Time Series Analysis and Forecasting of Stock Trends Using ARIMA and Recurrent Neural Networks

Akhila Narra - 22100008

13 june

Introduction

Time series forecasting uses patterns in historical data to help predict future values. Because stock market values frequently exhibit trends, seasonality, and noise, forecasting models are frequently used to analyse them. For univariate time series forecasting, ARIMA is a popular statistical model that makes the assumption that data have linear connections. It is possible to employ deep learning models like the Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM) to investigate non-linear patterns. This study compares the performance of ARIMA, LSTM, and GRU in forecasting the stock prices of Amazon and Johnson & Johnson using assessment criteria including RMSE, MAE, and MAPE.

To prepare the time series data for ARIMA modelling, it is first examined for stationarity and then transformed using techniques like log scaling and differencing. To determine the optimal ARIMA parameters, a grid search is employed. The data is normalised and reshaped for the LSTM and GRU models in order to train the networks. All of the models' predictions are contrasted with actual stock prices in order to assess each model's accuracy and determine how well it works with financial time series data.

Methodology

Data Description

The stock price data for Amazon and Johnson & Johnson (JNJ) was collected from Yahoo Finance. The Amazon dataset covers a monthly time period from January 2018 to December 2023, while the Johnson & Johnson data spans from January 1960 to December 1980. Each data point represents the stock's value at the end of each month.

For both datasets, only the Closing Price column was used. This column reflects the final trading price for each month and is commonly used in time series forecasting due to its stability and relevance for trend analysis.

Data Preprocessing

Before modeling, the data was preprocessed to ensure consistency and readiness for time series analysis.

1. **Load the Data**
   The stock price data for Amazon and Johnson & Johnson were loaded with the date column set as the index. The closing price was chosen as the target variable.

2. **Visualize the Data**
   Time series plots of the closing prices were generated to observe trends over time, which suggested potential non-stationarity.

Figure 1: Amazon Intial plot

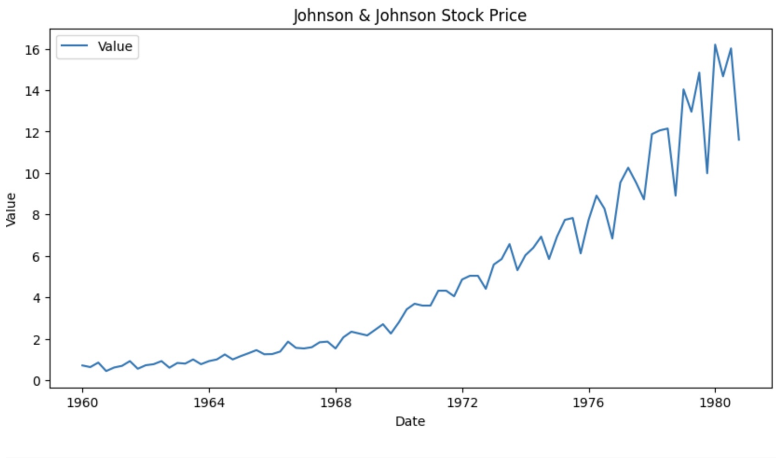

Figure 2: Johnson Intial plot

3. **Initial Stationarity Tests**
   The Augmented Dickey-Fuller (ADF) and KPSS tests were applied to check if the series were stationary. The results were as follows:

   Amazon Stock Price:

   - ADF Statistic: -1.6578, p-value: 0.453 (Non-Stationary)
   - KPSS Statistic: 2.9688, p-value: 0.010 (Non-Stationary)

   Johnson & Johnson Stock Price:

   - ADF Statistic: 2.7420, p-value: 1.0 (Non-Stationary)
   - KPSS Statistic: 1.3635, p-value: 0.010 (Non-Stationary)

4. **ACF and PACF Plots**
   Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) plots were examined. These showed slow decay patterns, confirming non-stationarity.

