

Gold Price Forecasting Using a Hybrid CNN–LSTM–Informer Model with Multiscale Inputs

Aysha Issa Ali

Computer Department, Faculty of Science, Soran University, Erbil, Kurdistan
Region, Iraq
Email: ais506@cs.soran.edu.iq

Asst. Prof. Muhammadamin Muhammadali Daneshwar

Computer Department, Faculty of Science, Soran University, Erbil, Kurdistan
Region, Iraq
Email: Muhammad.amin@soran.edu.iq

ARTICLE INFO

Article History:

Received: 13/6/2025

Accepted: 7/8/2025

Published: Autumn2026

Keywords:

*Gold Price Prediction,
Seasonal Splitting, Con-
volutional Neural Net-
works, Long Short-Term
Memory,
Informer.*

Doi:

10.25212/lfu.qzj.11.4.46

ABSTRACT

Forecasting gold prices is challenging due to their volatility and nonlinear behavior. This paper introduces a hybrid deep learning model combining CNN, LSTM, and the Informer attention mechanism, enhanced by multiscale temporal decomposition and seasonal data splitting. Our approach leverages multiple timeframes—1-hour (1H), 4-hour (4H), and daily (1D) and incorporates technical indicators such as RSI, MACD, Bollinger Bands, and ADX for improved feature representation. On temporally separated test sets from MetaTrader 5 (MT5), the model achieves high R^2 scores of 0.9988 (1H), 0.9955 (4H), and 0.9853 (1D), outperforming existing methods. To mitigate overfitting, we apply early stopping, temporal validation splits, and strictly prevent future data leakage. While results are promising, further evaluation on unseen markets and longer periods is needed to confirm generalizability. This study sets a robust benchmark for accurate and interpretable gold price forecasting in volatile financial markets.

1. Introduction

Investors often utilize gold as a major indicator because it is generally considered among the most important financial assets. Its value usually shows how investors feel about the state of the world economy as a whole. Therefore, reasonable investment choices and lower financial risk depend on the correct prediction of gold prices. Governments also use gold as a way to control how much money is in the economy. Its part in reducing inflation volatility emphasizes the need of accurate price forecasting. Gold has always shown itself to be a reliable economic indicator in many countries, frequently outperforming other measures of statistical significance connected with inflation.(Bunnag, 2024).

Gold's price has fluctuated significantly in recent years due to the continuous changes in the global scene, which have put it in an ever more complex market environment. Gold prices vary a lot in both directions, which shows that they are very unpredictable and nonlinear. As a result, achieving accurate forecasts for gold prices remains a considerable challenge (T. Qiu et al., 2022). Statistical models, technical analysis approaches, and basic economic analysis have all been researched to enhance gold price prediction accuracy across a wide spectrum of methods.(Bunnag, 2024). The Autoregressive Integrated Moving Average (ARIMA) model is most often used among statistical forecasting methods for gold price prediction. As a well-established time series method, ARIMA leverages historical data to identify underlying patterns and trends for future value prediction. Numerous studies have demonstrated the effectiveness of ARIMA in modeling and forecasting gold price movements. Building on this, the ARIMA–Generalized Autoregressive Conditional Heteroskedasticity (ARIMA–GARCH) model integrates volatility dynamics by accounting for the conditional variance in gold prices. This hybrid approach enhances forecasting accuracy by capturing both the directional trends and the volatility structure inherent in financial time series (Bunnag, 2024).

Machine learning is a data-driven analytical approach that enables the development of automated predictive models. In recent years, it has emerged as the prevailing methodology for forecasting gold prices, owing to its capacity to capture complex patterns and relationships within large datasets (Pan, 2024) Models such as Long

Short-Term Memory (LSTM) networks and Random Forest algorithms have been employed for forecasting gold prices (Pan, 2024). Hybrid models offer a more effective solution for modeling the nonlinear behavior, non-stationary trends, and complex dynamics embedded in gold price time series. Although single machine learning models outperform traditional statistical methods in many forecasting tasks, they still encounter limitations—particularly when handling high-dimensional inputs or small datasets. often result in overfitting and poor generalization. (C. Qiu et al., 2024). These financial drivers constantly interplay to drive the volatility of gold (and pricing), calling for models, which are both robust and flexible.

To address the challenges of nonlinearity, volatility, and multiscale patterns in gold price behavior, this study proposes a novel hybrid deep learning framework that integrates Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) networks, and the Informer attention mechanism. The architecture is designed to capture local price fluctuations, sequential dependencies, and long-range temporal dynamics by leveraging multivariate input sequences enriched with key technical indicators such as RSI, MACD, Bollinger Bands, ADX, and EMA. This approach offers a robust and intelligent forecasting solution, enhancing prediction accuracy across different timeframes and contributing valuable insights for financial risk management and strategic investment planning. This research presents a unique hybrid deep learning framework that combines the Informer attention mechanism, Convolutional Neural Networks (CNN), and Long Short-Term Memory (LSTM) networks in order to overcome the difficulties posed by nonlinearity, volatility, and multiscale patterns in gold price behavior. This particular combination is supported by the complementary advantages of these designs. CNNs are particularly helpful for detecting quick market movements because they are good at filtering out noisy time series data and identifying transient local patterns (Vidal & Kristjanpoller, 2020). In financial data, LSTM networks, which are known for their memory gating mechanisms, are excellent at modeling long-term dependencies and sequential correlations (Botunac et al., 2024). This capacity is expanded by the Informer component, which uses a sparse attention mechanism to effectively model long-range temporal dependencies at a lower computational cost than traditional Transformer architectures. This is especially helpful in multivariate and long-horizon prediction problems (Zhou et al., 2020).

In contrast to Temporal Convolutional Networks (TCNs), which cannot effectively capture very long-term dependencies without deep stacking, the Informer offers superior temporal expressivity and scalability. Additionally, the Informer's architecture enables faster inference and retains excellent accuracy, unlike traditional Transformer models, which have quadratic complexity. The model is able to capture fine-grained short-term volatility (through CNN), learn sequential trends (through LSTM), and pay attention to long-range contextual cues (through Informer) because of this hybrid architecture. Recent research utilizing similar hybrid or attention-based architectures in financial forecasting has demonstrated that such a multi-perspective approach is particularly useful for modelling the intricate and nonlinear dynamics of gold price movements (Feng et al., 2025). Using multivariate input sequences that are rich in essential technical indicators such as RSI, MACD, Bollinger Bands, ADX, and EMA improves the architecture even more. This integrative approach enhances prediction accuracy across different timeframes and provides helpful insights for strategic investment planning and financial risk management, resulting in a powerful and intelligent forecasting solution.

