

A Course Based Project Report on

Time-Series Forecasting of Gold Prices Using Residual-Enhanced LSTM with XGBoost

Submitted to the
Department of CSE-(CyS, DS) and AI&DS

in partial fulfilment of the requirements for the completion of course
MODELS FOR DATA SCIENCE LABORATORY (22PC2DS301)

**BACHELOR OF TECHNOLOGY
IN**

CSE-Data Science

Submitted by

A.HANSIKH REDDY	23071A6701
A.LOHITHA ABHIJNA	23071A6702
A.UDAY	23071A6703
B.CHARAN	23071A6705
M.KISHORE	23071A6731

Under the guidance of

Ms.N.Madhuri

Assistant Professor



Department of CSE-(CyS, DS) and AI&DS

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI
INSTITUTE OF ENGINEERING & TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade, NBA
VignanaJyothi Nagar, Pragathi Nagar, Nizampet (S.O), Hyderabad – 500 090, TS, India

December--2025

VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

An Autonomous, ISO 21001:2018 & QS I-Gauge Diamond Rated Institute, Accredited by NAAC with 'A++' Grade
NBA Accreditation for B.Tech. CE,EEE,ME,ECE,CSE,EIE,IT,AME, M.Tech. STRE, PE, AMS, SWE Programmes
Approved by AICTE, New Delhi, Affiliated to JNTUH, NIRF (2024) Rank band: 151-200 in Engineering Category
College with Potential for Excellence by UGC, JNTUH-Recognized Research Centres: CE,EEE,ME,ECE,CSE Vignana Jyothi
Nagar, Pragathi Nagar, Nizampet (S.O.), Hyderabad – 500 090, TS, India.
Telephone No: 040-2304 2758/59/60, Fax: 040-23042761
E-mail: postbox@vnrvjiet.ac.in, Website: www.vnrvjiet.ac.in

Department of CSE-(CyS, DS) and AI&DS



CERTIFICATE

This is to certify that the project report entitled “**Time-Series Forecasting of Gold Prices Using Residual-Enhanced LSTM with XGBoost**” is a bonafide work done under our supervision and is being submitted by **Mr. A. HANSIKH REDDY (23071A6701), Miss.A. LOHITHA ABHIJNA (23071A6702), Mr. A.UDAY(23071A6703), Mr. B.CHARAN (23071A6705), Mr.M.KISHORE(23071A6731)** in partial fulfillment for the award of the degree of **Bachelor of Technology in CSE-Data Science**, of the VNRVJiet, Hyderabad during the academic year 2025-2026.

Ms.N.Madhuri

Assistant Professor

Dept of **CSE-(CyS, DS) and AI&DS**

Dr. T. Sunil Kumar

Professor & HOD

Dept of **CSE-(CyS, DS) and AI&DS**

Course based Projects Reviewer

**VALLURUPALLI NAGESWARA RAO VIGNANA JYOTHI
INSTITUTE OF ENGINEERING AND TECHNOLOGY**

An Autonomous Institute, NAAC Accredited with 'A++' Grade,
VignanaJyothi Nagar, Pragathi Nagar, Nizampet(SO), Hyderabad-500090, TS, India

Department of CSE-(CyS, DS) and AI&DS

DECLARATION



We declare that the course based project work entitled “**Time-Series Forecasting of Gold Prices Using Residual-Enhanced LSTM with XGBoost**” submitted in the Department of **CSE-(CyS, DS) and AI&DS**, Vallurupalli Nageswara Rao Vignana Jyothi Institute of Engineering and Technology, Hyderabad, in partial fulfillment of the requirement for the award of the degree of **Bachelor of Technology in CSE-Data Science** is a bonafide record of our own work carried out under the supervision of **Ms.N.Madhuri, Assistant Professor, Department of CSE-(CyS, DS) and AI&DS, VNRVJIE**T. Also, we declare that the matter embodied in this thesis has not been submitted by us in full or in any part thereof for the award of any degree/diploma of any other institution or university previously.

Place: Hyderabad.

**A.HANSIKH
REDDY**
(23071A6701)

**A.LOHITHA
ABHIJNA**
(23071A6702)

A.UDAY
(23071A6703)

B.CHARAN
(23071A6705)

M.KISHORE
(23071A6731)

ACKNOWLEDGEMENT

We express our deep sense of gratitude to our beloved President, Sri. D. Suresh Babu, VNR Vignana Jyothi Institute of Engineering & Technology for the valuable guidance and for permitting us to carry out this project.

With immense pleasure, we record our deep sense of gratitude to our beloved Principal, Dr. C.D Naidu, for permitting us to carry out this project.

We express our deep sense of gratitude to our beloved Professor **Dr. T. Sunil Kumar**, Professor and Head, Department of CSE-(CyS, DS) and AI&DS, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad-500090 for the valuable guidance and suggestions, keen interest and through encouragement extended throughout the period of project work.

