

Forecasting Methods in Fintech
096292

Spring 2024

Final Project Exploration Phase

Submitted by:

Bashar Massad

Lana Haj

Kamilya Ziyan

Majd Bishara

Sami Mor

Models

The Models were chosen through trial and error and also using the Pycaret module — a python package that compares many models to find the best performing model for a given dataset.

In total 3 architectures were chosen, and to tune the parameters many experiments were conducted and compared through metrics which we will dive into later.

SARIMA:

For our project, we performed grid-search for the SARIMA comparing AIC and BIC values to find the best model, the following model achieved the best results:

$$\text{SARIMA}(0,1,2)(1,0,1,7)$$

The SARIMA model is an extension of the ARIMA model, specifically designed for time series data with seasonal components. It is suitable for forecasting gold futures because historical gold prices often exhibit trends and patterns influenced by seasonal and cyclical factors, such as economic cycles and market sentiment.

Additionally, we incorporated the (exogenous) variables mentioned before in our modeling.

Prophet:

Prophet is a robust and flexible time series forecasting model developed by Facebook. It is particularly effective for data with strong seasonality, holiday effects, and missing values, making it ideal for gold futures, which are influenced by macroeconomic factors, market volatility, and investor behavior. In our implementation, we enhanced Prophet by adding external regressors such as crude oil prices, silver prices, exchange rates, and the VIX. These exogenous variables helped capture the broader economic and financial conditions that influence gold prices. The **daily frequency** of our data aligns well with Prophet's capabilities, allowing it to automatically handle trends, seasonality, and external factors.

LSTM (Long Short-Term Memory):

LSTM is a type of recurrent neural network (RNN) designed to learn long-term dependencies in sequential data. It is well-suited for forecasting tasks involving time series with complex patterns, as it can learn from both short-term and long-term dependencies. For our project, we trained the LSTM on scaled input data using the past 9 days of information (including exogenous variables such as crude oil prices, stock indices, exchange rates, and interest rates) to predict the gold price on the 10th day.

We chose a model with a hidden size of 256 and 2 layers to ensure it could capture complex temporal patterns and it was trained for 80 epochs on the scaled data using the past 9 days of features (including exogenous variables such as crude oil prices, stock indices, exchange rates, interest rates, and more).

Model Comparison:

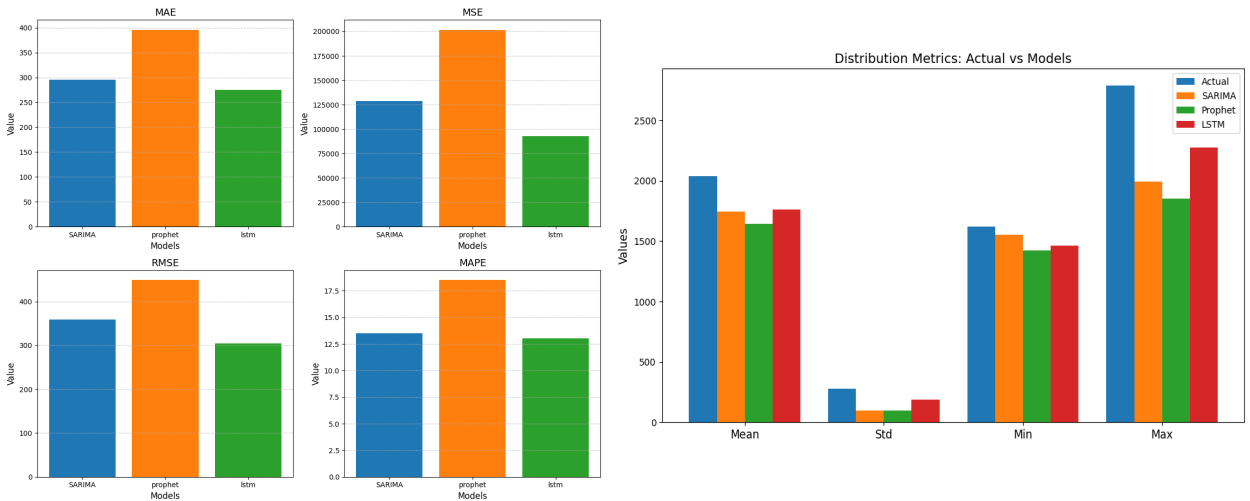
To compare the models, we used several error metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). Additionally, we examined the distributions of the actual and predicted values, including their mean, standard deviation, minimum, and maximum, to assess how well the models captured the behavior of the target variable.

