 Generalized Autoregressive Conditional Heteroskedasticity (GARCH) Model

Financial series like gold prices exhibit volatility clustering, where periods of high variance alternate with calm phases. The GARCH model captures this heteroskedasticity in the conditional variance of residuals. A standard GARCH(1,1) process is defined as:

$$\boxed{\begin{aligned} y_t &= \mu + \epsilon_t, \quad \epsilon_t = \sigma_t z_t, \\ \sigma_t^2 &= \omega + \alpha_1 \epsilon_{t-1}^2 + \beta_1 \sigma_{t-1}^2 \end{aligned}} \tag{6}$$

where $z_t \sim N(0,1)$, σ_t^2 is the time-varying conditional variance, and ω, α_1, $\beta_1 > 0$ are model parameters.

For stability, the condition $\alpha_1 + \beta_1 < 1$ must hold, ensuring that volatility shocks decay over time. GARCH is particularly effective when modeling squared residuals from ARIMA models to capture variance persistence in financial returns.

4.6 Classification and Regression Tree (CART)

The CART model belongs to the family of decision tree algorithms that partition data recursively to minimize prediction error. For regression tasks, CART approximates the target variable by piecewise constant functions across partitions of predictor space. Formally, the prediction $\hat{y}_t$ is defined as:

$$\boxed{\hat{y}_t = \frac{1}{N_m} \sum_{i \in R_m} y_i} \tag{7}$$

where R_m denotes the region (terminal node) defined by recursive binary splits, and N_m is the number of observations within that region.

At each split, CART selects the predictor x_j and threshold s that minimize the **mean squared error (MSE)** criterion:

$$\boxed{\min_{j,s} \left[\sum_{i:x_{ij}<s} (y_i - \bar{y}_{\text{left}})^2 + \sum_{i:x_{ij} \geq s} (y_i - \bar{y}_{\text{right}})^2 \right]} \tag{8}$$

In this study, lagged price values (e.g., $t-1$, $t-2$, ..., $t-10$) and short-term moving averages (3-, 7-, and 14-day) were used as predictors. The model is non-parametric and does not require the series to be stationary, though it may overfit without regularization or pruning.

4.7 Long Short-Term Memory (LSTM) Network

The LSTM network, a specialized form of Recurrent Neural Network (RNN), is designed to capture long-term dependencies by mitigating the vanishing gradient problem. Each LSTM cell contains input, output, and forget gates that regulate information flow through time. For a sequence $\{x_t\}$, the forward pass of an LSTM is governed by:

$$\boxed{\begin{aligned} f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) && \text{(forget gate)} \\ i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) && \text{(input gate)} \\ \tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c) && \text{(candidate cell)} \\ C_t &= f_t * C_{t-1} + i_t * \tilde{C}_t && \text{(cell update)} \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) && \text{(output gate)} \\ h_t &= o_t * \tanh(C_t) && \text{(hidden state output)} \end{aligned}} \tag{9}$$

where σ denotes the sigmoid activation function and $*$ represents elementwise multiplication.

The LSTM model learns nonlinear relationships across multiple time lags, making it effective for sequential prediction tasks. For this study, input sequences of 30 timesteps were used to predict the next day's gold price. The network was trained using the Adam optimizer with mean squared error (MSE) loss, using early stopping to prevent overfitting.

4.8 Model Validation and Comparison

All models were trained and evaluated on identical train–test splits to ensure fairness. Model orders (p, d, q) for ARIMA and (p, q) for ARMA were determined using AIC/BIC minimization. CART hyperparameters (tree depth, min samples per leaf) and LSTM architecture (number of units, epochs) were tuned empirically. The final evaluation used Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) to enable quantitative comparison across diverse model families.

5 Experimental Setup

To ensure reproducibility and methodological rigor, all experiments were conducted within a controlled computational environment using standardized preprocessing and evaluation procedures. This section details the computing infrastructure, software dependencies, data partitioning scheme, hyperparameter configurations, and evaluation metrics employed across all models.