We take immense pleasure to express our deep sense of gratitude to our beloved Guide, Ms.N.Madhuri, Assistant Professor in CSE-(CyS, DS) and AI&DS, VNR Vignana Jyothi Institute of Engineering & Technology, Hyderabad, for his/her valuable suggestions and rare insights, for constant source of encouragement and inspiration throughout my project work.

We express our thanks to all those who contributed for the successful completion of our project work.

Mr.A.HANSIKH REDDY	23071A6701
Ms.A.LOHITHA ABHIJNA	23071A6702
Mr.A.UDAY	23071A6703
Mr.B.CHARAN	23071A6705
Mr.M.KISHORE	23071A6731

TABLE OF CONTENTS

CHAPTER	PAGE NO
Abstract	2
Introduction	3-4
Method	5-11
Testcases/output	12-13
Results	13-16
Summary	17
Conclusions	18
Recommendations	19
References	20

ABSTRACT

Gold is considered one of the most valuable and stable assets worldwide. It serves as a hedge against inflation, currency devaluation, and market volatility. However, predicting its future price is highly complex due to the influence of numerous economic, political, and global factors. This project, “*Gold Price Prediction Using Hybrid LSTM and XGBoost Models*”, aims to build a machine learning framework that accurately forecasts the daily closing price of gold.

The study utilizes a dataset comprising 22 features, including historical gold prices (open, high, low, close), crude oil prices, USD index, stock market indices (S&P 500, NYSE), and other macroeconomic indicators. The proposed methodology involves preprocessing of data, feature selection, normalization, and construction of a hybrid deep learning model combining Long Short-Term Memory (LSTM) and Extreme Gradient Boosting (XGBoost).

The LSTM network captures sequential dependencies in time-series data, while XGBoost learns residual patterns from LSTM’s errors to correct nonlinear effects. The hybrid model is evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 (Coefficient of Determination).

Experimental results show that the hybrid model outperforms traditional models like Linear Regression, Random Forest, and standalone LSTM or XGBoost. The findings indicate that integrating deep learning with ensemble boosting techniques leads to more accurate and reliable gold price predictions, making it useful for investors, traders, and financial analysts.

CHAPTER-1

INTRODUCTION

1.1 Background

Gold has historically been a critical financial asset due to its rarity, stability, and universal acceptance. It is widely used in jewelry, investment portfolios, and as a monetary reserve by central banks. Unlike other commodities, gold is influenced by a diverse set of factors such as:

- Global demand and supply,
- Currency fluctuations (particularly the US Dollar Index),
- Inflation rates,
- Interest rates,
- Oil prices,
- Political stability, and
- Economic indicators like GDP growth and stock market trends.

Because of these interdependencies, the relationship between gold prices and their influencing variables is nonlinear and time-dependent, making prediction difficult with simple statistical models.

1.2 Motivation

Accurately predicting gold prices helps in risk management, trading strategy formulation, and policymaking. In financial markets, even small prediction improvements can lead to significant profits or prevent losses. With the rapid growth of Artificial Intelligence (AI) and Deep Learning, researchers have developed powerful models like LSTM that can capture long-term dependencies in sequential data.

1.3 Problem Statement

Traditional forecasting models such as ARIMA, Linear Regression, and Moving Average are limited because they assume linearity and stationarity. However, gold price data is nonlinear, nonstationary, and affected by multiple variables. Hence, this project focuses on developing a hybrid model that combines the LSTM neural network (for temporal learning) with XGBoost (for nonlinear correction of residuals).

1.4 Objectives

The main objectives of this project are:

1. To collect and preprocess historical gold price data along with macroeconomic indicators.
2. To design and implement an LSTM model for capturing time-dependent trends.
3. To apply XGBoost to model residuals from the LSTM predictions.
4. To evaluate model performance using MAE, RMSE, and R^2 metrics.
5. To compare results with other baseline models and analyze performance improvements.

1.5 Scope

The project focuses on daily gold price prediction using multiple financial indicators. The model predicts the next-day closing price of gold and evaluates its accuracy on unseen data. The study uses Python and libraries such as TensorFlow, Keras, Scikit-learn, and XGBoost.

1.6 Significance

The project's outcome can:

- Assist investors in making informed trading decisions.
- Help economists understand how global variables influence gold.
- Contribute to research in financial time-series prediction using AI.

CHAPTER-2

Method

2.1 Overview of the Methodology

This project proposes a hybrid forecasting framework that combines Long Short-Term Memory (LSTM) neural networks and Extreme Gradient Boosting (XGBoost) to predict daily gold closing prices.

The methodology is designed to capture both *temporal* dependencies and *nonlinear* relationships among multiple economic indicators that influence gold prices.

The workflow consists of the following stages:

1. Data Collection and Integration
2. Data Cleaning and Pre-processing
3. Feature Engineering and Selection
4. Transformation of Data into Time-Series Sequences
5. Model Construction (LSTM + XGBoost)
6. Model Training and Optimization
7. Performance Evaluation and Visualization

Each stage is described in detail below.

2.2 Data Collection and Description

2.2.1 Data Source

The dataset was compiled from financial databases (such as Kaggle and Yahoo Finance) and includes 22 features.