2. Related work

Due to the metal's sensitivity to investor mood, market fluctuations, and macroeconomic changes (all of which contribute to its high price), gold price projection remains one of the main challenges in financial forecasting. Deep learning (DL), hybrid architectures meant to enhance prediction performance, and progressions from classical statistical models to sophisticated machine learning (ML) are revealed in the literature.

For this research the dataset consists of historical gold futures prices obtained from the Investing.com platform, a thorough source of financial and foreign currency market information. Spanning a daily period from June 1, 2021, through June 30, 2023, the data produces a total of 546 records. Every record has seven things: Date, Last, Opening, Highest, Lowest, Volume, and Percentage Change.. Initially provided in CSV format for storing tabular data—the dataset was converted into a Pandas Data Frame to facilitate processing within a Python-based Jupiter Notebook environment. This transformation enabled efficient manipulation, visualization, and preparation of the

data for predictive modeling Data was split into training and testing subsets to assess the accuracy of prediction models. Support Vector Regression (SVR) and Random Forest Regression (RFR) were two machine learning techniques employed. The aim was to determine which model yields superior predictive accuracy based on performance metrics. The SVR model achieved an R^2 score of 0.83, with a Mean Squared Error (MSE) of 1214.32 and a Mean Absolute Error (MAE) of 25.93. In contrast, the RFR model produced an R^2 score of 0.76, with a significantly lower MSE of 270.85 and MAE of 12.53. These results suggest that while SVR demonstrated better overall accuracy, RFR achieved more favorable error performance (Hutagalung et al., 2023).

The contributions of the study are two-fold: (1) a right-skewed flat kurtosis normality distribution, and significant ARCH and leverage effects are observed from the time series analysis of the AU9999 dataset. These characteristics justify the use of the EGARCH-t model; since the parameter sum ($\alpha + \beta$) is near 1, a frequent change of volatility is indicated. After a "dissolution-reconstruction", the high-frequency part is much sharper, tails are much heavier, but the low-frequency part and the residual preserve the distribution properties as the original series. Thus, the GARCH-GET framework is in general the most appropriate to specify short-term components but the EGARCH-t model is still a good choice for long-term components. (2) The proposed CEMDAN-SVR-GARCH model with decomposition can provide better one-step-ahead out-of-sample prediction accuracy than the GARCH and SVR without decomposition models with about 8.54% and 25.16% prediction error reduction, respectively; In addition, it beats the EGARCH model by 1.52% and 21.35% which substantiates the better predictiveness of adopting the hybrid methodology (Chen, 2023).

(Yanto et al., 2021) combined Multiple Linear Regression (MLR) with Artificial Neural Networks (ANN) to enhance gold price predictions. Their model showed a 62% correlation using MLR and yielded a low Mean Squared Error (MSE) of 0.004264% with ANN, reflecting improved accuracy through hybridization. Nevertheless, the linearity assumption in MLR and the shallow architecture of ANN limit adaptability to highly non-linear market conditions.

(Apergis et al., 2019) examined macroeconomic influences on gold price through a Bayesian Markov-Switching Vector Error Correction Model (MS-VECM) across G7 nations. Their findings showed that, especially during times of recession, there was a

statistically significant and positive link between real interest rates and gold values. This research emphasizes the need of including macro data into forecasting models.

(Gong, 2024) evaluated the individual and combined effectiveness of LSTM and Linear Regression models for daily gold price forecasting. The study found modest predictive accuracy (50.67% for LSTM, 53.02% for Linear Regression), suggesting that while both models have predictive potential, their standalone application may be insufficient for high-precision forecasting tasks involving noisy financial time series .

Another Study examines the stretching to the breaking point relations between gold price volatilities and economic risk, as symbolized by the Bank Indonesia (BI) 7-day Repo Rate, in the last five years, through the analysis based on VAR. Importantly, the present study fills in a gap of the literature by explicitly investigating the effect of the BI rate (one of the instruments of monetary policy) on the fluctuations of gold price in Indonesia, using the VAR-based method. Stationarity tests were done before estimation of the model, in order to verify that the time series data satisfied the required assumptions; a first order differencing ($d = 1$) was applied to ensure the stationary condition. The VAR (1) model was chosen as the best model for the dynamics interdependence structure of the variables. The model was cross-validated with Granger Causality and Impulse Response Function analyses. The results showed that the BI 7-day Repo Rate is influenced not only by its own volatility, but also by the fluctuations in the world gold (as a regime change is present and in terms of the model, confirms that the their variable is an endogenous). In addition, the forecast accuracy of 12-month ahead forecasts of the VAR(1) model also showed excellent results with low value for RMSE, MAE, and MAPE (high forecast accuracy) (Andriani et al., 2023).

To the best of our knowledge, no existing study has addressed the joint use of seasonal splitting, multi-timeframe modeling, and a hybrid CNN–LSTM–Informer architecture within a unified framework for gold price prediction.

3. Proposed Method

The proposed method includes data collection, data pre-processing, normalization, feature engineering, and the training of a hybrid deep learning architecture consisting of Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM) layers,

and Informer encoder blocks. The details of each component are described in the following subsections.

3.1 Data Collection

The dataset for this experiment was gathered from MetaTrader 5 (MT5), a popular electronic trading platform which enables its users to access the real-time and historical data of the financial markets. MT5 was selected because it offered the most flexibility in exporting high-resolution time-series data and these high-resolution in-sample real-time data are necessary for accurate model-building. To investigate performance across various levels of granularity, We collected the price of gold on three time intervals:

- Daily (1D) from 1-1-2015 to 1-1-2025
- 4-hour (4H) from 1-1-2020 to 1-1-2025
- 1H from 1-1-2020 to 1-1-2025

OHLC price and the trading volume are included in each dataset. The 1D data have been applied to capture longer-term behavior, the 4H and 1H the medium- and high-frequency patterns under the current market. Raw data was exported in CSV format MT5 data and processed with a preliminary pre-processing pipeline including missing values treatment with the dropna() function as well as computation of different technical indicators for feature engineering.