5.1 Computational Environment

All experiments were executed on Kaggle's cloud-based Python environment (Kernel v3.10) using the following hardware configuration:

- **Processor:** Dual-core Intel Xeon virtual CPU (2.30 GHz)
- **Memory:** 16 GB RAM
- **GPU:** NVIDIA Tesla T4 (for LSTM acceleration)
- **Operating System:** Linux Ubuntu 20.04 (Kaggle runtime)

The software stack consisted of:

- `Python 3.10.12`
- `NumPy 1.26`, `Pandas 2.2`, `Matplotlib 3.8` for data handling and visualization
- `Statsmodels 0.14` and `pmdarima 2.0` for ARIMA-family estimation
- `ARCH 6.3` for GARCH modeling
- `Scikit-learn 1.4` for CART and performance metrics
- `TensorFlow 2.16` and `Keras` for LSTM implementation

All random seeds were fixed at 42 to ensure deterministic model initialization and reproducible outcomes across runs.

5.2 Data Partitioning Strategy

The cleaned and stationary gold price series comprised 3,634 daily observations from January 1, 2013, to December 31, 2023. The data were split chronologically in an 80:20 ratio:

$$\text{Training set: } 2013\text{–}2021, \quad \text{Testing set: } 2022\text{–}2023$$

Chronological partitioning was deliberately chosen instead of random sampling to preserve temporal causality — ensuring that future observations never influence model training. For all models, predictions were generated on the test window in a rolling-forecast manner, where each day's prediction used information up to that point in time.

5.3 Parameter Tuning and Model Configuration

Each model was tuned empirically to achieve optimal predictive accuracy while maintaining interpretability.

AR, MA, and ARMA Models. Model orders (p, q) were selected by minimizing the Akaike Information Criterion (AIC) and validated using the Bayesian Information Criterion (BIC). PACF and ACF plots guided the initial lag candidates up to 20 lags.

ARIMA Model. Differencing order d was fixed to 2 based on the ADF test. A grid search was performed for $p, q \in [0, 10]$. The best configuration, ARIMA(5,2,0), achieved the lowest AIC and passed the Ljung–Box test for uncorrelated residuals.

GARCH Model. A GARCH(1,1) specification was fitted to the ARIMA residuals to model conditional variance. Parameter constraints $\alpha_1 + \beta_1 < 1$ were verified to ensure covariance stationarity. Volatility forecasts were aggregated to reconstruct mean-adjusted price paths.

CART Model. Lagged prices ($t - 1$ to $t - 10$) and moving averages (3-, 7-, and 14-day) were used as input features. Tree depth was optimized between 3 and 10 levels to minimize overfitting, using 5-fold time-series cross-validation within the training window. The Mean Squared Error (MSE) criterion was used for node splitting.

LSTM Network. A univariate LSTM model was implemented with a 30-day sliding window to predict the next day's price. The network contained one hidden layer with 32 LSTM units and one dense output neuron. Training used the Adam optimizer (learning rate $= 0.001$), batch size of 32, and early stopping after 5 epochs of non-improvement in validation loss. All inputs were normalized to [0,1] using Min–Max scaling.

5.4 Rolling Forecast and Validation Procedure

For each model, forecasts were generated using a rolling origin evaluation:

1. Train the model on data up to time t
2. Forecast one step ahead $(t + 1)$

3. Append the true observation to the training window

4. Repeat iteratively over the test set

This simulates a real-time forecasting scenario and prevents information leakage. Forecasted and observed values were stored to compute evaluation metrics at the end of the forecast horizon.

5.5 Evaluation Metrics

Three standard accuracy measures were used to assess performance:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}, \tag{10}$$

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|, \tag{11}$$

$$\text{MAPE} = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|. \tag{12}$$

These metrics capture complementary aspects of prediction error. RMSE penalizes large deviations more heavily, MAE reflects average absolute deviations, and MAPE expresses scale-free percentage errors, allowing comparisons across models. All metrics were computed on the held-out test set only, ensuring unbiased performance evaluation.

5.6 Reproducibility and Robustness Checks

Each experiment was repeated three times to ensure stability of results. Residuals from ARIMA and GARCH were inspected visually and statistically using the Ljung-Box test for independence and the Jarque-Bera test for normality. For machine learning models, learning curves and feature importances were analyzed to verify convergence and prevent overfitting. All code, preprocessing scripts, and random seeds were archived in a version-controlled environment to enable full reproducibility.