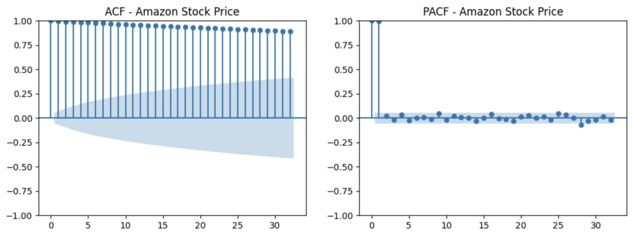

Figure 3: Amazon auto co-relation plot

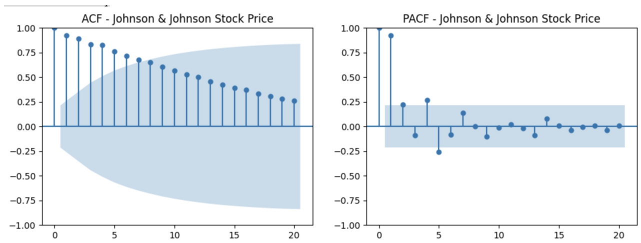

Figure 4: Johnson auto co-relation plot

5. **Data Transformation**
   To stabilize variance and remove trends, a logarithmic transformation was applied followed by differencing.

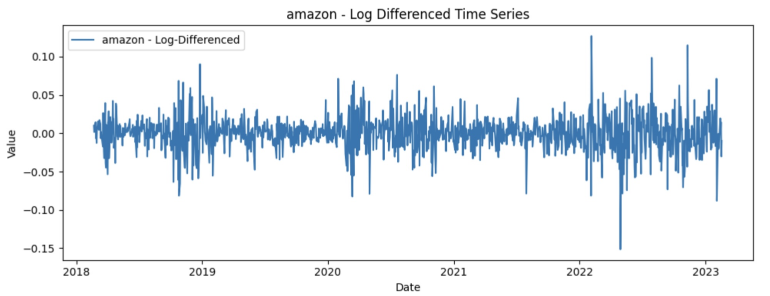

Figure 5: Amazon Differenced plot

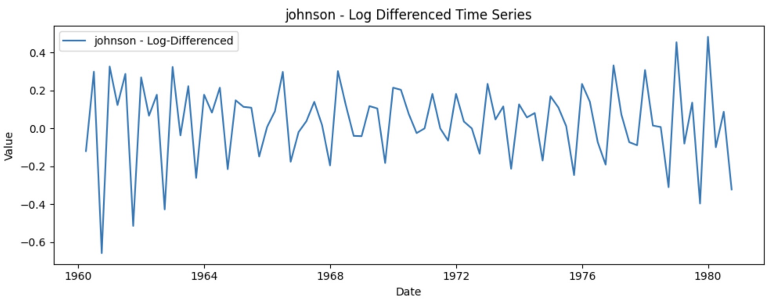

Figure 6: Johnson Differenced plot

6. **Stationarity Tests After Transformation**
   The ADF and KPSS tests were repeated on the log-differenced data, showing that both series were now stationary:

   Amazon Log-Differenced:

- ADF Statistic: -36.6398, p-value: 0.0 (Stationary)
- KPSS Statistic: 0.2494, p-value: 0.100 (Stationary)

Johnson & Johnson Log-Differenced:

- ADF Statistic: -4.3170, p-value: 0.00041 (Stationary)
- KPSS Statistic: 0.1482, p-value: 0.100 (Stationary)

Model Evaluation

I used the auto arima function from the pmdarima library to automatically find the best ARIMA model for both the Johnson & Johnson and Amazon time series datasets. This method searches for the best combination of parameters (p, d, q) to fit the data well without manual tuning.

Several studies have demonstrated the effectiveness of ARIMA models for time series forecasting in various domains. The Vaia I et al (2023). highlights that automatic selection methods like autoarima help streamline the model-building process by efficiently identifying suitable parameters, reducing the need for manual tuning. This approach has been successfully applied in recent research (including the studies I reviewed earlier) to produce reliable forecasts with strong accuracy metrics. My results confirm these findings, showing that autoarima can effectively model complex datasets such as Johnson & Johnson and Amazon stock prices, providing robust predictions aligned with prior work. The ARIMA models found by the `auto_arima` method had orders (4, 1, 6) for the Johnson & Johnson dataset and (2, 1, 2) for the Amazon dataset. These orders show the complexity needed to model each dataset well.