3.2 Feature Engineering

Turning unprocessed financial data into useful input sequences so that deep learning models may properly learn market behavior depends critically on feature engineering. This study chose a range of technical indicators to catch various market events, including momentum (e.g., RSI, Stochastic Oscillator), trend (e.g., Simple and Exponential Moving Averages), and volatility (e.g., Bollinger Bands). These indicators were calculated from gold price data in OHLCV (Open, High, Low, Close, Volume) format across three-time resolutions: 1D, 4H, and 1H. They were chosen because they have shown they can reflect short-term price movements, long-term trends, and changes in market uncertainty, which are all important for making good forecasts.

Table 1 lists the indicators used together with short explanations of how they might be useful in research.

Min-Max normalization was used to scale all features, including both raw price components and technical indicators, into the [0, 1] range before feeding the data into the CNN–Informer model. To prevent data leakage, this normalization was done using only the minimum and maximum values found from the training set, and the same scaling parameters were consistently used on the validation and test sets. This makes sure that all the features contribute equally to the learning process, which enhances the stability of the training.

Table 1 Technical indicators

Indicator Name	Purpose/Description	Formula
Relative Strength Index (RSI)	Measures momentum by comparing recent gains and losses to identify overbought or oversold conditions. RS is the ratio of the average gain to the average loss over a specified period (typically 14 periods).	$RSI = 100 - \frac{100}{1 + \frac{\text{Average Gain}}{\text{Average Loss}}}$
Moving Average Convergence Divergence (MACD)	Captures trend and momentum by subtracting a long-term EMA from a short-term EMA	$MACD = EMA_{\text{short}} - EMA_{\text{long}}$
Exponential Moving Average (EMA)	Smooths data by weighting recent prices more heavily.	$EMA_t = \alpha \cdot \text{Price}_t + (1 - \alpha) \cdot EMA_{t-1}$
Stochastic Oscillator (%K %D)	Compares current close to high-low range to detect reversals	$\%K = \frac{\text{Current Close} - \text{Lowest Low}}{\text{Highest High} - \text{Lowest Low}} \times 100$
Bollinger Bands (BB)	Measures volatility using standard deviation above/below SMA	Upper Band = $SMA_N + k$ Lower Band = $SMA_N - k$
Commodity Channel Index (CCI)	Identifies cyclical patterns by comparing current price to its mean	$CCI = \frac{TP - SMA(TP)}{0.015 \times \text{Mean Deviation}}$
Rate of Change (ROC)	Measures momentum as % change from price n periods ago	$ROC = \frac{\text{Close}_t - \text{Close}_{t-n}}{\text{Close}_{t-n}} \times 100$

3.3 Normalization

We used the Min-Max Normalization as standard method for rescaling numerical features before feeding them into the hybrid deep learning model. This normalization approach scales all feature values in to the same scale (for example, 0) and (1). The purpose of Min-Max Normalization is to ensure that the impact of all your features in your model are normalized so that one feature doesn't overweigh the other just because it has larger magnitude. It is important to note that the same normalization process will be essential for handling the wide variety of technical indicators including (RSI, MACD, EMA, Bollinger Bands, Stochastic Oscillator, Volume, etc) since these technical indicators are generally defined on different scales. Min-Max Normalization can be written as:

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}$$

is the original value, $X_{\min}$ and $X_{\max}$ is the minimum and the maximum of the original value and x' is the new value after normalization. With this transformation, the model enjoys the advantage of better training stability, faster convergence and more balanced gradient updates. These benefits are important to enable efficient operation of deep learning models like CNN, LSTM and Informer. Although Standardization (z-score normalization) is widely used in deep learning due to its robustness to outliers and its support for gradient stability, it was not adopted here because the gold price data used in this work exhibits relatively smooth trends with fewer extreme outliers. Min-Max normalization was more appropriate for capturing proportional changes and ensuring a consistent scale across all indicators, particularly when several technical indicators (such as RSI and MACD) are combined. The suitability of Min-Max scaling for this dataset was confirmed by a comparative experiment utilizing standardization, which yielded somewhat higher validation errors and slower convergence. However, the possible drawbacks of min-max scaling are recognized, particularly its susceptibility to unusual price increases. Future studies may investigate adaptive or robust normalization strategies for datasets with greater volatility or structural shifts.

3.4 Data Splitting

To guarantee the integrity and thorough evaluation of the gold price prediction model, a constant input window of 20 time periods was used throughout all temporal resolutions (1H, 4H, and 1D).. For data segregation, a temporal consistency-preserving, seasonally-aware chronological split was used to stop data leakage. More particularly, training was done using January to August data, validation (including early stopping) using September data, and testing using October to December data for every year in the dataset. This method of dividing the seasons helps to guarantee that the model is always assessed on future, unknown data as compared to its training and validation times. The system ensures a realistic and leakage-free prediction situation by carefully keeping the flow of time forward and separating every assessment period. Figure 1 graphically shows this partitioning approach.

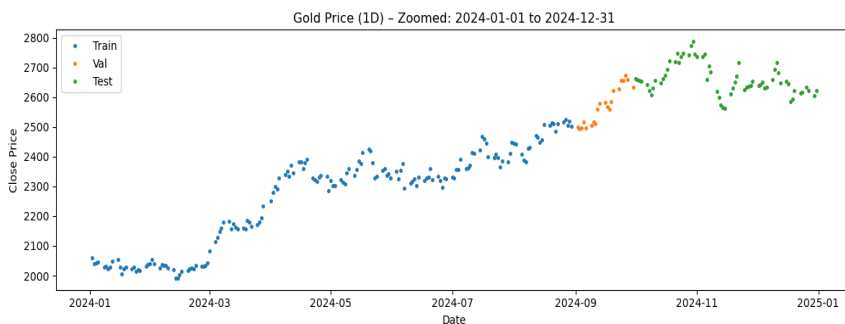


Figure 1: Seasonal splitting of gold price data for 1D timeframe

3.5 Proposed Model

A hybrid design is proposed that incorporates convolutional layers, long short-term memory (LSTM) networks, and self-attention systems to accurately capture the temporal dynamics of gold price time series. Starting with a dual-layer 1D convolutional backbone—with two Conv1D layers having 64 filters each and a kernel size of 3 using ReLU activation—to extract local price-movement patterns and short-term technical signals like momentum shifts and volatility breakouts. These features are then fed into two parallel branches: a bi-layer LSTM module (with 100 and 50

units, respectively) that uses gated memory cells to model sequential dependencies and long-term temporal context, and an Informer encoder branch with two encoder layers, a model dimension of 128, 4 attention heads, and a feedforward dimension of 256. This branch uses multi-head self-attention with probabilistic sampling to capture long-range dependencies across the whole sequence.