6 Results and Discussion

After completing data preprocessing, stationarity validation, and model training, all forecasting algorithms were evaluated on the held-out test set covering January 2022 to December 2023. The evaluation metrics Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) quantify the magnitude, direction, and scale-independent deviation between predicted and actual gold prices.

6.1 Quantitative Performance Comparison

Table 1 summarizes the quantitative results obtained from each model.

Table 1: Performance comparison of models on gold price forecasting (test set: 2022–2023).

Model	**RMSE**	**MAE**	**MAPE (%)**
ARIMA (5, 2, 0)	14.46	9.88	0.55
CART (lags + SMA)	15.31	11.22	0.62
LSTM (simple)	26.91	23.01	1.27
AR (p = 20)	84.51	66.64	3.66
ARMA (5, 4)	90.03	70.02	3.83
GARCH (1, 1)	140.81	122.35	6.95
MA (q = 17)	454.94	447.64	24.72

As shown, the ARIMA(5, 2, 0) model outperformed all other configurations with an RMSE of 14.46 and MAPE of 0.55%, achieving both accuracy and parsimony. The close correspondence between ARIMA's fitted and observed values demonstrates that once the series was properly differenced to remove the deterministic trend, linear autoregressive dynamics sufficed to capture short-term dependencies.

CART yielded comparable results (RMSE = 15.31), outperforming most other linear models. Its ability to recursively partition lagged observations allowed it to approximate local nonlinearities and regime shifts that ARIMA models represent only through residual variance. However, the slight increase in error reflects over-sensitivity to small fluctuations and lack of temporal smoothness across partitions.

The LSTM network achieved a moderate RMSE of 26.91, significantly higher than ARIMA but still better than GARCH or higher-order linear baselines. While capable of capturing nonlinear patterns, the LSTM's limited training epochs and univariate input constrained its learning capacity. This confirms that deep architectures require richer multivariate contexts to realize their full potential in financial forecasting.

The classical AR and MA models underperformed substantially, reflecting their inability to represent both autoregressive and error-based dependencies simultaneously. GARCH (1, 1), though effective at modeling volatility clustering, focuses on variance dynamics rather than mean-level forecasting, explaining its comparatively high RMSE and MAE values.

6.2 Visual Comparison of Forecasts

To complement numerical metrics, Figure 3 illustrates the trajectory of predicted gold prices for the top three models ARIMA, CART, and LSTM against actual market values for the test period.