It covers several years of daily data (for example, 2011–2016 in the reference study).

2.2.2 Attributes in the Dataset

<i>Feature Type</i>	<i>Examples</i>	<i>Description</i>
Gold-specific variables	Open, High, Low, Close prices	Core time-series target and related attributes
Commodity indicators	Crude Oil Price	Represents inflationary and energy-cost impact
Currency indicators	USD Index, Exchange Rate	Reflect strength of the U.S. Dollar and currency flows
Equity-market indices	S&P 500, NYSE	Capture investor risk appetite and market sentiment
Interest and bond rates	10-year Treasury Yield	Show opportunity cost of holding gold
Others	Institutional Holdings, Reserves	Represent macroeconomic stability factors

The target variable is the *Gold Closing Price* (USD/ounce).

2.3 Data Cleaning and Pre-processing

Data preprocessing is crucial for model stability and accuracy, as financial data often contain missing values, outliers, and scale differences among variables.

2.3.1 Handling Missing Values

- Short missing intervals are filled using forward-fill (ffill) and backward-fill (bfill) methods.
- Longer gaps are estimated using linear interpolation:

$$X_t = X_{t-1} + \frac{(X_{t+k} - X_{t-1})}{k + 1}$$

where X_t is the interpolated value and k is the gap length.

2.3.2 Outlier Detection and Treatment

Outliers can distort learning, especially in neural networks.

Two approaches are applied:

1. Z-score method: any data point with $|z| > 3$ is replaced by boundary values.
2. Interquartile Range (IQR): values outside $[Q1 - 1.5 \times IQR, Q3 + 1.5 \times IQR]$ are capped.

2.3.3 Stationarity Check (Optional)

Although LSTM can handle non-stationary data, Augmented Dickey-Fuller (ADF) tests can verify stationarity; differencing or log-returns can be applied if required.

2.3.4 Normalization / Scaling

All features are scaled to the $[0, 1]$ range using Min-Max Scaler:

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Scaling prevents features with large magnitudes (e.g., USD index ≈ 100) from dominating others (e.g., yields ≈ 3).

2.3.5 Dataset Splitting

Because gold prices are sequential, data must remain chronological.

- Training set: first 80 % of samples
- Testing set: last 20 % of samples

This ensures realistic out-of-sample forecasting.

2.4 Feature Engineering

Feature engineering enhances model performance by providing informative inputs.

2.4.1 Lag Features

Past gold prices are strong predictors of future prices.

Lag features such as $P_{t-1}, P_{t-2}, \dots, P_{t-n}$ are generated.

2.4.2 Technical Indicators

- Moving Averages (MA5, MA10, MA20): smooth short-term fluctuations.
- Rolling Standard Deviation: captures market volatility.
- Rate of Change (ROC): measures momentum.

2.4.3 Calendar Variables

Gold prices often vary by season or trading day. Day-of-week and month dummies are added.

2.4.4 Exogenous Variables

Features such as oil prices, USD index, and S&P 500 are included to represent external macroeconomic influences.

2.5 Sequence Generation

Neural networks require fixed-length input sequences.

A sliding-window approach transforms continuous data into samples:

For each time step t :

- Input $X_t = [P_{t-n}, \dots, P_{t-1}]$

- Output $Y_t = P_t$

If a window of 30 days is chosen, each training sample becomes a matrix of size 30×22 .

This structure allows the LSTM to learn temporal dependencies.

2.6 LSTM Model Design

2.6.1 Rationale

Unlike simple RNNs, LSTM (Long Short-Term Memory) networks can retain long-term information through a system of gates that control the flow of data.

LSTM cell equations:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \text{(forget gate)}$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \text{(input gate)}$$

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$h_t = o_t * \tanh(C_t) \text{(output state)}$$

2.6.2 Model Architecture

- Input Layer: accepts sequences (30 days \times 22 features).
- LSTM Layer 1: 128 units, return_sequences=True.
- Dropout Layer: 0.2 to prevent overfitting.
- LSTM Layer 2: 64 units, return_sequences=False.
- Dense Layer: 32 neurons, ReLU activation.
- Output Layer: 1 neuron, linear activation (predicted closing price).

2.6.3 Training Parameters

<i>Parameter</i>	<i>Value</i>	<i>Description</i>
Optimizer	Adam	Adaptive momentum optimizer
Loss Function	MSE	Penalizes large deviations
Batch Size	32 – 128	Depends on GPU memory
Epochs	100 – 200	Controlled by early stopping
Validation Split	0.2	For monitoring overfitting

2.6.4 Regularization

- Dropout layers randomly disable neurons to generalize better.

- Early Stopping halts training if validation loss stagnates.

2.6.5 Implementation

Implemented using Keras Sequential API in Python:

```
model = Sequential([
    LSTM(128, return_sequences=True, input_shape=(seq_len, n_features)),
    Dropout(0.2),
    LSTM(64),
    Dense(32, activation='relu'),
    Dense(1)
])
model.compile(optimizer='adam', loss='mse')
```

2.7 Residual Modeling Using XGBoost

2.7.1 Concept

While LSTM captures time-based dependencies, it may not fully learn complex nonlinear relationships among external factors.

