Gold Price Prediction Using Time Series Forecasting

Nirva Patel
ID-202201071

Riddhi Mistry
ID-202201238

***Abstract*—This study presents a comprehensive analysis of gold price forecasting using both classical time series models and modern deep learning approaches. The historical gold price dataset was collected and preprocessed to ensure data quality. Initial analysis involved baseline models to establish reference performance levels, followed by advanced statistical models. It was observed that the complex patterns in the data were not captured. This lead us to use deep learning models. The results highlight the superior capability of deep learning methods, particularly GRU, in modeling nonlinear dependencies in gold price time series data, suggesting their practical utility for more accurate financial forecasting.**

I. Introduction

This report is about predicting the gold price using the historical data. Before beginning with the prediction, we have performed data preprocessing to eliminate any missing values. First of all, the baseline models were used to set a benchmark that ensured that no models works worse than the maximum rmse computed. The baseline models like statistical mean, mean of last 30, naive seasonal,etc. were used. Futher, Time-Series models like Autoregressive (AR), Moving Average (MA), and their extensions (ARMA, ARIMA, and SARIMA) have long been used for financial prediction tasks. These models are effective in capuring the linear trends but fail to capture the long-term autocorrelation. Hence, these models were ineffective in predicting the long term trends and seasonality. This ineffectiveness lead to deep learning models. Deep learning models like Recurrect Neural Network(RNN), Long-short Term Memory(LSTM) and Grated Recurrent Unit(GRU) were used. These models are designed to learn long term dependencies and prefict accordingly. The evaluation of model(s) is done using RMSE(Root Mean Square Error).

The report highlights the shortcomings of classical models and demonstrates the potential of deep learning for more accurate gold price prediction.

II. Methodology

A. Data Collection

We collected daily gold price data from a historical dataset. It includes the date, open, high, low, and close prices. We used only the closing price for forecasting because it shows the final value of gold each day, making it more reliable and stable than other prices. This helps in better prediction results.

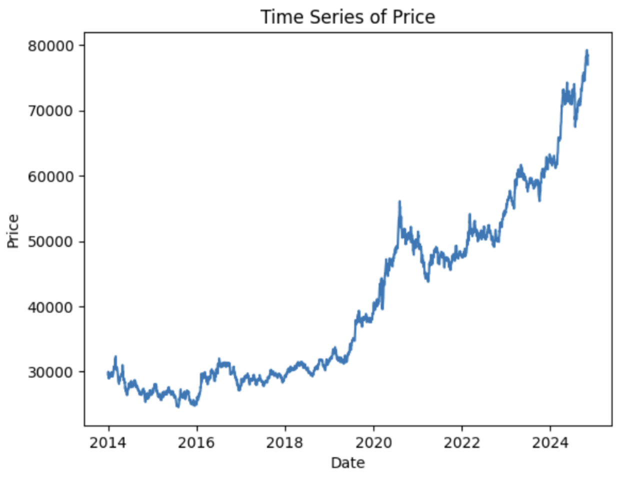

Fig. 1: Gold-Price Dataset

B. Preprocessing

Data was analyzed to understand the patterns within the dataset. Data was cleaned, followed by handeling missing values and ensure the data is consistent for training and modeling.

C. Data Splitting

The data was then split into training and testing sets. Data up to 2024 was used for training, and 2024 onwards for testing and model evaluation.

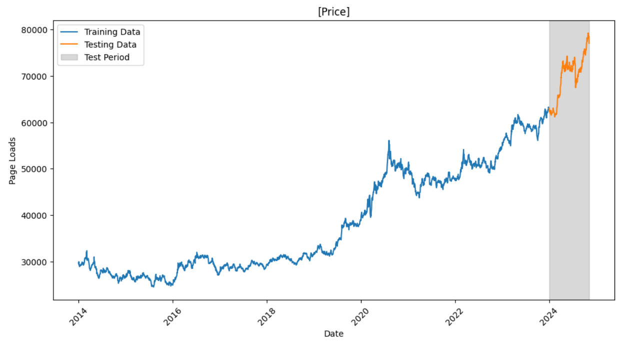

Fig. 2: Train-Test Split of the dataset

D. Baseline Models

Baseline models

Baseline models in time series forecasting are simple reference models used to evaluate more complex methods. If a complex model can't outperform a baseline, it's likely not worth the added complexity.