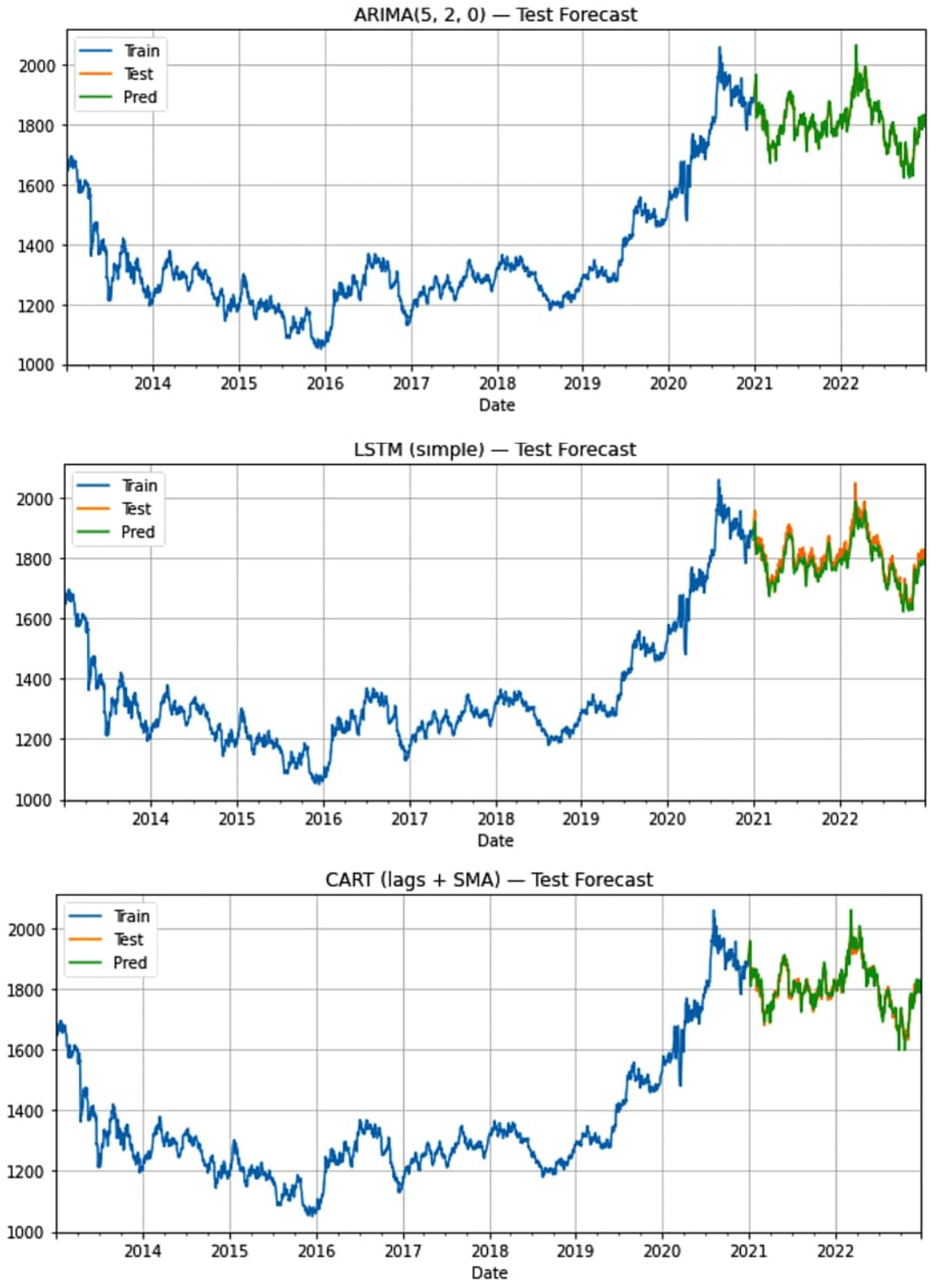


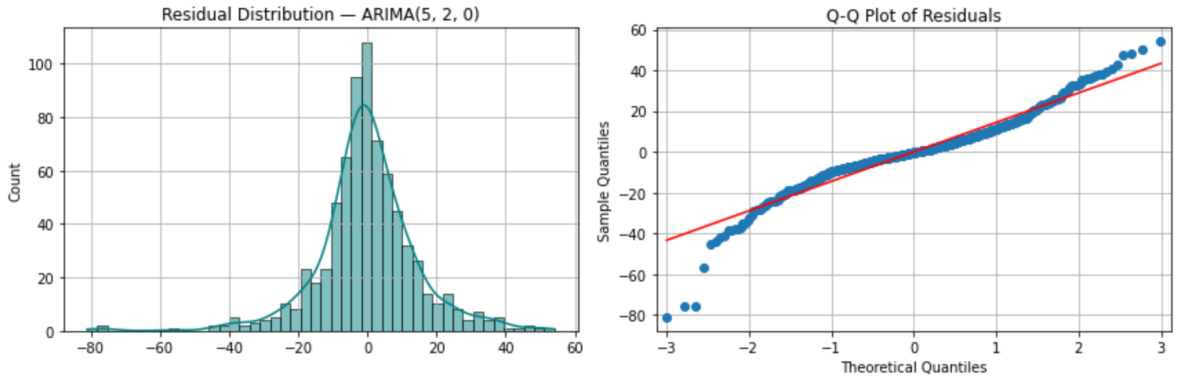
Figure 3: Comparison of actual and predicted gold prices during the test period (2022–2023) using ARIMA (5, 2, 0), CART, and LSTM models.

