A Comparative Analysis of Time Series Models and Machine Learning Methods on Bitcoin, Gold, and S&P 500 Prices

A Anish
Dept. of Mathematics
IIT Hyderabad

Sai Prateek Katkam
Dept. of Mathematics
IIT Hyderabad

Ashok Kumar
Dept. of Mathematics
IIT Hyderabad

Guide: Dr.Sameen Naqvi
Dept. of Mathematics
IIT Hyderabad

***Abstract*—This project endeavors to assess the predictive accuracy of diverse time series models and Machine Learning techniques in forecasting the prices of Bitcoin, Gold, and the S&P 500. The rationale behind selecting these three financial instruments lies in their distinct levels of volatility, providing valuable insights into how these models can effectively capture and interpret market fluctuations. The comparative analysis will shed light on the robustness and adaptability of the chosen methodologies across varying degrees of market dynamics inherent in these investment assets.**

I. Introduction

In this study, we delve into diverse ML prediction models to assess their accuracy in forecasting Bitcoin, Gold, and the S&P 500.

Bitcoin, a pioneering digital currency introduced in 2009 by the anonymous entity Satoshi Nakamoto, has revolutionized finance. Operating on a decentralized, peer-to-peer network, Bitcoin eliminates the need for traditional intermediaries in financial transactions.

Functioning as a cryptocurrency, Bitcoin employs cryptographic techniques to secure transactions and control the creation of new units, designed to serve as both a store of value and a medium of exchange with unique features.

The spot price represents the current market value at which a commodity can be instantly bought or sold with immediate payment and physical delivery. Gold, subject to supply and demand dynamics, experiences price fluctuations influenced by economic, geopolitical, and market factors. Even minor changes in gold prices can yield significant impacts on investors and government banks.

The S&P 500, a widely tracked stock market index, holds immense importance in gauging U.S. stock market performance and serves as an economic barometer. Often referred to as the S&P, or Standard & Poor's 500, it plays a crucial role in reflecting the broader economy.

Our dataset encompasses the closing prices of Bitcoin, gold futures rates, and S&P 500 stock prices, spanning from September 1, 2015, to August 31, 2023.

II. Data

A. Sourcing the data

- Gold spot price data was sourced from Gold.
- S&P 500 historical data was obtained from S&P.
- Bitcoin historical data was retrieved from Bitcoin.

B. Data Cleaning

- To ensure data integrity, we systematically addressed null values by removing them from the dataset. Additionally, we optimized the date column format to %Y-%m-%d for improved processing and analysis.