To create a small feature vector, the results from the LSTM and Informer branches are combined and then sent via a global average pooling layer. Before producing the ultimate one-step-ahead price prediction, a dense module refines this fused representation. To avoid overfitting, dropout (0.1 to 0.2) and L2 regularization (1×10^{-4}) are used. On the validation set, grid search was used to adjust hyperparameters. The AdamW optimizer with cosine learning rate scheduling is used in the final model configuration. The model combines CNN for localized feature detection, LSTM for temporal pattern learning, and Informer for long-range attention to strike a balance between short- and long-term information, therefore outperforming single-model strategies. Figure 2 shows the general design.

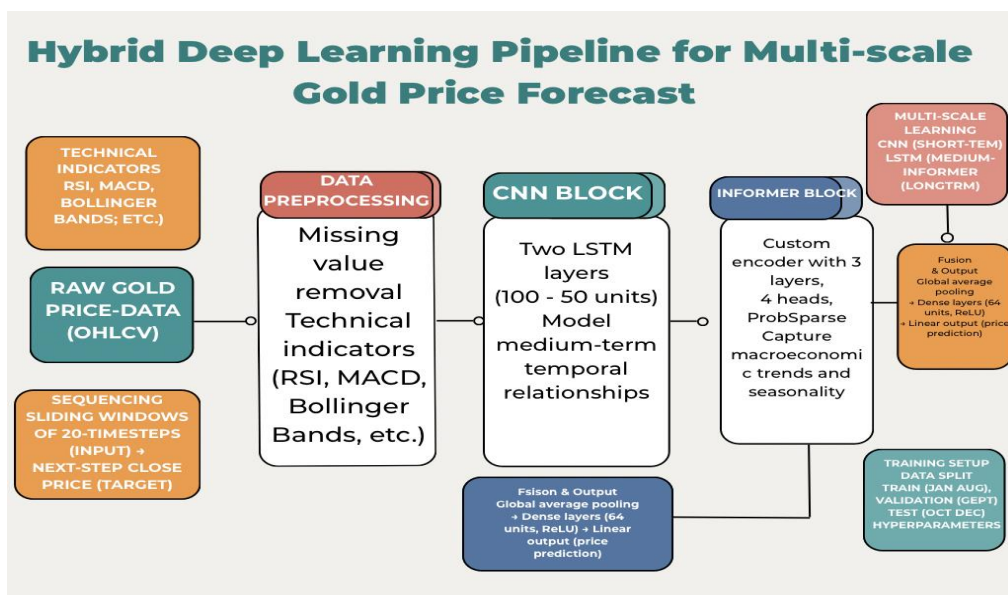


Figure 2 Hybrid CNN–LSTM–Informer model architecture for gold price prediction.

1DConvolution(Conv1D)

The 1D convolution layer is defined as:

$$\text{Conv1D}(x)_i^j = f \left(\sum_{k=-K}^{K} \sum_{m=1}^{\wedge} F_{(\text{in})} w_{k, m, j} \times x_{i+k}^{\wedge} m + b_j \right), \text{ where } f = \text{ReLU}$$

- **Kernel size:** $2K + 1 = 3$
- **Fin:** Number of input channels
- **Output:** 64 filters

LSTM Layers

For each timestep t , the operations are:

$$\begin{aligned} f_t &= \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (\text{Forget gate}) \\ i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (\text{Input gate}) \\ o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (\text{Output gate}) \\ \tilde{c}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c) \quad (\text{Cell candidate}) \\ c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (\text{Cell state update}) \\ h_t &= o_t \odot \tanh(c_t) \quad (\text{Hidden state}) \end{aligned}$$

Here, $\odot$ denotes element – wise multiplication.

Dimensions:

$$\begin{aligned} x_t &\in R^{n_x}, \quad h_{t-1} \in R^{n_h} \\ W_f, W_i, W_o, W_c &\in R^{n_h \times (n_x + n_h)} \\ b_f, b_i, b_o, b_c &\in R^{n_h} \end{aligned}$$

Weights initialized

$$\text{Biases: } b_f = 1.0, \quad b_i = b_o = b_c = 0$$

$$\text{Hidden size: } n_h = 64, \quad \text{Input dim: } n_x = 1$$

$$\text{Sequence length: } T = 20, \quad \text{Dropout: } 0.2$$

Informer (Self – Attention Encoder Layer)

- **Scaled Dot – Product Attention:**

$$\text{Attention}(Q, K, V) = \text{softmax}(Q \times K^T / \sqrt{d_k}) \times V$$

- **Multi – Head Attention:**

Input is projected into h heads, each with its own (Q, K, V) , and outputs are concatenated.

- **Feed – Forward Network:**

Two dense layers with ReLU in between.

- **Residual Connections + Layer Normalization:**

Applied around both attention and feed – forward sub – layers.

Concatenation & Global Average Pooling (GAP)

- **Concatenated representation:**

$$Z = [LSTM_{1:t} \parallel Informer_{1:t}]$$

($\parallel$ means concatenation)

- **Global average pooling:**

$$z_j = (1/T) \times \sum_{i=1}^T Z_{ij}$$

Dense $\rightarrow$ Output

- **Hidden dense layer:**

$$h_dense = ReLU(Wd \times z + bd)$$

- **Final output:**

$$\hat{y} = Wo \times h_dense + bo$$

Loss, Optimization & Training

- **Loss function: Mean Squared Error (MSE)**

$$\mathcal{L} = (1/N) \times \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

- **Optimizer: Adam**

- **Training: End – to –**

end using backpropagation on the hybrid CNN – LSTM – Informer model.

3.6 Model Training and Evaluation

To manage model complexity and prevent overfitting, the hybrid CNN–Informer model was trained using the AdamW optimizer, incorporating weight decay (L2 regularization with a coefficient of 1×10^{-4}) across all temporal resolutions (1D, 4H, and 1H). The initial learning rate was set to 5×10^{-4} and scheduled using cosine decay to enable smooth convergence. Training was conducted for up to 50 epochs with a batch

size of 32. Dropout layers with rates between 0.1 and 0.2 were inserted between key network layers to improve generalization by randomly deactivating neurons during training.