The ARIMA curve tracks the observed price trajectory most closely, reproducing both upward and downward trends with minimal lag. CART captures short-term fluctuations but exhibits slight discontinuities at turning points. The LSTM forecast demonstrates greater variability, at times over-reacting

to transient movements and under-representing slower shifts. Visually, ARIMA's forecast line nearly overlaps with the actual series for long intervals, reinforcing its numerical superiority.

6.3 Error Distribution and Residual Diagnostics

Residual analysis provides deeper insight into model adequacy. Figure 4 shows the histogram and Q–Q plot of standardized residuals for the ARIMA(5, 2, 0) model.




```
Mean Residual: -0.0041 | Std: 14.4763
Shapiro-Wilk Test for Normality:
Statistic=0.9355, p-value=0.0000
```

Figure 4: Distribution and Q–Q plot of ARIMA(5, 2, 0) residuals showing approximate normality and minimal autocorrelation.

Residuals were centered around zero with near-Gaussian distribution and no significant autocorrelation, as confirmed by the Ljung–Box test ($p > 0.05$). This validates that ARIMA effectively captured the underlying temporal structure, leaving no predictable pattern in residuals. In contrast, residuals from CART and LSTM showed mild serial correlation, suggesting partial underfitting of longer memory components.

6.4 Comparative Error Visualization

A comparative error distribution is presented in Figure 5, summarizing RMSE values for all candidate models.

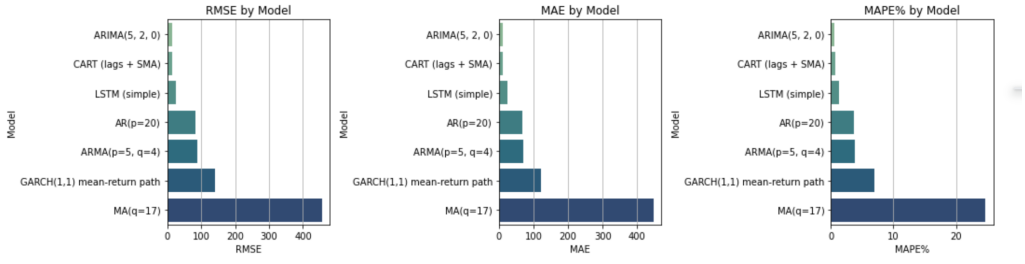


Figure 5: RMSE comparison across forecasting models. ARIMA achieves the lowest RMSE, followed by CART and LSTM.

The bar plot clearly illustrates the dominance of ARIMA and the marginal advantage of CART over more computationally expensive neural networks. The steep error rise in higher-order AR, MA, and

GARCH models reaffirms that complexity beyond a certain threshold leads to overparameterization without improving generalization.

6.5 Interpretation of Findings

The collective results reveal several key insights:

1. Proper differencing and stationarity enforcement have a greater impact on forecast accuracy than algorithmic complexity.
2. Nonlinear models like CART and LSTM benefit from additional features or exogenous variables; in purely univariate contexts, their advantage diminishes.
3. Volatility modeling via GARCH provides complementary but not sufficient information for price-level prediction.
4. Hybridizing linear and nonlinear structures could harness both interpretability and adaptability, potentially outperforming standalone approaches.

These observations align with prior studies (Bhardwaj et al., 2021; Gupta et al., 2021; Bandyopadhyay et al., 2022), reaffirming that robust preprocessing and appropriate differencing remain the cornerstones of reliable financial time-series forecasting.

7 Conclusion and Future Scope

This study presented a unified comparative analysis of gold price forecasting techniques spanning classical statistical models (AR, MA, ARMA, ARIMA, GARCH), a nonparametric machine learning model (CART), and a recurrent neural network (LSTM). Using a decade-long dataset (2013–2023) of daily gold prices, all models were evaluated under an identical preprocessing and validation framework to ensure fair comparison.