C. Exploratory Data Analysis

- Bitcoin - Starting Date of data 2015-09-01; Ending Date of data 2023-08-31

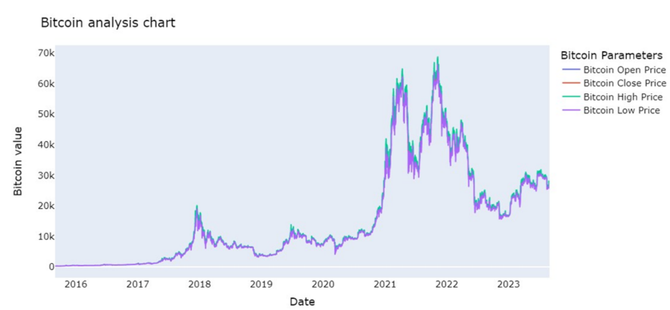

Fig. 1. Bitcoin Price Chart

- S&P 500 - Starting Date of data 2015-09-01; Ending Date of data 2023-08-31

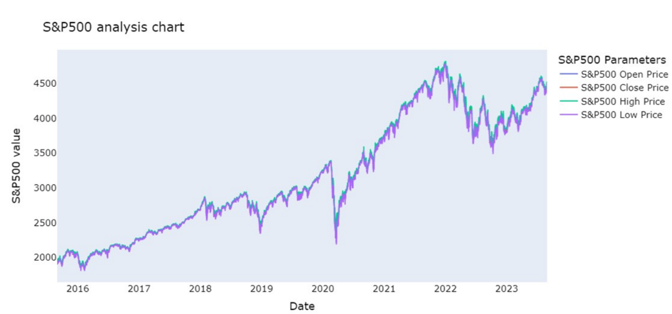

Fig. 2. S&P 500 Price Chart

- Gold - Starting Date of data 2015-09-01; Ending Date of data 2023-08-31

Fig. 3. Gold Price Chart

D. *Normalization of Data*

1) **Scale Consistency:** Normalizing data ensures that all features have the same scale. Machine learning algorithms often perform better when features are on a similar scale.
2) **Convergence Speed:** Algorithms relying on optimization (e.g., gradient descent) converge faster when features are normalized. This prevents one feature from dominating the learning process due to its larger scale.
3) **Improved Model Performance:** Normalization can lead to improved performance for models like k-Nearest Neighbors, clustering algorithms, and neural networks, where distances between data points are considered.

E. *Train-Test Split*

1) We split the data into train and test in the ratio 75:25
2) **Model Evaluation:** The primary reason for splitting data into training and testing sets is to evaluate the model's performance on unseen data.
3) **Overfitting Detection:** Train-test split helps in detecting overfitting by evaluating the model's performance on separate testing data.
4) **Hyperparameter Tuning:** During the model-building process, especially in machine learning, the testing set acts as a validation set to evaluate the model's performance with different hyperparameter configurations.
5) **Generalization Performance:** The ultimate goal is to build a model that generalizes well to new, unseen data. The testing set provides an unbiased evaluation of the model's performance.

In summary, normalizing data ensures that features are on a similar scale, benefiting various algorithms. The train-test split is crucial for evaluating a model's performance on unseen data, detecting overfitting, tuning hyperparameters, and assessing the model's generalization capabilities.

III. Analysis Methodology

A. *XGBoost Model*

XGBoost, or Extreme Gradient Boosting, is a powerful machine learning algorithm that belongs to the family of ensemble learning methods. It is particularly effective for regression and classification tasks. XGBoost builds a series of decision trees and combines their predictions to improve accuracy and reduce overfitting.

The analysis follows a structured methodology, as outlined below:

1) **Data Pre-processing:** Null values in the dataset are removed, and date formats are adjusted for ease of processing.
2) **Data Normalization:** The data is normalized to bring all features to a similar scale. Normalization ensures that each feature contributes proportionally to the model's learning process, preventing certain features from dominating others.
3) **XGBoost Model Training:** An XGBoost regressor model is trained using the training dataset. The model is configured with 1000 estimators.
4) **Prediction and Evaluation:** The trained model is used to make predictions on the test set. Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) metrics are calculated to assess the model's accuracy.
5) **Inverse Transformation:** Predictions and original data are transformed back to their original scale using the inverse of the MinMaxScaler.
6) **Visualization:** Various visualizations are created to compare original close prices, predicted prices on the training set, and predicted prices on the test set. Additionally, the model's performance in predicting the next 30 days is visualized.

Fig. 4. XGBoost Bitcoin Forecast

Fig. 5. XGBoost Gold Forecast

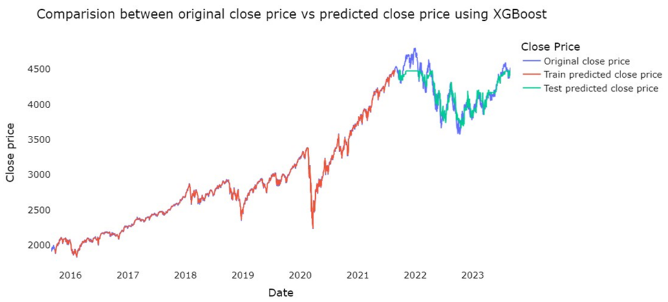

Fig. 6. XGBoost S&P 500 Forecast

TABLE I
XGBOOST ERROR VALUES

Metric	Bitcoin	S&P 500	Gold
MAE	0.0389	0.0269	0.0623
MSE	0.0025	0.00133	0.0066
RMSE	0.050	0.0365	0.0812

B. LSTM Model

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) architecture designed to capture and learn patterns in sequential data. It is particularly effective for time series prediction tasks. Here is an explanation of LSTM in points:

- **Architecture:** LSTM consists of memory cells that can store and access information over long sequences. It includes input, output, and forget gates that control the flow of information within the network.
- **Sequential Learning:** LSTM is capable of learning dependencies over long sequences, making it suitable for time-dependent data where patterns may exist over extended periods.
- **Memory Cells:** The memory cells in LSTM allow the model to selectively remember or forget information. This helps in capturing long-term dependencies in sequential data.
- **Preventing Vanishing/Exploding Gradients:** LSTM addresses the vanishing and exploding gradient problems encountered by traditional RNNs. This enables more stable and effective training on sequential data.
- **Training:** The LSTM model is trained on historical time series data to understand patterns and relationships within the data.
- **Prediction:** Once trained, the LSTM model can predict future values in a time series based on the learned patterns.
- **Evaluation Metrics:** Common evaluation metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE) are used to assess the accuracy of predictions.
- **Inverse Transformation:** Predictions and original data are transformed back to their original scale using the inverse of the normalization technique applied during preprocessing.
- **Visualization:** Visualizations are created to compare the original close prices with the predicted prices on both the training and test sets.

Fig. 7. LSTM Bitcoin Forecast

Fig. 8. LSTM Gold Forecast

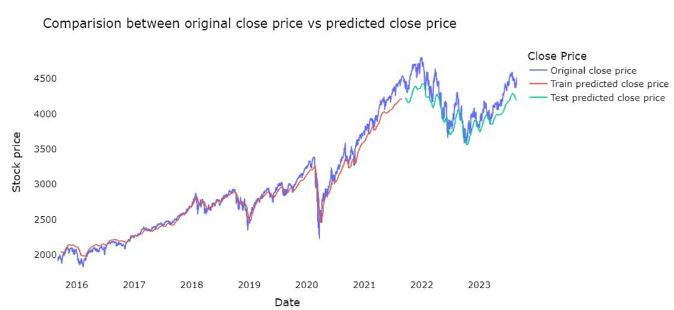

Fig. 9. LSTM S&P 500 Forecast

TABLE II
LSTM ERROR VALUES

Metric	Bitcoin	S&P 500	Gold
MAE	0.0929	0.0432	0.0138
MSE	0.0239	0.00296	0.00034
RMSE	0.1545	0.0544	0.0185

C. Prophet Model

Prophet is an open-source forecasting tool developed by Facebook for time series prediction. It is designed to handle daily observations with strong seasonal patterns.

- **Data Preparation:** The dataset is structured with a 'ds' column representing the date and a 'y' column containing the target variable (close prices). Additionally, the 'y' values are normalized using a MinMaxScaler.
- **Model Initialization:** An instance of the Prophet class is created and fitted to the dataset. The model is trained to capture seasonality and trends in the time series.
- **Future Date Generation:** Future dates are generated for the next 30 days to make predictions beyond the existing dataset.
- **Prediction:** The model predicts the stock prices for the future dates, providing predictions, lower and upper bounds for uncertainty.
- **Evaluation Metrics:** Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) are calculated to assess the accuracy of the predictions.
- **Visualization:** A plot of the predicted values, along with uncertainty bounds, is generated using the Prophet plot function.