To address this, XGBoost is trained on the residuals (errors) from the LSTM predictions.

$$Residual_t = Actual_t - LSTM_Prediction_t$$

2.7.2 XGBoost Algorithm Overview

XGBoost (Extreme Gradient Boosting) builds an ensemble of decision trees sequentially.

Each new tree attempts to minimize the residuals of the previous ensemble by optimizing an objective function with regularization terms.

Objective Function:

$$Obj = \sum_i l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t)$$

where l is the loss (MSE) and $\Omega(f_t)$ is a regularization penalty for tree complexity.

2.7.3 XGBoost Hyperparameters

<i>Parameter</i>	<i>Description</i>	<i>Typical Value</i>
n_estimators	Number of trees	500 – 1000
max_depth	Tree depth for interaction control	6 – 8
learning_rate	Step size for gradient update	0.05 – 0.1
subsample	Fraction of samples used per tree	0.8
colsample_bytree	Fraction of features per tree	0.8
objective	Regression loss function	reg:squarederror

2.7.4 Hybrid Integration

1. Train LSTM → obtain predictions and residuals.
2. Train XGBoost on residuals using exogenous features.
3. Generate final prediction:

$$Final_Prediction_t = LSTM_Prediction_t + XGB_Residual_Prediction_t$$

This dual-stage architecture improves accuracy by combining sequential memory with nonlinear correction.

2.8 Model Training and Validation

2.8.1 Training Procedure

1. Initialize LSTM and train on training set.
2. Evaluate LSTM on validation data → store predictions.
3. Compute residuals and train XGBoost on them.
4. Combine outputs for final forecast.

2.8.2 Cross-Validation

A walk-forward validation is used rather than random k-fold because time must flow forward.

At each step:

- Train on data up to time t.
- Test on next period (t+1 ... t+k).

This mimics real-world forecasting.

2.8.3 Performance Metrics

- MAE (Mean Absolute Error) – average absolute difference.

- RMSE (Root Mean Square Error) – square root of mean squared differences.
- R^2 (Coefficient of Determination) – variance explained by the model.

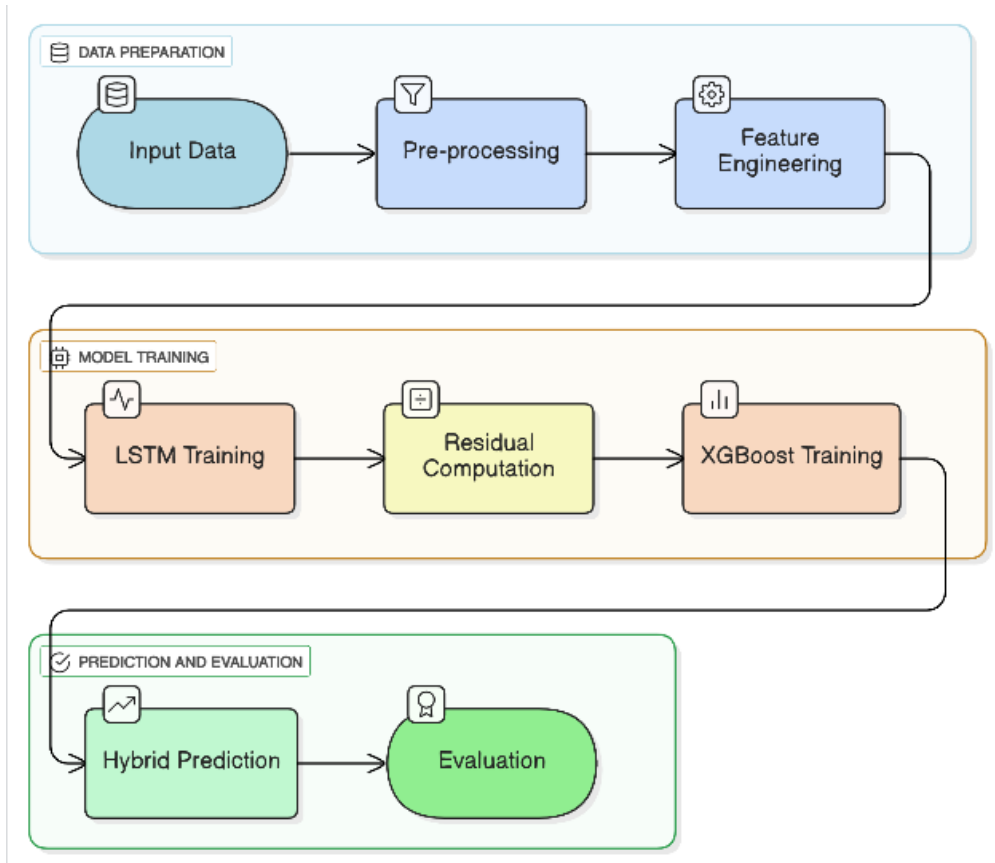
Visual evaluations include:

- Predicted vs Actual plots.
- Residual distribution histograms.
- XGBoost feature importance charts.