For initial comparison, several baseline models were applied to the training data:

- Mean: The simplest approach, the forecast for every period is the mean of all past observations.

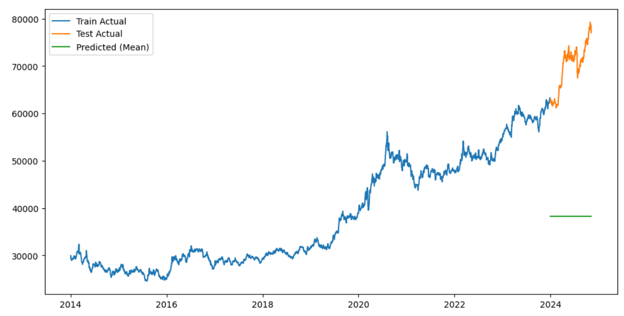

Fig. 3: The predicted Value based on mean

- Mean of Last Thirty Data Points: The forecast is based on the average of the previous five data points.

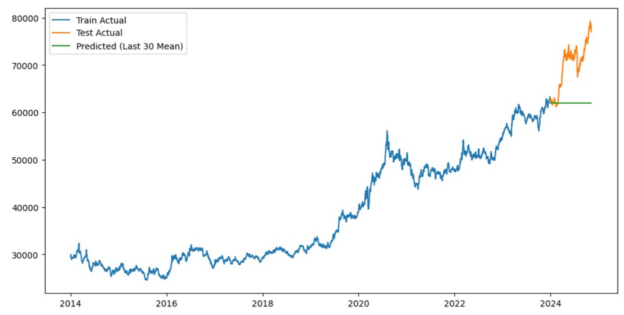

Fig. 4: The predicted Value is based on mean of last 30 data points

- Naive Seasonal Method: This model uses the previous season's value as the forecast for the current season.

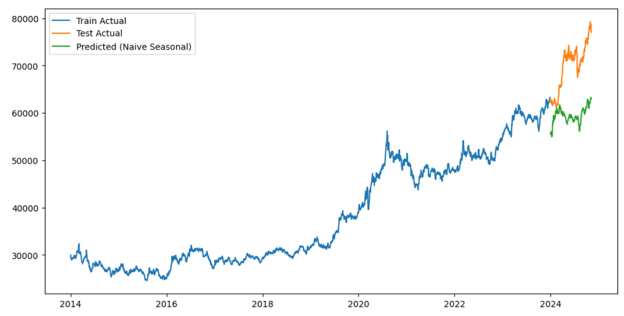

Fig. 5: The predicted Value based on last season's Seasonality

E. Decomposition

The dataset is decomposed into trend, seasonality, and residual to to better understand its underlying patterns and improve forecasting accuracy.

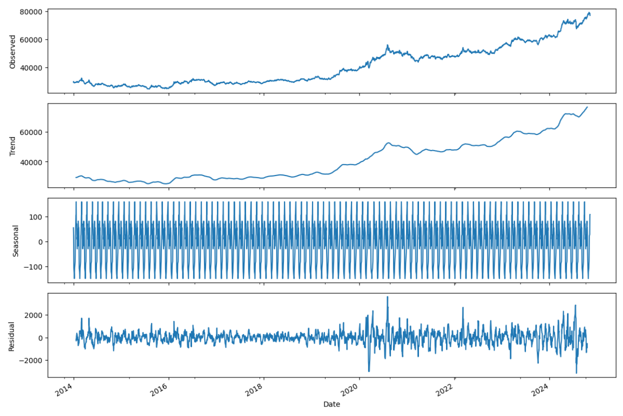

Fig. 6: Decomposition of Gold-Price Dataset

F. Stationarity Check

Stationarity

Stationarity in time series means the data's statistical properties (mean, variance, autocovariance) stay constant over time. It's important because many forecasting models require stationary input for accurate predictions.

The time series data was initially tested for stationarity using the Augmented Dickey-Fuller (ADF) test. The null hypothesis of the ADF test assumes the presence of a unit root (i.e., the time series is non-stationary). The result indicated that the series was non-stationary.