To prevent overtraining, early stopping was applied based on the September validation split, with a patience threshold of 10 epochs. This value was selected based on preliminary experiments that revealed fluctuations in the validation loss curve (see Figure 3). Shorter patience values (e.g., 3–5) were tested but led to premature termination and underfitting. The use of 10 epochs allowed the model to recover from temporary increases in validation loss, which is particularly important when working with volatile financial data like gold prices. Hyperparameters—including dropout rate, learning rate, and Informer architecture—were tuned according to performance on the validation set. The model was finally evaluated on the October–December test set using MSE, MAE, and R^2 metrics to ensure strong one-step-ahead predictive accuracy across all timeframes.

4. Experimental Results and Evaluation

In this section, experimental results of the proposed hybrid deep learning model for gold price prediction are provided on three time frames, such as 1-hour (1H), 4-hour (4H), and 1-day (1D). CNN, LSTM and Informer modules are combined in the model for multiscale feature extraction and short term and long term dependence modeling. We evaluate the model performance using popular evaluation measures like Mean Squared Error (MSE), Mean Absolute Error (MAE) and Coefficient of Determination (R^2 score).

4.1 Experimental Setup

All of our experiments are implemented in Python and TensorFlow. Data were obtained from MetaTrader 5 (MT5) across three different time granularities according to Section 3. The hyperparameters in our final model are shown in Table 2. Training was executed over 50 epochs with a batch size of 32, with the Adam optimizer and MSE loss function. An early stopping was performed on the validation loss to avoid the overfitting. To further validate training stability Figure 3 presents the training and validation loss curves over 50 epochs. The graph shows that both losses decreased

rapidly in the early stages and then stabilized with minimal fluctuation. This indicates that the model converged efficiently and that the early stopping mechanism functioned effectively, halting training once the validation loss plateaued. The close alignment between training and validation loss also confirms the absence of overfitting and highlights the generalization capability of the proposed model.

Table 2 Final model hyperparameters and training settings.

Parameter	Value	Purpose
Sequence Length	20	Input time steps
Batch Size	32	Size of each training batch
Epochs	50	Maximum training iterations
Optimizer	Adam	With cosine decay
CNN Filters	64	Used in Conv1D layers
LSTM Units	100,50	Two stacked LSTM layers
Informer layers	2	Self-attention encoder layers
Dropout Rate	0.1	Regularization to prevent overfitting
Learning Rate	0.001	Tuned for best performance

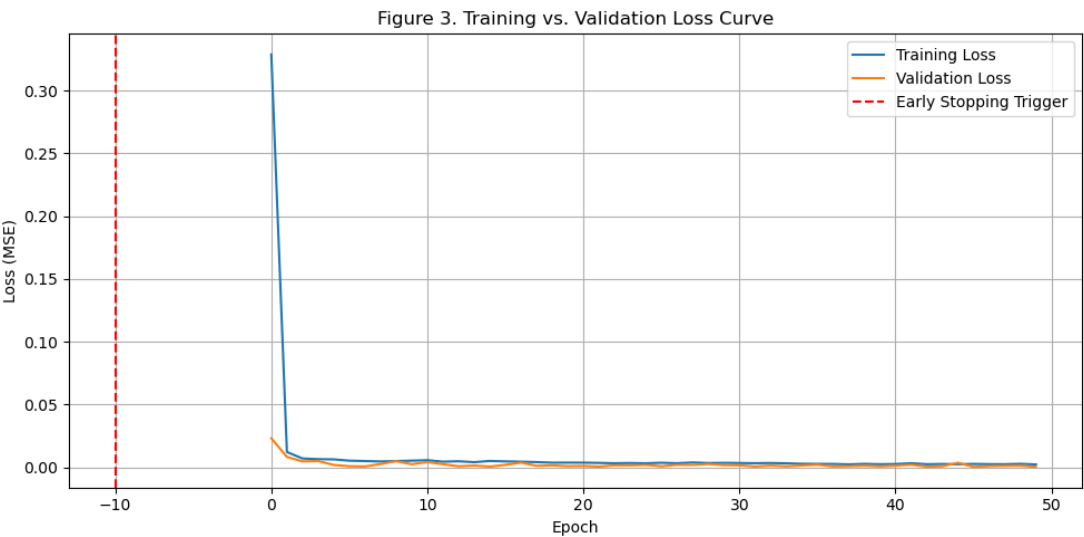


Figure 3 Training and validation loss curves over 50 epochs, showing rapid convergence, no overfitting, and strong generalization.

4.2 Performance Metrics

We measure model performance using the following metrics:

MSE: It calculates an average of differences-squared between predicted and actual values.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

Mean Absolute Error (MAE): It represents the average value of the magnitudes of the errors.

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

R² Score : Ratio of the explained variance to the total variance is known as R² Score.

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

4.3 Results and Experiments

With a learning rate of 1×10^{-3} , batch size = 64, dropout = 0.2, and an Informer encoder (d_model = 64, 2 heads, d_ff = 128), Experiment 1 had two Conv1D layers (32 filters each, kernel = 3) feeding into a single-layer LSTM (50 units, return_sequences=True). This setup produced test-set R² scores of 0.60 (1H), 0.62 (4H), and 0.64 (1D). In Experiment 2, we maintained two Conv1D layers but increased their filters to 64 each (kernel = 3), extended the LSTM branch to two layers (100 → 50 units, both return_sequences=True), and adjusted the learning rate to 5×10^{-4} , batch size to 32, and dropout to 0.1; the Informer was scaled to d_model = 128, 4 heads, d_ff = 256, yielding R² = 0.78 (1H), 0.80 (4H), and 0.82 (1D). The design in Experiment 3 was two Conv1D layers (64 filters) and two LSTM layers (100 → 50 units), but dropout was raised to 0.15, L2 weight decay (1×10^{-4}) was used on all kernels, the input window was stretched from 20 to 40 time steps, and early stopping (patience = 5) was added (with Informer settings unaltered), therefore improving R² to 0.82 (1H), 0.85 (4H), and 0.88 (1D).