Results:

Error Metrics				
Model	MAE	MSE	RMSE	MAPE
SARIMA	295.35	128,595.96	358.60	13.52%
Prophet	395.81	201,700.21	449.11	18.51%
LSTM	274.79	92,863.23	304.73	13.03%

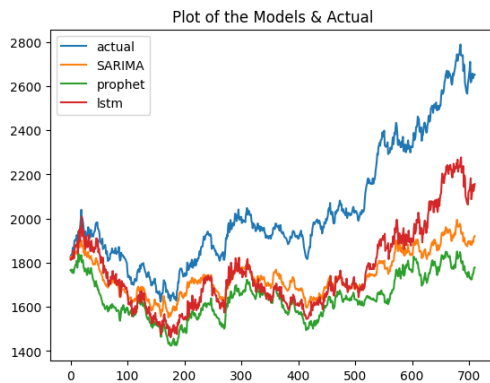
Distribution Metrics				
Statistic	Actual	SARIMA	Prophet	LSTM
Mean	2038.66	1743.32	1642.85	1764.04
Std	280.42	95.71	97.75	186.68
Min	1623.30	1551.76	1424.23	1462.12
Max	2788.50	1994.78	1851.28	2277.17

Visual Comparison:

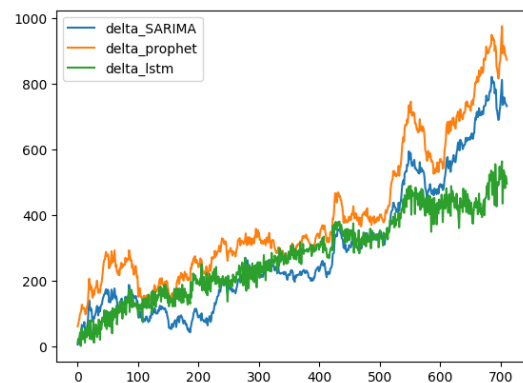


Error Metrics

Distribution Metrics



Plot of models and actual values



Plot of the difference between the actual value and predicted value per model

Error Metric Analysis

The LSTM model outperformed SARIMA and Prophet in all error metrics. Its lower MAE and MAPE indicate it provided predictions closer to the actual values on average, while its lower MSE and RMSE suggest fewer large deviations in its predictions.

Distribution Analysis

1. Mean:

LSTM's mean prediction (1764.04) was closer to the actual mean (2038.66) compared to SARIMA and Prophet, but it still underpredicted on average. SARIMA and Prophet also underpredicted significantly, with Prophet having the lowest mean (1642.85), highlighting its larger bias.

2. Standard Deviation (Std):

LSTM exhibited a standard deviation of 186.68, much closer to the actual value of 280.42 compared to SARIMA (95.71) and Prophet (97.75). This shows that LSTM better captured the variability in the target variable, while SARIMA and Prophet smoothed out the fluctuations too heavily.

3. Min and Max:

LSTM provided a maximum prediction (2277.17) closest to the actual maximum (2788.50), demonstrating its ability to approximate peaks more effectively. Both SARIMA and Prophet underestimated the maximum, with Prophet's maximum being the lowest (1851.28). Similarly, LSTM's minimum (1462.12) was closer to the actual minimum (1623.30) than the other models.

Conclusion

The LSTM model performed the best in terms of both error metrics and capturing the distribution of the target variable. Its ability to handle complex temporal relationships and incorporate exogenous variables enabled it to provide more accurate and dynamic predictions. SARIMA and Prophet, while simpler and faster to implement, struggled to capture the full variability of the data, leading to higher errors and smoothed-out predictions. This analysis suggests that for forecasting gold prices, especially when incorporating exogenous variables, LSTM is the most suitable choice among the tested models.

However, it is important to note that all models showed a decline in performance further into the future (this can be observed in the graph with the deltas).This is likely due to the accumulation of uncertainty and error as forecasts rely more heavily on model assumptions and less on observed data. Temporal dependencies may weaken over longer horizons, and external factors not included in the model may begin to play a larger role, further impacting accuracy. This highlights the inherent challenge in long-term forecasting and underscores the importance of periodic retraining and fine-tuning of the models as new data becomes available.

Reference

The following models were tested using Pycaret:

Model
Auto ARIMA
ARIMA
Light Gradient Boosting w/
Decision Tree w/ Cond. De
Random Forest w/ Cond. I
Extra Trees w/ Cond. Dese
Huber w/ Cond. Deseason
Gradient Boosting w/ Con
K Neighbors w/ Cond. Des
Linear w/ Cond. Deseason
Ridge w/ Cond. Deseason
Bayesian Ridge w/ Cond. I
Lasso w/ Cond. Deseason
Lasso Least Angular Regre
Elastic Net w/ Cond. Dese
AdaBoost w/ Cond. Desea
Croston
Orthogonal Matching Purs