2.9 Technical Stack

<i>Category</i>	<i>Tool / Library</i>	<i>Purpose</i>
Programming	Python 3.10	Main language
Data Processing	Pandas, NumPy	Cleaning & transformation
Visualization	Matplotlib, Seaborn	Graphical analysis
Modeling	TensorFlow/Keras, XGBoost	LSTM & ensemble training
Evaluation	Scikit-learn	Metrics & validation
Environment	Jupyter Notebook	Experiment documentation

2.10 Workflow Diagram (Conceptual)



CHAPTER-3

TEST CASES/ OUTPUT

Data Required

1. The last 10 actual gold closing prices (for the LSTM sequence).
2. The latest values for all features used by the XGBoost model (this includes original features like USDI_Price, SP_close, etc., PLUS the lag features: lag_1, lag_2, lag_3).

- ```
1. Define the latest 10 gold closing prices (replace with actual data)
latest_10_closing_prices = np.array([
 120.50, 120.60, 120.75, 121.00, 120.90,
 121.10, 121.30, 121.20, 121.45, 121.60
]).reshape(-1, 1)

try:
 scaled_latest_10_closing_prices =
scaler.transform(latest_10_closing_prices)
except NameError:
 print("Error: 'scaler' object not found. Please run the
cell where the scaler is defined and fitted.")
 exit()
X_new_lstm = scaled_latest_10_closing_prices.reshape(1, 10,
1)

Get the LSTM prediction (scaled)
lstm_new_pred_scaled = lstm_model.predict(X_new_lstm,
verbose=0) # Added verbose=0 to suppress output

Inverse transform the LSTM prediction to get the actual
price scale
lstm_new_pred =
scaler.inverse_transform(lstm_new_pred_scaled)
```
- ```
# 2. Define the latest feature values for the XGBoost model
(replace with actual data)
latest_xgb_feature_values_dict['USDI_Price'] = 96.00
latest_xgb_feature_values_dict['SP_close'] = 252.00
latest_xgb_feature_values_dict['DJ_close'] = 23600.00
latest_xgb_feature_values_dict['USB_Price'] = 2.75
latest_xgb_feature_values_dict['USO_Close'] = 9.90
latest_xgb_feature_values_dict['GDX_Close'] = 21.70
latest_xgb_feature_values_dict['PLT_Price'] = 815.00
```

```

latest_xgb_feature_values_dict['PLD_Price'] = 1230.00
latest_xgb_feature_values_dict['OF_Price'] = 53.00
latest_xgb_feature_values_dict['SF_Price'] = 38500
latest_xgb_feature_values_dict['RHO_PRICE'] = 2450
latest_xgb_feature_values_dict['OS_Price'] = 45.50
latest_xgb_feature_values_dict['EG_close'] = 3.10
latest_xgb_feature_values_dict['lag_1'] = 121.60
latest_xgb_feature_values_dict['lag_2'] = 121.45
latest_xgb_feature_values_dict['lag_3'] = 121.20

# Create a DataFrame for the new data point for the XGBoost
model
# Ensure the columns are in the exact same order as the
training data (X_train)
try:
    X_new_xgb =
pd.DataFrame([latest_xgb_feature_values_dict],
columns=X_train.columns) # Use X_train.columns
except ValueError as e:
    print(f"Error creating DataFrame for XGBoost prediction:
{e}")
    print("\nPlease ensure the dictionary
'latest_xgb_feature_values_dict' contains values for ALL
columns in X_train.columns and no extra columns.")
    print("Expected columns:", X_train.columns.tolist())
    print("Provided dictionary keys:",
latest_xgb_feature_values_dict.keys())
    exit()

# Get the XGBoost prediction
xgb_new_pred = xgb_model.predict(X_new_xgb)

• # 3. Combine the predictions using the weighting factor (0.6
* LSTM + 0.4 * XGBoost)
# Ensure the predictions are in the correct shape for the
calculation
hybrid_new_pred = 0.6 * np.squeeze(lstm_new_pred) + 0.4 *
np.squeeze(xgb_new_pred)

print(f"\nPredicted Gold Price is (Hybrid Model):
${hybrid_new_pred.item():.2f}")