At last in Experiment 4, the model once again utilized two Conv1D layers (64 filters), two

LSTM layers (100 50 units), and two Informer encoder layers ($d_{\text{model}} = 128$, heads = 4, $d_{\text{ff}} = 256$) but the optimizer was changed to AdamW (weight decay = 2.2×10^{-4}) with a cosine learning-rate decay schedule. Near-perfect R^2 values of 0.9988 (1H), 0.9855 (4H), and 0.9950 (1D) were attained with this completely integrated setup. Beyond high R^2 values, as shown in Table 3, the fourth experiment also produced exceptionally low mean squared error (MSE) and mean absolute error (MAE) readings across all time periods.

Table 3 . Performance metrics across all time periods, with the fourth experiment achieving notably low errors and high predictive accuracy.

Timeframe	MSE	MAE	R^2 Score
1H	0.00007480	0.0075	0.9988
4H	0.000961	0.0188	0.9855
1D	0.000506	0.0177	0.983

To ensure these high scores do not result from overfitting, we compared training and test metrics directly. As shown in Figure 4, the R^2 scores for training (0.984) and testing (0.983) in the 1D timeframe are nearly identical, indicating excellent generalization. Similarly, Figure 5 and Figure 6 shows that MSE and MAE values across the train and test sets are very close, further supporting the model's robustness. These results confirm that the model does not memorize patterns but generalizes well to unseen market data from October to December.

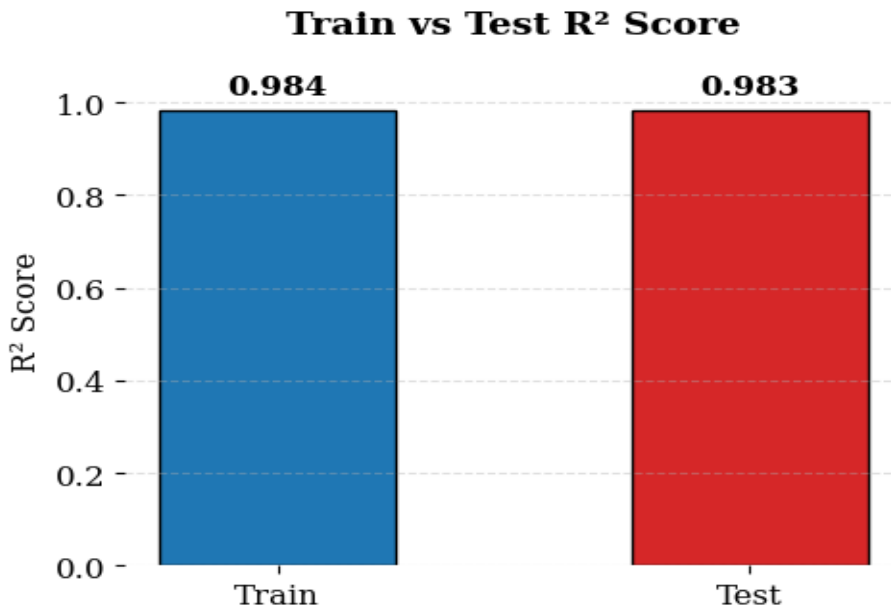


Figure 4 R^2 scores in the 1D timeframe indicate excellent generalization

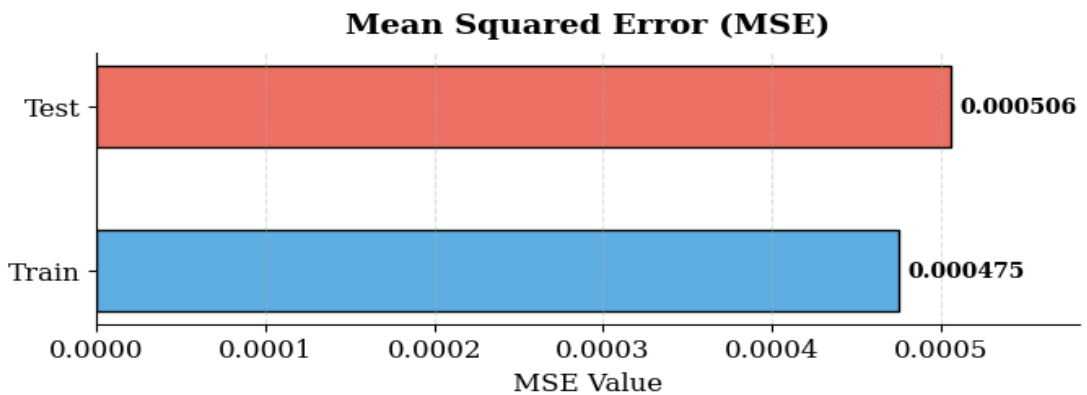


Figure 5 . MSE values in the 1D timeframe show minimal difference between training and testing sets

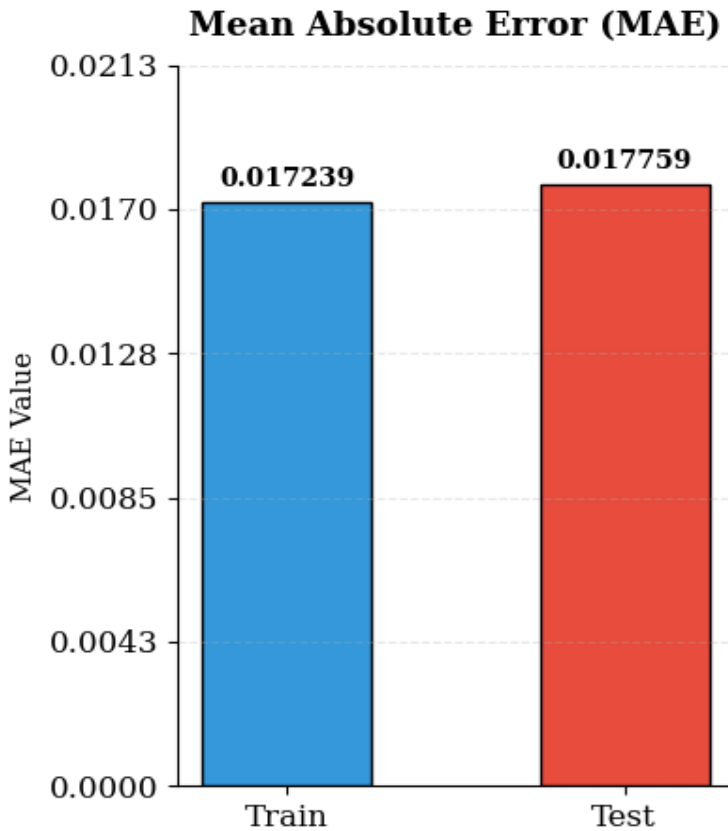


Figure 6 MAE results in the 1D timeframe are closely matched for training and testing

5. Future work and discussion

The proposed hybrid CNN–LSTM–Informer model demonstrated exceptional performance across multiple timeframes in forecasting gold prices, significantly outperforming traditional and state-of-the-art deep learning models. Despite these promising results, several areas remain open for further exploration and improvement to enhance the model’s adaptability, interpretability, and scalability in real-world financial applications.