ADF Test

- ADF Test Result
- **ADF Statistic:** -2.166764
- **p-value:** 0.508568
- **Conclusion:** Since the p-value is greater than 0.05, we fail to reject the null hypothesis (H_0). The time series is likely **non-stationary**.

To make the series stationary, we perform transformations. Here we are performing first-order differencing.

G. Differencing

To address non-stationarity, first-order differencing was applied to the series. Post-differencing, the ADF test was conducted again to confirm stationarity. The differenced series exhibited stationarity based on the revised ADF test results.

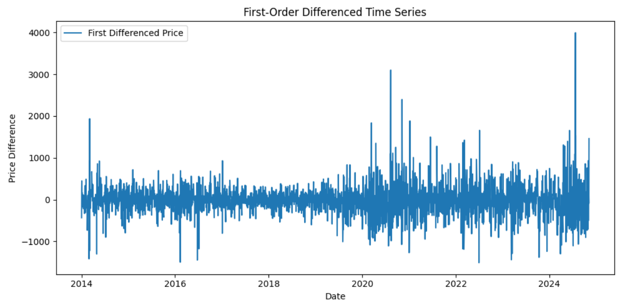

Fig. 7: Differenced Time-Series

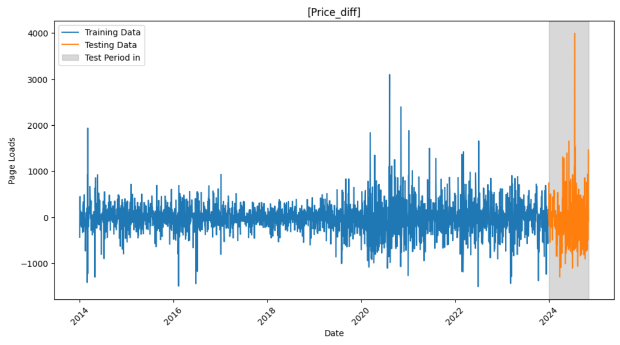

Fig. 8: Train-Test Split of Differenced dataset

H. *Autocorrelation Analysis*

The Autocorrelation Function (ACF) plot was generated for the original series to identify the nature and extent of autocorrelation in the data. This aided in selecting appropriate model parameters.Here, the ACF showed a slow decay, and the PACF cut off after lag 2, suggesting an AR(2) process. Based on this, the ARIMA(2,1,0) model was chosen for forecasting.

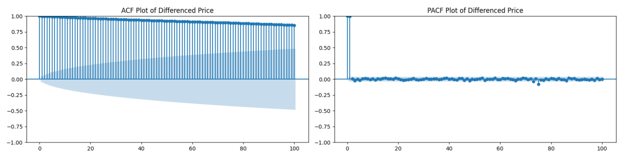

Fig. 9: ACF and PACF

I. *Model Training*

Multiple time series models were trained, including

- Autoregressive Model (AR)
- Moving Average (MA)
- Autoregressive Moving Average (ARMA)
- Autoregressive Integrated Moving Average (ARIMA)
- Seasonal Autoregressive Integrated Moving Average (SARIMA)

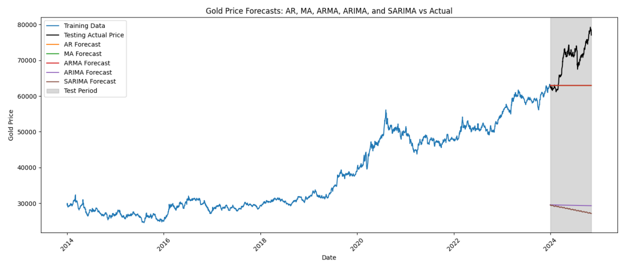

Fig. 10: Gold price forecasts using statistical models—AR, MA, ARMA, ARIMA, and SARIMA—compared to actual gold prices. The shaded region represents the test period where models' predictions are evaluated against real market performance.

Different combinations of parameters (p, d, q) were tested for these models. The selection of optimal values for p and q was decided by the use of the Akaike Information Criterion (AIC), which balances the number of parameters and likelihood. Where p is the order of the autoregressive (AR) model, d is the number of differences needed to make the series stationary, and q is the order of moving average (MA) model.