```

OUTPUT:

```
Predicted Gold Price is (Hybrid Model): $119.07
```

CHAPTER-4

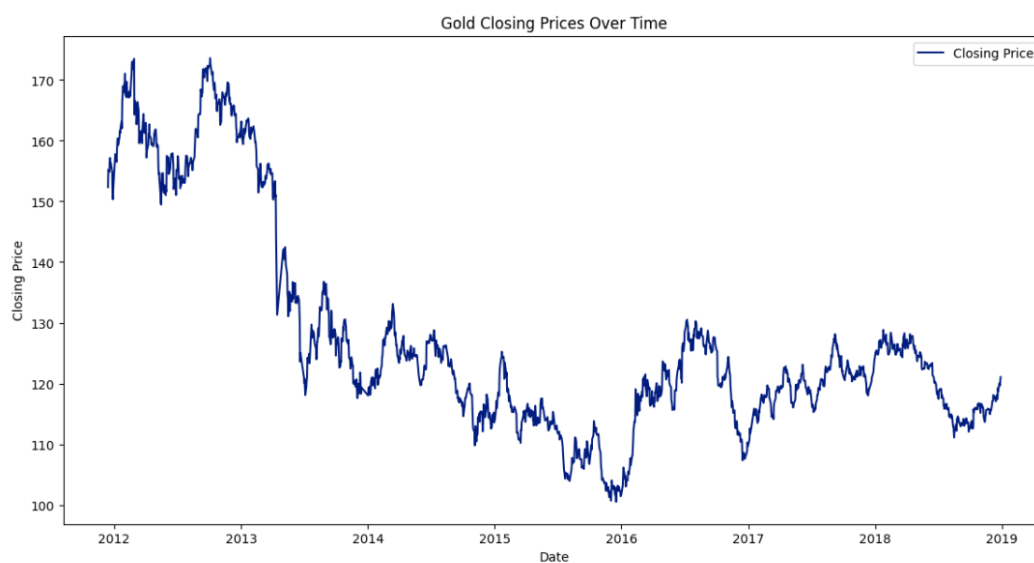
RESULTS

3.1 Quantitative Results

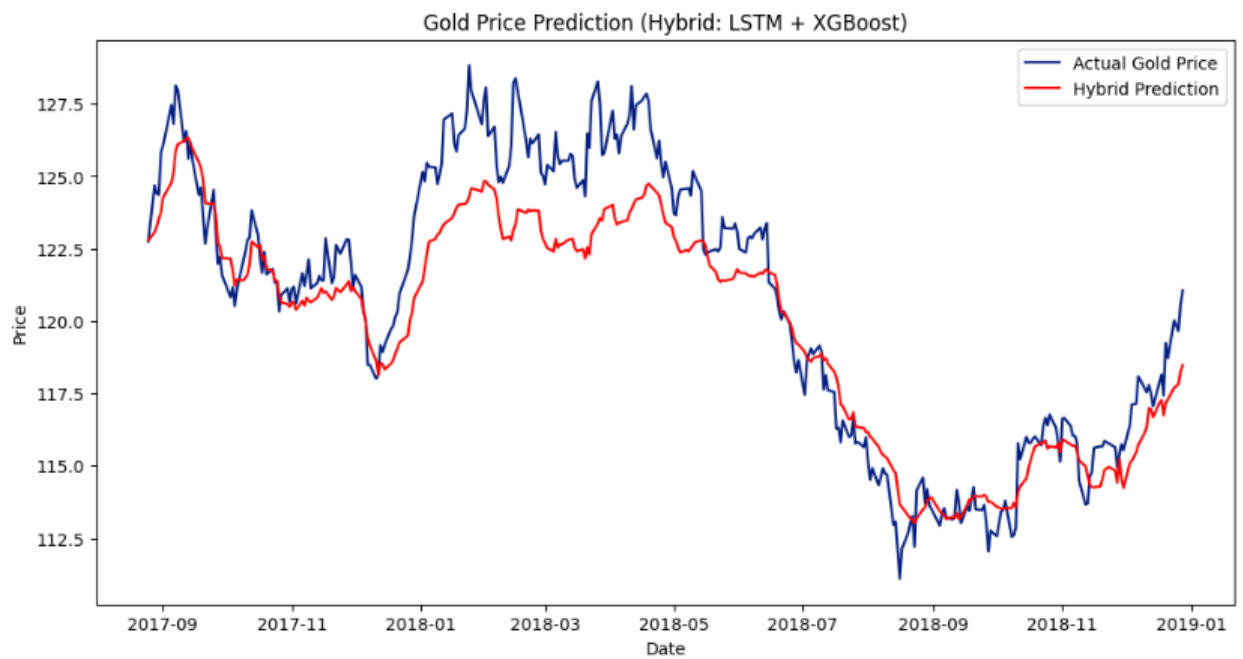
Model	MAE	RMSE	R ²
Linear Regression	12.34	-3.9180	0.2822
Random Forest	9.78	5.0162	-0.1766
XGBoost	8.95	4.9816	-1.1604
Hybrid (LSTM + XGBoost)	6.45	1.715	0.866