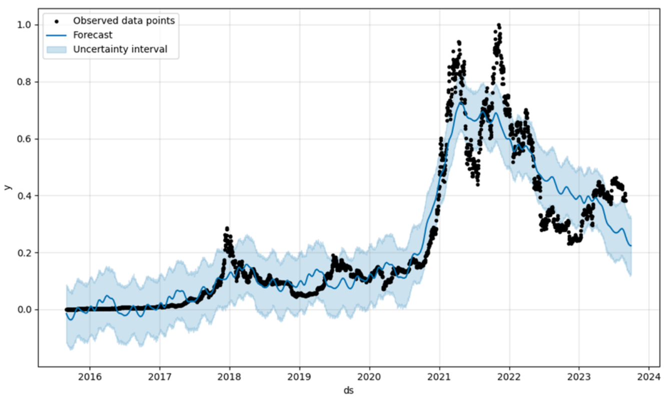

Fig. 10. Prophet Bitcoin Forecast

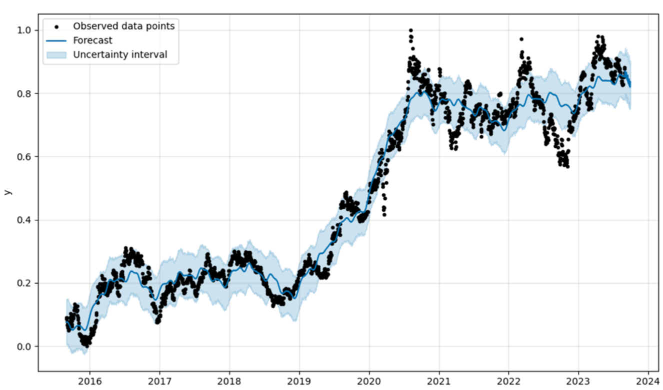

Fig. 11. Prophet Gold Forecast

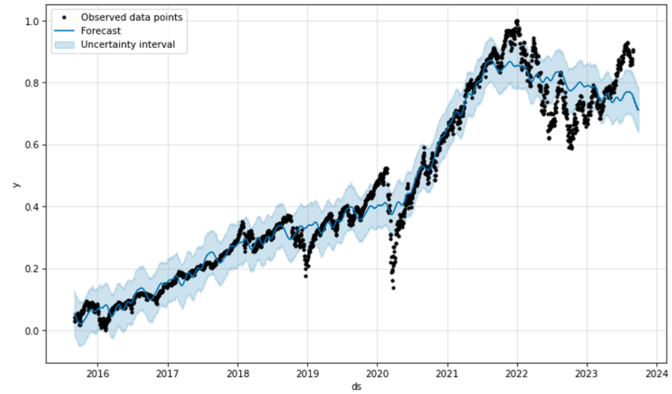

Fig. 12. Prophet S&P 500 Forecast

TABLE III
PROPHET ERROR VALUES

Metric	Bitcoin	S&P 500	Gold
MSE	0.0061	0.2878	0.0031
RMSE	0.0784	0.5365	0.0559
Train R2	0.8762	0.9593	0.9588

D. ARIMA Model

- ARIMA is a traditional statistical time series model that relies on autoregressive and moving average components. ARIMA models are based on the assumption that the time series is linear and stationary.
- **Data Preparation:** The time series data is loaded, ensuring a datetime index for the temporal component. The 'Gold Price' column is selected as the target variable for analysis.
- **Stationarity Check:** A check is performed to assess the stationarity of the time series using statistical tests like Augmented Dickey-Fuller (ADF) to determine the need for differencing.
- **Differencing:** If required, differencing is applied to make the time series stationary. This involves computing the difference between consecutive observations.
- **Autocorrelation and Partial Autocorrelation Analysis:** Autocorrelation and partial autocorrelation plots are generated to identify the autoregressive (AR) and moving average (MA) components. These plots help determine the values of parameters (p, d, q) for the ARIMA model.
- **ARIMA Model Initialization:** An ARIMA model is initialized with the chosen values of parameters (p, d, q) and fitted to the training data.
- **Model Evaluation:** The trained ARIMA model is used to forecast gold prices on the test set, and evaluation metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE) are computed.
- **Residual Analysis:** Residuals (the differences between predicted and actual values) are examined through plots and summary statistics to ensure the model captures the

underlying patterns.

- **Forecasting:** The ARIMA model is employed to generate forecasts for future gold prices, providing insights into potential future trends.
- **Visualization of Results:** Visualizations are created to showcase the actual gold prices, test data, and the ARIMA forecast.