Akaike Information Criterion (AIC)

The Akaike Information Criterion (AIC) is a measure of the quality of a model in relation to other models. It is used for model selection.
The AIC is a function of the number of parameters k in a model and the maximum value of the likelihood function L:

$$\text{AIC} = 2k - 2\ln(L)$$

The lower the value of the AIC, the better the model. Selecting according to the AIC allows us to keep a balance between the complexity of a model and its goodness of fit to the data.

J. *Model Evaluation*

The performance of the classical time series models was evaluated by plotting the predicted values $(\hat{Y})$ against the actual test values (Y_{test}). The root mean square error (RMSE) was used as the primary evaluation metric to quantify the prediction accuracy of each model. RMSE is preferred because it penalizes large errors more and provides error in the same units as the data. It's differentiable, sensitive to outliers, and widely accepted for evaluating prediction accuracy.

K. *Deep Learning Approaches*

Following the evaluation of classical models, deep learning methods were employed to further enhance forecasting performance. Deep learning models can capture complex, nonlinear patterns that classical models often miss. Deep learning models learn long-term dependencies automatically, making them powerful for sequence forecasting. Overall, they offer higher flexibility and scalability for complex real-world time series problems. The following architectures were implemented:

*1) **RNN**:* A recurrent neural network (RNN) is a type of neural network designed for processing sequential data, like time series. It retains information from earlier inputs by maintaining a form of memory, which it uses to influence future predictions or outputs.

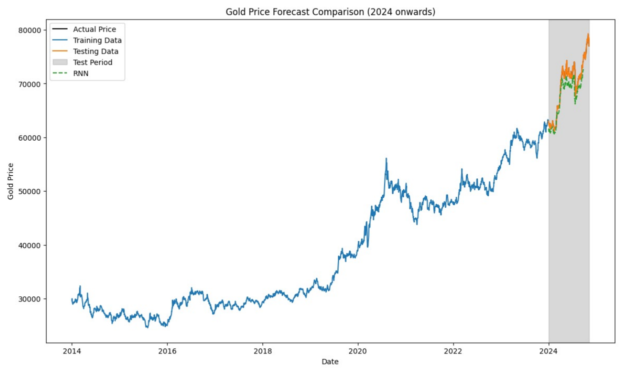

Fig. 11: Forecasting of Gold-Price using RNN

Limitations

- **Vanishing/Exploding Gradient Problem:** During training, gradients can become very small or very large, making learning difficult—especially for long sequences.
- **Short-Term Memory:** RNNs struggle to retain information from earlier time steps in long sequences, making it hard to capture long-term dependencies.
- **Difficulty Handling Long Sequences:** As sequences get longer, performance often degrades because the network can't remember early inputs well.

*2) **LSTM**:* A special type of RNN capable of learning long-term dependencies by using memory cells and gates to control the flow of information.

Gates in LSTM:

- **Forget Gate:** Decides what information from the previous cell state should be discarded.
- **Input Gate:** Determines which new information should be added to the cell state.
- **Output Gate:** Controls what information is sent as output from the current cell state.

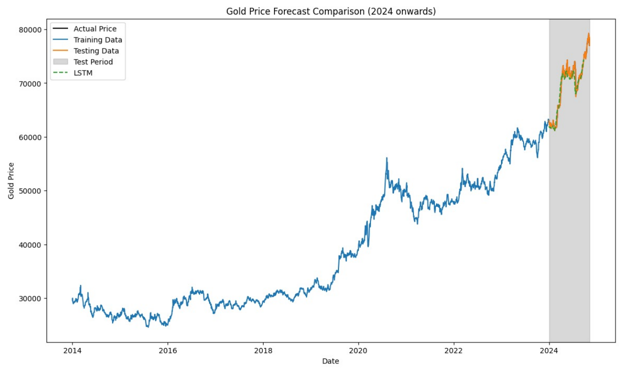

Fig. 12: Forecasting of Gold-Price using LSTM

*3) **GRU**:* A simplified version of LSTM that is computationally efficient while still capturing long-term dependencies. It is suitable for smaller datasets.