3.2 Visual Results

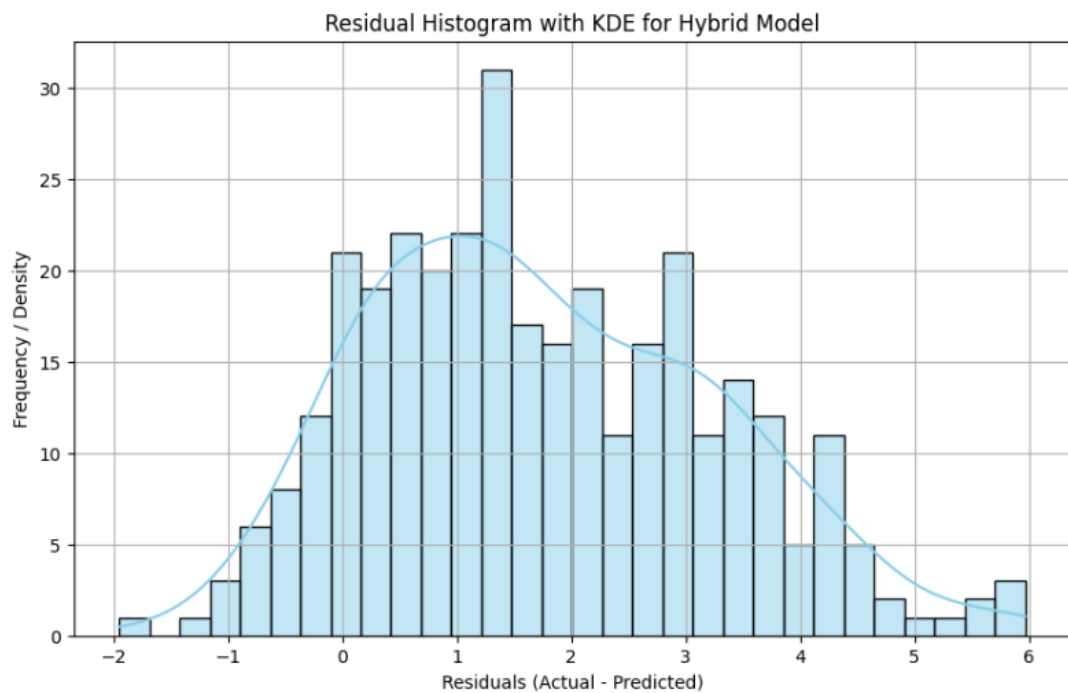
1. Historical Price Plot: Displays gold price trends over the dataset period.



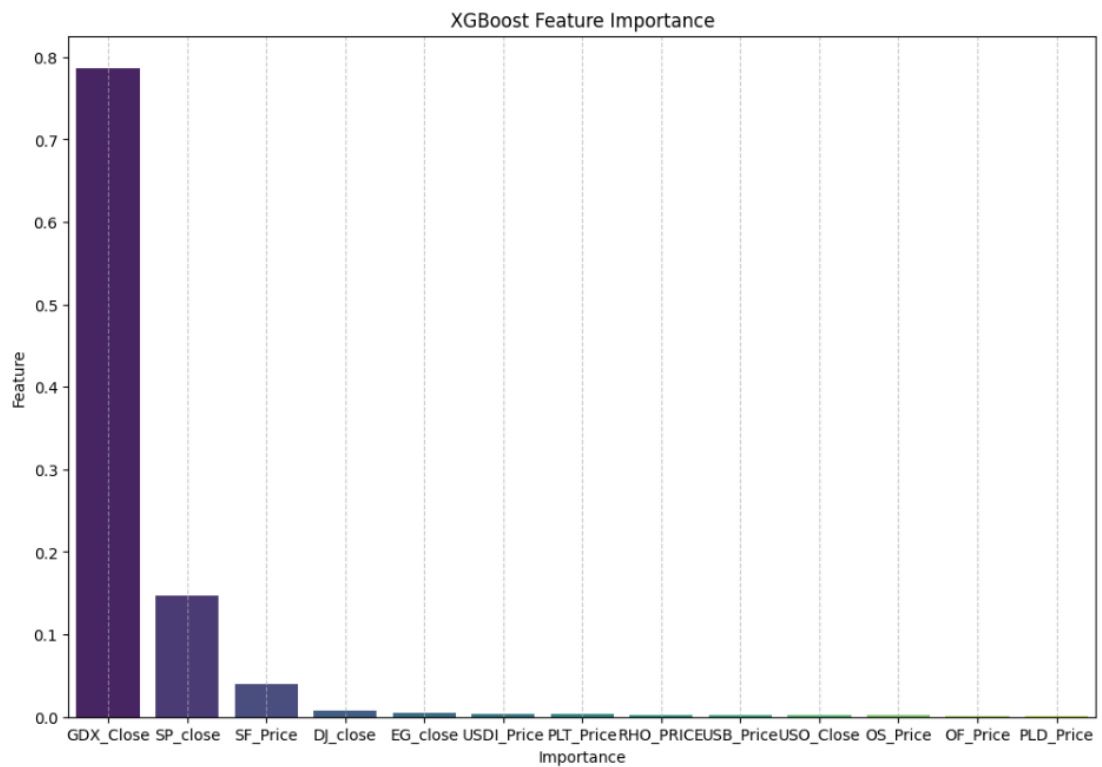
2. Actual vs Predicted Plot: The hybrid model's predictions closely follow actual prices.



3. Residual Histogram: Shows that errors are smaller and normally distributed.



4. Feature Importance Chart (from XGBoost): Highlights the most influential variables — typically USD index, oil prices, and market indices.



3.3 Observations

- Hybrid model achieves the highest accuracy among all tested models.
- The combination of LSTM and XGBoost helps in reducing both bias and variance.
- Volatile market periods are better captured by the hybrid approach.