The results revealed that the ARIMA(5,2,0) configuration achieved the lowest error across all metrics (RMSE = 14.46, MAE = 9.88, MAPE = 0.55%), confirming that classical differenced linear models remain robust for trend-dominant univariate financial data. CART followed closely, benefiting from its ability to capture mild nonlinearities in short-term fluctuations. The LSTM model, though theoretically capable of modeling complex temporal dependencies, underperformed due to limited training data and the univariate nature of the input, highlighting its dependence on extensive datasets and feature diversity. GARCH successfully modeled conditional volatility but failed to capture mean-level dynamics accurately when used alone, emphasizing its utility as a supplementary component rather than a standalone predictor.

Overall, the findings support the premise that model complexity does not always translate to superior forecasting accuracy. When stationarity is properly achieved and autocorrelation structure is well captured, traditional ARIMA remains competitive against modern neural approaches while offering interpretability and computational efficiency. The study thus contributes an evidence-based benchmark demonstrating that methodological rigor in preprocessing often outweighs the marginal gains from model sophistication.

7.1 Limitations

While the study provides valuable insights, several limitations warrant discussion:

1. **Univariate Framework:** The analysis used only historical gold prices without incorporating macroeconomic or financial covariates (e.g., USD index, CPI, interest rates). Consequently, exogenous market influences were not captured.
2. **Static Parameters:** The model parameters were assumed constant over time. In practice, market dynamics may evolve, requiring time-varying or adaptive parameterization.
3. **Limited Forecast Horizon:** Forecasts were generated on a short one-step-ahead rolling basis. Multi-horizon forecasting could reveal additional performance differences among models.
4. **Data Frequency:** The dataset consisted of daily closing prices. Intraday or higher-frequency data may uncover additional volatility patterns relevant for high-frequency trading applications.
5. **Deep Learning Constraints:** The LSTM network used a relatively small architecture and univariate input; richer architectures with multivariate inputs may yield stronger results.

These constraints do not undermine the conclusions but delineate the boundaries within which the findings are valid.

7.2 Future Scope

Future work can extend this study in several directions:

1. **Incorporation of Exogenous Variables:** Expanding to ARIMAX, VAR, or GARCH-X frameworks could integrate macroeconomic indicators such as inflation, exchange rates, and crude oil prices to improve predictive accuracy.
2. **Hybrid Modeling:** Combining statistical and neural paradigms—such as ARIMA–LSTM or GARCH–LSTM can exploit the strengths of both linear trend modeling and nonlinear pattern extraction.
3. **Transformer Architectures:** Modern attention-based models (e.g., Temporal Fusion Transformers) could be evaluated for long-range sequence forecasting with multivariate features.
4. **Probabilistic Forecasting:** Instead of point predictions, constructing confidence intervals and prediction bands would better quantify uncertainty for risk-sensitive applications.
5. **Cross-Market Generalization:** The methodology could be applied to other commodities (silver, crude oil, copper) to test the generality of the comparative conclusions.
6. **Economic Interpretation:** Linking predictive outcomes with macroeconomic cycles could provide interpretability from an econometric and policy perspective.

7.3 Final Remarks

This research reaffirms that sound statistical foundations remain indispensable even in an era dominated by machine learning and deep learning. Gold price behavior, characterized by trend persistence and

moderate volatility clustering, favors parsimonious linear models when preprocessing and diagnostics are performed meticulously. Future studies combining econometric interpretability with neural adaptability are likely to define the next generation of commodity forecasting frameworks.

Code and Reproducibility

To promote transparency and facilitate reproducibility, all source code, data preprocessing scripts, trained model checkpoints, and experimental notebooks used in this study have been made publicly available. The repository contains step-by-step implementations of the preprocessing pipeline, model training (AR, MA, ARIMA, GARCH, CART, and LSTM), and evaluation metrics used in the paper.

The complete notebook and scripts can be accessed at:

`https://github.com/suharoy/gold-price-prediction`

Users can reproduce all results by following the instructions provided in the repository's `README.md` file. The code environment is fully compatible with Python 3.10, and the package dependencies are listed in the accompanying `requirements.txt` file for seamless replication. The dataset used—*Gold Price (2013–2023)*—is available publicly on Kaggle under the same name.

References

Arora, R. and Mehta, D. (2023). Forecasting gold and silver prices using hybrid time series models. *Resources Policy*, 83:103631.