- **Update Gate:** Combines the forget and input gates from LSTM into a single gate. It controls how much of the previous state should be passed forward and how much of the new information should be added to the current state.
- **Reset Gate:** Determines how much of the previous memory to forget. It controls how much of the previous hidden state should be used to compute the current hidden state.

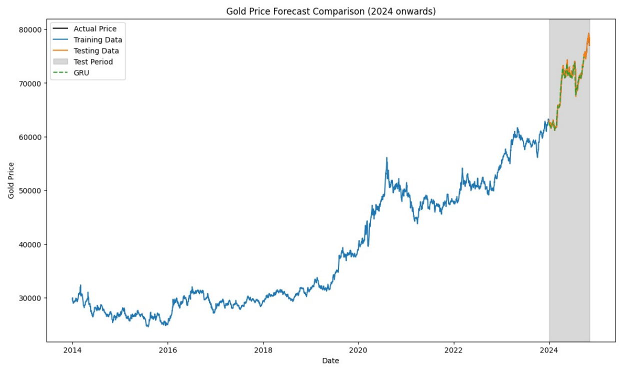

Fig. 13: Forecasting of Gold-Price using GRU

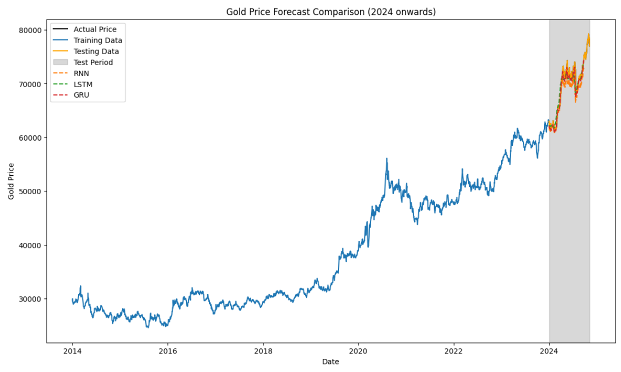

Fig. 14: Prediction of Deep learning models

L. Decision

Each deep learning model was trained and evaluated using the same dataset. Among these, the GRU model produced the best results based on RMSE, demonstrating superior predictive performance compared to the other approaches.

III. Results

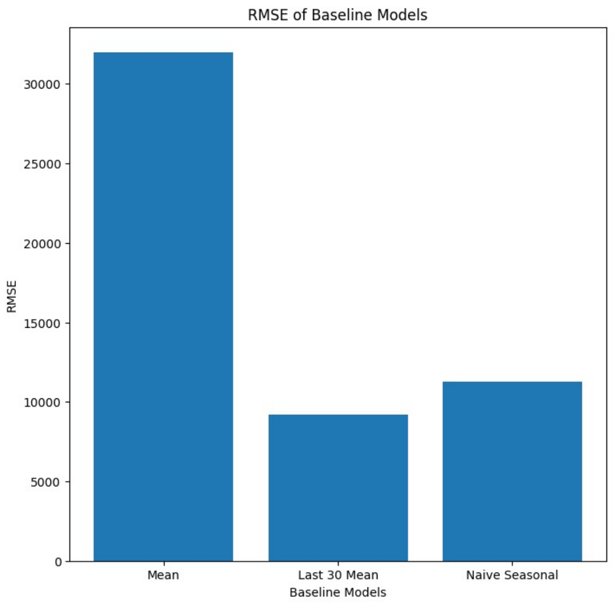

Fig. 15: RMSE of Baseline Models

Model	Name	RMSE
1	Mean	31,979.19
2	Last 30 Mean	40,721.53
3	Naive Seasonal	11,270.45

TABLE I: Model comparison based on RMSE values of Baseline models

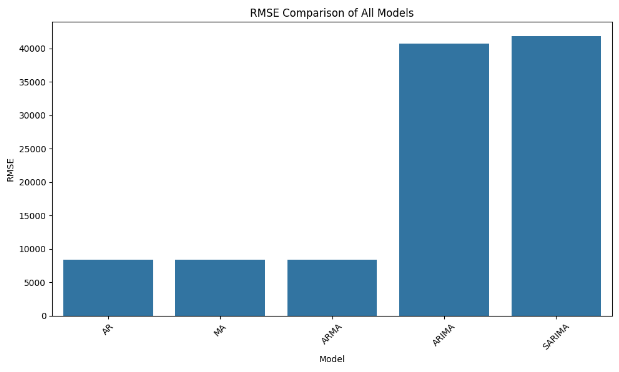

Fig. 16: RMSE of Time-series models

Model No.	Model	RMSE
1	AR	8405.498547
2	MA	8405.100736
3	ARMA	8407.455215
4	ARIMA	40703.601935
5	SARIMA	41866.452980