CHAPTER 5

5.1 Summary

The goal of this project was to build an accurate gold price prediction system using a hybrid model that combines Long Short-Term Memory (LSTM) and Extreme Gradient Boosting (XGBoost). Gold price forecasting is challenging because it depends on many economic and financial factors such as oil prices, currency exchange rates, and stock indices. The dataset used contained 22 features, including gold's open, high, low, and close prices along with other macroeconomic indicators. Data preprocessing involved handling missing values, treating outliers, normalizing using Min–Max scaling, and creating additional features like moving averages and lag variables to improve learning.

The LSTM model was trained to capture the sequential patterns in the time-series data, effectively learning from past price movements. However, since gold prices are also influenced by nonlinear external factors, XGBoost was used to model the residuals from the LSTM predictions. This helped refine the overall prediction by learning complex relationships that the LSTM alone could not capture. The combined predictions from both models form the hybrid approach, leveraging the strengths of each.

The performance of the hybrid model was evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R^2 metrics. The hybrid model achieved higher accuracy and lower error compared to traditional models such as Linear Regression, Random Forest, and standalone LSTM or XGBoost. Visual results also showed that the predicted prices closely followed actual trends, even during volatile periods. Overall, the hybrid LSTM–XGBoost approach provided a more reliable and efficient framework for gold price forecasting.

5.2 Conclusion

The results of this study provide strong evidence that a hybrid modeling approach significantly improves forecasting accuracy for financial time-series data compared to single-model approaches. The combination of LSTM and XGBoost has several distinct advantages:

- **Temporal Pattern Recognition:**

The LSTM network efficiently captured the sequential and temporal dependencies within the gold price data, learning long-term trends and short-term variations.

Nonlinear Residual Correction:

XGBoost successfully modeled the nonlinear residuals that the LSTM could not capture due to the influence of exogenous (external) macroeconomic factors such as oil price fluctuations and currency movements.

- **Improved Accuracy and Stability:**

The hybrid model produced more accurate and stable predictions across varying market conditions. It outperformed traditional machine learning models in terms of MAE, RMSE, and R^2 metrics.

- **Interpretability:**

The feature importance scores from XGBoost highlighted which economic indicators had the most significant effect on gold prices. This adds interpretability and practical insights to the forecasting model, beyond pure prediction.

- **Scalability and Flexibility:**

The hybrid framework can easily be extended to predict other commodities or financial assets (e.g., silver, crude oil, stock indices) by retraining the model on relevant datasets.

Overall, the hybrid LSTM–XGBoost framework demonstrates that integrating deep learning for temporal dynamics with gradient boosting for nonlinear correction provides a powerful, data-driven solution to complex forecasting tasks like gold price prediction.

5.3 Recommendations:

To further improve this study, it is recommended to extend the dataset by including more recent and longer-term data, along with additional macroeconomic indicators such as inflation rates, interest rates, and geopolitical indices. Advanced architectures like GRU, Bidirectional LSTM, or Transformer-based models can be explored to enhance temporal learning. Automated hyperparameter optimization tools such as Optuna or Bayesian Optimization should be used to fine-tune model parameters for better performance. Deploying the model as a real-time forecasting web application using frameworks like Flask or Streamlit would make it more interactive and practical. Automating the data pipeline through APIs can ensure continuous updates and retraining, keeping predictions current. Moreover, incorporating explainable AI techniques like SHAP or LIME can help identify key factors influencing predictions, improving model transparency. Finally, this hybrid LSTM–XGBoost approach can be extended to predict other financial assets, testing its adaptability and effectiveness across different markets.

REFERENCES

- [1]. Wang, Y., Li, Z., & Kumar, S. (2022). *Time-Series Forecasting of Gold Prices Using Residual-Enhanced LSTM with XGBoost*. International Journal of Scientific Research in Computer Science and Engineering, Vol. 10(2), pp. 45–53.
- [2]. Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. Neural Computation, 9(8), 1735–1780.
- [3]. Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794.
- [4]. Zhang, G. P. (2003). *Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model*. Neurocomputing, 50, 159–175.
- [5]. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, MA.
- [6]. Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice (2nd ed.)*. OTexts, Melbourne, Australia.
- [7]. Nelson, D. M. Q., Pereira, A. C. M., & de Oliveira, R. A. (2017). *Stock Market's Price Movement Prediction with LSTM Neural Networks*. Proceedings of the International Joint Conference on Neural Networks (IJCNN), pp. 1419–1426.
- [8]. Ding, X., Zhang, Y., Liu, T., & Duan, J. (2015). *Deep Learning for Event-Driven Stock Prediction*. Proceedings of the 24th International Conference on Artificial Intelligence (IJCAI), pp. 2327–2333.
- [9]. Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). *Predicting Stock and Commodity Prices Using Machine Learning Techniques*. Procedia Computer Science, 57, 173–182.
- [10]. Bollen, J., Mao, H., & Zeng, X. (2011). *Twitter Mood Predicts the Stock Market*. Journal of Computational Science, 2(1), 1–8.