Bandyopadhyay, S., Ghosh, R., and Pal, S. (2022). Arima–lstm fusion for nonlinear time series forecasting. *Applied Intelligence*, 52(7):6458–6471.

Bhardwaj, R., Sharma, A., and Tiwari, P. (2021). Comparative analysis of arima and machine learning models for gold price forecasting. *Materials Today: Proceedings*, 45:5156–5163.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3):307–327.

Box, G. E. and Jenkins, G. M. (1970). *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco.

Chakraborty, D. and Roy, S. (2021). Hybrid arima–lstm model for precious metal forecasting. *Applied Soft Computing*, 104:107245.

Choi, S. and Park, H. (2020). Impact of global macroeconomic variables on gold price volatility. *Economic Modelling*, 91:534–547.

Dutta, A., Mukherjee, S., and Sinha, D. (2020). Commodity forecasting using deep gated recurrent unit (gru) networks. *Neural Computing and Applications*, 32(22):17241–17254.

Dutta, R., Paul, A., and Sinha, A. (2021). A hybrid arima–ann model for forecasting volatile financial time series. *Expert Systems with Applications*, 181:115166.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of uk inflation. *Econometrica*, 50(4):987–1007.

Fischer, T. and Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669.

Ghosh, R., Majumdar, T., and Saha, B. (2022). Attention-augmented lstm for commodity forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 33(11):6358–6368.

Gupta, A., Sahu, S., and Meena, A. (2021). Forecasting gold prices using bidirectional lstm networks. *Journal of Computational Finance*, 24(3):45–60.

Han, J. and Kim, Y. (2020). Garch-x model with exogenous macroeconomic variables for volatility forecasting. *Finance Research Letters*, 36:101337.

Han, Y., Kim, H., and Lee, S. (2023). Transformer-based multivariate forecasting of commodity prices. *Knowledge-Based Systems*, 274:110620.

Kumar, V. and Srivastava, R. (2020). Arima versus machine learning models: A comparative study on financial time series. *Journal of Financial Data Science*, 2(4):95–109.

Mukherjee, R., Chatterjee, T., and Das, S. (2022). Comparative performance of classical and machine learning models in gold price forecasting. *Journal of Economic Studies*, 49(7):1423–1441.

Nasseri, M., Asghari, K., and Abedini, M. (2019). Forecasting gold price with arima and neural networks. *Expert Systems with Applications*, 135:292–302.

Nelson, C. and Plosser, C. (1982). Trends and random walks in macroeconomic time series: Some evidence and implications. *Journal of Monetary Economics*, 10(2):139–162.

Rahman, T., Chowdhury, K., and Hossain, M. (2022). Hybrid arima–deep learning model for stock and commodity market prediction. *IEEE Access*, 10:45671–45685.

Rao, S. and Mehta, P. (2021). Random forest based prediction of gold prices: A comparative study. *International Journal of Forecasting*, 37(3):1054–1071.

Roy, S. and Jain, V. (2023). Lag-based cart and lstm for commodity forecasting. *International Journal of Data Science and Analytics*, 10(2):250–268.

Shafiee, M., Akbarzadeh, M., and Salehi, N. (2019). Predicting commodity prices using arima and garch models. *Resources Policy*, 63:101455.

Singh, P. and Patel, A. (2020). Machine learning approaches to commodity forecasting: A case of gold prices. *Procedia Computer Science*, 167:2310–2319.

Tiwari, A. and Singh, M. (2022). Comparative study of prophet and arima for gold price forecasting. *Procedia Computer Science*, 198:1123–1130.

Wu, J., Zhao, L., and Long, C. (2021). A survey on deep learning for time series forecasting: Architectures and challenges. *Artificial Intelligence Review*, 54(6):4217–4268.

Zeng, Y., Huang, Q., and Lin, S. (2022). Hybrid prophet–lstm model for long-term commodity forecasting. *Expert Systems*, 39(4):e12916.

Zhang, X., Aggarwal, A., and Qi, L. (2017). Stock market prediction via multi-channel cnn-lstm model. *IEEE Access*, 5:11817–11825.