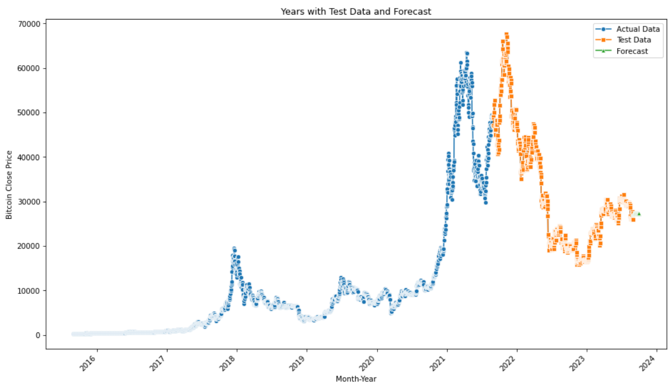

Fig. 13. ARIMA Bitcoin Forecast

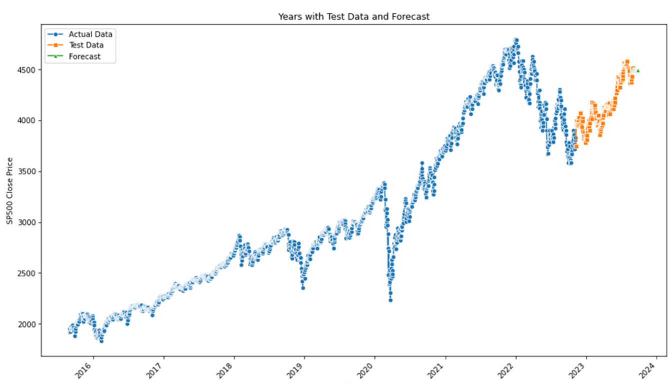

Fig. 14. ARIMA S&P 500 Forecast

Fig. 15. ARIMA Gold Forecast

TABLE IV
ARIMA ERROR VALUES

Metric	Bitcoin	S&P 500	Gold
MAPE	0.0225	0.0095	0.0440
MAE	763.23	39.39	79.21
MSE	4253371.99	2638.70	10071.31
RMSE	2062.37	51.37	100.36

IV. OBSERVATIONS

- **XGBoost Performance:**
  - XGBoost demonstrates superior performance across all three metrics (MAE, MSE, RMSE) for Bitcoin, S&P 500, and Gold.
  - The low error values indicate accurate predictions and robustness, making XGBoost a strong performer in financial time series forecasting.
- **Prophet Performance:**
  - Prophet exhibits reasonable performance, especially in terms of RMSE for Gold. However, it shows higher errors for Bitcoin and S&P 500 compared to XGBoost.
  - The R2 scores, while respectable, are lower than those achieved by XGBoost, indicating that XGBoost outperforms Prophet in capturing the underlying patterns.
- **ARIMA Performance:**
  - ARIMA performs well in terms of MAE and RMSE for Bitcoin and S&P 500, but the higher values for Gold suggest challenges in capturing its underlying patterns.
  - The high test R2 scores for Bitcoin and S&P 500 indicate good predictive power on the test set.
- **LSTM Performance:**
  - LSTM achieves competitive performance, particularly in terms of low MAE and RMSE for S&P 500 and Gold.
  - While LSTM demonstrates accuracy, the relatively higher errors for Bitcoin suggest challenges in predicting Bitcoin prices accurately.

V. CONCLUSION

A. Overall Comparison

- XGBoost emerges as the most robust model, consistently providing accurate predictions for all three datasets.
- LSTM and ARIMA showcase competitive performance but with variations across different datasets, indicating the importance of tailoring models to specific characteristics.
- Prophet, while performing reasonably well, lags behind XGBoost, suggesting that further optimization may enhance its predictive capabilities.

B. Conclusion

- The choice of the best model depends on the specific characteristics of the dataset and the desired balance between interpretability and accuracy.

- XGBoost stands out as the top performer, offering consistent accuracy and robustness in financial time series forecasting.
- LSTM and ARIMA present viable alternatives, each excelling in certain aspects, while Prophet may benefit from further fine-tuning to match the performance of XGBoost.

ACKNOWLEDGMENT

We express our sincere gratitude to Professor Sameen Naqvi for her invaluable guidance and unwavering support throughout the project. Her timely feedback played a crucial role in refining our work, contributing significantly to its improvement.

REFERENCES

[1] Iftikhar Sami et al. *Predicting Future Gold Rates using Machine Learning Approach*. Available at: https://www.researchgate.net/publication/322222520_Predicting_Future_Gold_Rates_using_Machine_Learning_Approach.

[2] Yang Wang et al. *A machine learning approach for stock price prediction*. Available at: https://dl.acm.org/doi/abs/10.1145/2628194.2628211.

[3] Obthong et al. *A survey on machine learning for stock price prediction: algorithms and techniques*. Available at: https://eprints.soton.ac.uk/437785/.