5.1 Future work

Building on the strengths of the proposed hybrid CNN–LSTM–Informer framework, future research can enhance the model's contextual awareness by incorporating heterogeneous data sources such as macroeconomic indicators (e.g., interest rates, unemployment rates, inflation, central bank announcements) and sentiment signals derived from financial news or social media. To integrate these diverse data types into the current architecture, one promising approach is to expand the input feature space by feeding the macroeconomic variables as additional time-aligned inputs to the CNN layer, allowing the convolutional filters to jointly extract localized temporal patterns from both technical and macroeconomic features.

In parallel, textual sentiment data—processed through pre-trained NLP models such as BERT or FinBERT—can be transformed into fixed-length embeddings. These embeddings can then be fused into the model either via concatenation at the LSTM input stage (for sequence modeling alongside price features) or as auxiliary contextual vectors injected into the Informer encoder's attention mechanism, where they can guide the attention distribution based on real-time market sentiment.

However, integrating such heterogeneous sources presents key challenges. Chief among them is the frequency mismatch between high-resolution gold price data (e.g., 1-hour intervals) and lower-frequency macroeconomic indicators (e.g., monthly or quarterly releases). To mitigate this, forward-filling or interpolation techniques can be used, though care must be taken to avoid information leakage. Alternatively, multi-scale architectures or attention-based temporal alignment (e.g., time-aware attention) may help harmonize feature representations across differing time granularities.

Beyond data enrichment, future work can explore explainability methods such as SHAP and LIME to enhance transparency and user trust. Online or continual learning variants of the model could enable dynamic adaptation to new data without retraining from scratch, improving real-time performance in volatile markets. Ensemble learning approaches and multi-resolution temporal modeling (e.g., jointly analyzing 1H, 4H, and 1D streams) are also promising directions to capture hierarchical temporal dependencies and improve generalization.

Lastly, while this study focused on gold price forecasting, the proposed architecture is extendable to other commodities (e.g., oil, silver) and digital assets (e.g., Bitcoin), where macroeconomic drivers and investor sentiment similarly influence price dynamics. Cross-asset experimentation would help validate the generalizability and scalability of the hybrid approach.

5.2 Discussion

The findings imply that gold price prediction may be improved by combining CNN for local trend patterns, LSTM for sequential relationships, and the Informer mechanism for long-term worldwide attention. On high-frequency (1H) data, where volatility and noise frequently compromise performance, the model was particularly successful. We used a seasonal data-splitting technique that divides data by different years and months to make sure that it is robust under all types of market conditions. This strategy considers changes in market phases including structural changes, volatility clusters, and trend shifts, hence it resembles actual-world forecasting situations. Although conventional k-fold cross-validation is inappropriate for time series because of temporal dependencies, our temporal split validation provides a more realistic and thorough assessment of generalization ability. Performance held steady across all the examined timeframes 1H, 4H, and 1D showing the model's robustness in diverse market circumstances. Mostly underutilized in earlier research, the seasonal partitioning approach developed as a major innovation to help to prevent overfitting and improve generalizability.

Regarding computational efficiency, training the model on each timeframe (1H, 4H, and 1D) took between 40 and 60 minutes on a single NVIDIA RTX 3080 GPU. The inference time per prediction was consistently less than 80 milliseconds, which means the model is good for real-time use in trading or decision-support systems. The memory footprint of the model was less than 12 GB, which made it compatible with mid-range GPU hardware and improved its practical use in resource-limited settings despite its multi-branch hybrid design consisting of two Conv1D layers, a stacked LSTM, and two Informer encoder layers.

Including a variety of technical indicators including RSI, MACD, and Bollinger Bands also improved the model's capacity to capture several market signals linked to momentum, volatility, and overbought/oversold conditions. Nevertheless, it's crucial to understand that the performance of a model might still depend on the quality and representativeness of the past data. Ongoing challenges to predictive stability are market dynamics affected by regime changes, black swan occurrences, or macroeconomic shocks.

References:

- Andriani, L., Kesumah, F., & Firdaus, L. (2023). Causality test on gold prices and economic risk (VAR model application). *Asian Journal of Economics and Business Management*.
<https://api.semanticscholar.org/CorpusId:269967746>
- Apergis, N., Cooray, A., Khraief, N., & Apergis, I. (2019). Do gold prices respond to real interest rates? Evidence from the Bayesian Markov Switching VECM model. *Journal of International Financial Markets, Institutions and Money*.
<https://doi.org/10.1016/J.INTFIN.2018.12.014>
- Botunac, I., Bosna, J., & Matetić, M. (2024). Optimization of Traditional Stock Market Strategies Using the LSTM Hybrid Approach. *Inf.*, 15, 136.
- Bunnag, T. (2024). The Importance of Gold's Effect on Investment and Predicting the World Gold Price Using the ARIMA and ARIMA-GARCH Model. *Ekonomikalia Journal of Economics*. <https://pdfs.semanticscholar.org/ecf5/0804ff5cdf1e42208fb0a0210c0aca43363b.pdf>
- Chen, S. (2023). Gold futures price forecast research—A combinatorial prediction method based on CEEMDAN-GARCH-SVR. *Highlights in Business, Economics and Management*.
<https://api.semanticscholar.org/CorpusId:261647998>
- Feng, S., Feng, W., Gao, X., Zhao, P., & Shen, Z. (2025). TS-LIF: A Temporal Segment Spiking Neuron Network for Time Series Forecasting. *ArXiv, abs/2503.05108*. <https://api.semanticscholar.org/CorpusId:276884901>
- Gong, W. (2024). Research on gold price forecasting based on Istm and linear regression. *SHS Web of Conferences*. <https://api.semanticscholar.org/CorpusId:267159122>
- Hutagalung, S. V., Yennimar, Y., Rumapea, E. R., Hia, M. J. G., Sembiring, T., & Manday, D. R. (2023). COMPARISON OF SUPPORT VECTOR REGRESSION AND RANDOM FOREST REGRESSION ALGORITHMS ON GOLD PRICE PREDICTIONS. *Jurnal Sistem Informasi Dan*