TABLE II: Model comparison based on RMSE values of Time-Series Models

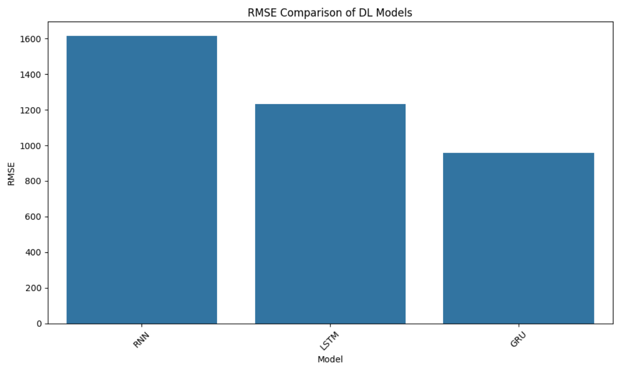

Fig. 17: RMSE of Deep-Learning Models

Model No.	Model	RMSE
1	RNN	1,815.9048
2	LSTM	1,222.5227
3	GRU	780.605

TABLE III: Comparison of Deep Learning Models Based on RMSE

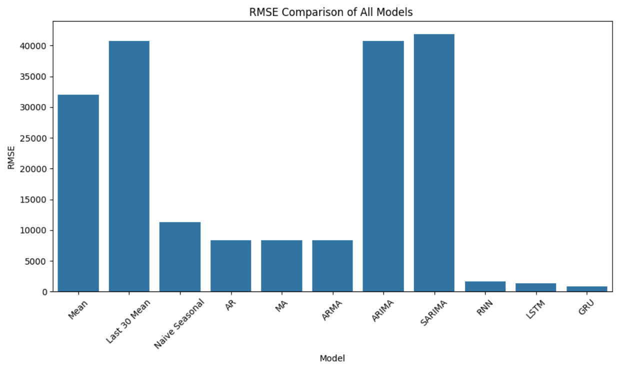

Fig. 18: RMSE comparison across all forecasting models. Deep learning models (RNN, LSTM, GRU) achieved the lowest RMSE, significantly outperforming both statistical models (AR, MA, ARMA, ARIMA, SARIMA) and baseline methods.

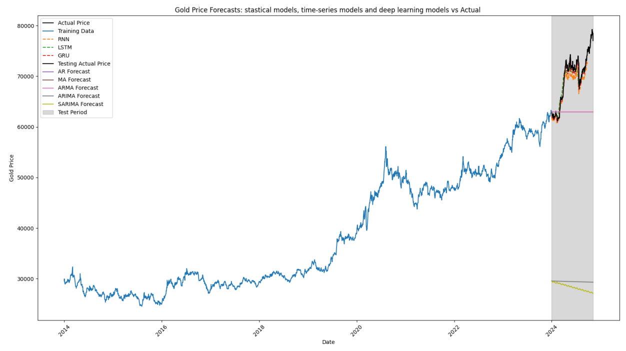

Fig. 19: Comparison of Gold Price Forecasts Using Classical and Deep Learning Models. The plot displays actual gold prices alongside predictions from AR, MA, ARMA, ARIMA, SARIMA, RNN, LSTM, and GRU models. The shaded region highlights the test period, indicating how each model performed on unseen data.

IV. Conclusion

The study shows the comparison between classical models for time series and their limitations, which led to the development of modern deep learning techniques for gold-price prediction. GRU, a deep learning model, effectively captures complex non-linear patterns that classical models fail to account for in forecasting.

References

[1] Marco Peixeiro, *Time Series Forecasting in Python*. Manning Publications, 2022. ISBN: 9781617299889.