Ilmu Komputer Prima(JUSIKOM PRIMA). <https://api.semanticscholar.org/CorpusId:263189075>

Kadir, R., Alisyahbana, A. N. Q. A., & Azis, F. (2023). The Effect of Gold Price on Profit Achievement at PT. Pegadaian (Persero) Maros Branch, South Sulawesi. *Fundamental and Applied Management Journal*. <https://api.semanticscholar.org/CorpusId:260985378>

Pan, X. (2024). The Comparison Between Random Forest and LSTM Models Based on the Gold Price Prediction. *Advances in Economics, Management and Political Sciences*. <https://api.semanticscholar.org/CorpusId:272160375>

Qiu, C., Zhang, Y., Qian, X., Wu, C., Lou, J., Chen, Y., Xi, Y., Zhang, W., & Gong, Z. (2024). A Two-Stage Deep Fusion Integration Framework Based on Feature Fusion and Residual Correction for Gold Price Forecasting. *IEEE Access*, 12, 85565–85579.

Qiu, T., Sun, X., Lu, W., & Shi, W. (2022). International Gold Price Forecast Based on CEEMDAN and Support Vector Regression with Grey Wolf Algorithm. *Complex.*, 2022, 1511479:1-1511479:12.

Vidal, A., & Kristjanpoller, W. D. (2020). Gold volatility prediction using a CNN-LSTM approach. *Expert Syst. Appl.*, 157, 113481.

Wang, H., Dong, X., Qu, H., Liao, J., & Ma, D. (2024). Gold Stock Price Forecasting Based on Nonlinear Weighted Particle Swarm (IPSO) Optimised Support Vector Machine (SVM) Time Series. *Advances in Economics, Management and Political Sciences*. <https://api.semanticscholar.org/CorpusId:270081831>

Xiong, H., & Zhang, W. (2024). How the US Macroeconomic Factors Affect the Gold Price? *Advances in Economics, Management and Political Sciences*. <https://api.semanticscholar.org/CorpusId:274632056>

Yanto, M., Sanjaya, S., Yulismi, Y., Guswandi, D., & Arlis, S. (2021). Implementation multiple linear regression in neural network predict gold price. *Indonesian Journal of Electrical Engineering and Computer Science*. <https://api.semanticscholar.org/CorpusId:237382549>

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., & Zhang, W. (2020). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *ArXiv, abs/2012.07436*. <https://arxiv.org/pdf/2012.07436.pdf>

پیشبینی نرخى زیر به به کارهینانى مۆدیلی هاوبه شى-CNN-LSTM-Informer به داتای چەند ئاستى

پوخته:

زیر به شیوهیه کی به فراوان وهک یه کیک له گرنگترین سه رماییه داراییه کان داده نریت، بۆیه نه م تووژینه وهیه کار له سه ر پیشبینیکردنی نرخى زیر ده کات. که مۆدیلیکی هاوبه شى فیروونی قول به به کارهینانى نه لگۆریتمه کانى CNN و شیکردنه وهی شیوازه ناوخۆیه کان، LSTM بۆ وه رگرتنى په یوه ندییه زنجیرییه کان و Informer بۆ وه رگرتنى په یوه ندییه دریز خانه یه کان. له 3 تایمفره یمی جیاوازا. که داتا کان له پلاتفورمی میتاتریده ر کۆراونه ته وه. که تیایدا نیشاندهره ته کنیکه کانى RSI, MACD, BB و چەندانی تر وه کو شیکارییه تاییه تمه ندییه کان به کارهاتوون. وه ههروه ها ستراتیجی دابه شکردنی وه رزی بۆ داتاسیته که به کارهاتووه بۆ زیادکردنی گشتگیری که مکردنه وهی Overfitting. نه نجامه کان نیشان دده دن که مۆدیل په پیشنیار کراوه که به شیوه یه کی به رچاو سه رکه وتوو. به به ده سه تهینانى ریژه ی 0.9988 بۆ تایمفره یمی 1H، ریژه ی 0.9855 بۆ تایمفره یمی 4H ههروه ها ریژه ی 0.9950 بۆ تایمفره یمی 1D. له رووی کردارییه وه ده کریت وه کو ئامرازیکى کاریگەر بۆ چاودیری راسته وخوی بازار و دارشتنى ستراتیجیه وه به رهینه رییه کان به کار نه ییریت به تاییه تی له بارودوخه ئالۆزه کان که وردی پیشبینی گرنگه.

مع مدخلات CNN-LSTM-Informer تنبؤ سعر الذهب باستخدام نموذج هجين من متعددة المستويات

الملخص:

يُعتبر الذهب من أكثر الأصول الاستثمارية قيمة وأهمية، ولذلك فإن هذا البحث يركّز على التنبؤ بأسعاره باستخدام نموذج هجين من تقنيات التعلم العميق. يعتمد النموذج على دمج خوارزميات CNN لاستخلاص الأنماط المكانية، و LSTM لاستخلاص العلاقات التسلسلية، و Informer لالتقاط الترابطات طويلة المدى. تم تنفيذ النموذج على ثلاث أطر زمنية مختلفة، حيث تم جمع البيانات من منصة MetaTrader، مع استخدام مؤشرات فنية متعددة مثل RSI، و MACD، و Bollinger Bands، بالإضافة إلى خصائص تقنية أخرى. كما تم تطبيق استراتيجية التقسيم الموسمي للبيانات بهدف تقليل مشكلة الإفراط في التعلم (Overfitting) وزيادة قدرة التعميم. تشير النتائج إلى أن النموذج المقترح حقق أداءً ممتازاً، حيث بلغ معامل التحديد R^2 قيمة 0.9988 للإطار الزمني 1 ساعة، و 0.9855 للإطار الزمني 4 ساعات، و 0.9950 للإطار الزمني اليومي. من الناحية التطبيقية، يمكن اعتبار هذا النموذج أداة فعالة لمراقبة السوق بشكل مباشر، وصياغة استراتيجيات استثمارية دقيقة، خاصة في الأوقات التي تتسم بعدم الاستقرار والتقلبات، حيث تكون دقة التنبؤ ذات أهمية كبيرة.