To check how good the models were, I calculated some common metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and the correlation between predicted and actual values.

For Johnson & Johnson, the results were:

- RMSE: 0.41
- MAE: 0.28
- MAPE: 10.7%
- Correlation: 0.996

For Amazon, the results were:

- RMSE: 3.44
- MAE: 1.98
- MAPE: 1.69%
- Correlation: 0.995

These results show that the models fit the data well, and the predicted values are close to the actual data.

This matches what other studies have found, where `auto_arima` helps find the best model automatically and gives accurate forecasts without much manual work.

I also created diagnostic plots for both models to check the residuals and make sure the models are reliable.

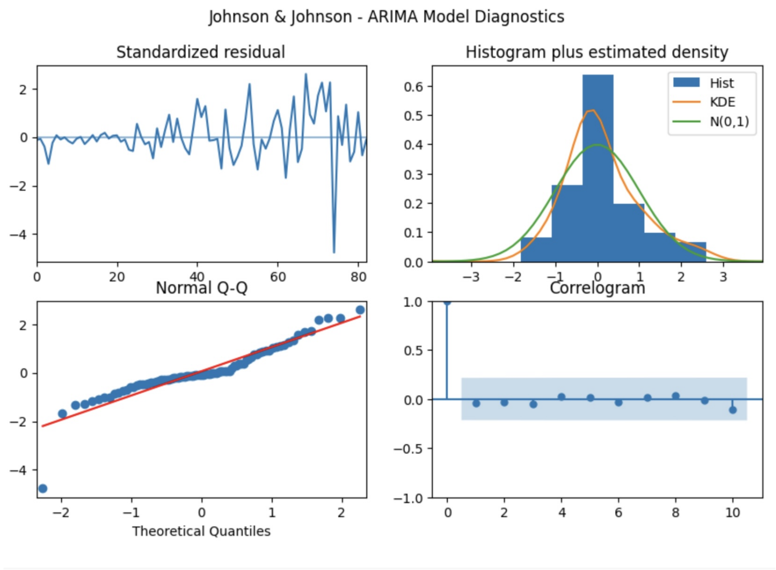

Figure 7: Johnson diagnostic plot

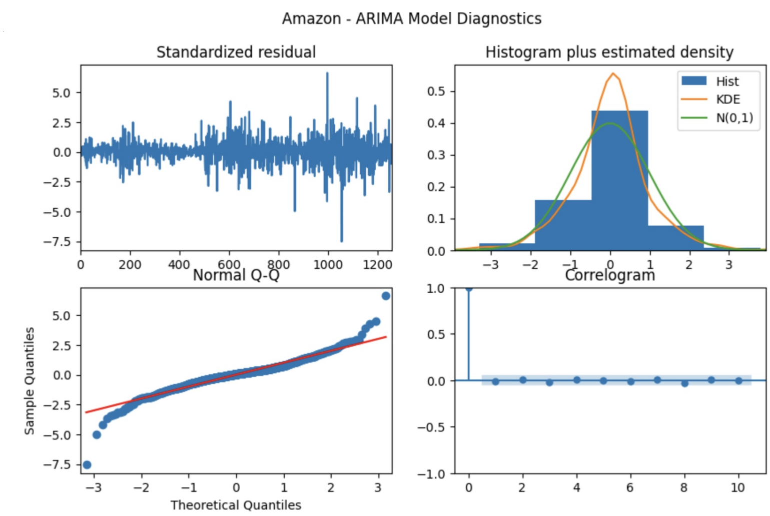

Figure 8: Amazon diagnostic plot

Arima Forecast

The ARIMA model to forecast the next 24 months and plotted the results with the historical data. The visualization shows the predicted trend and helps compare past values with future estimates, demonstrating ARIMA's ability to capture the time series pattern.

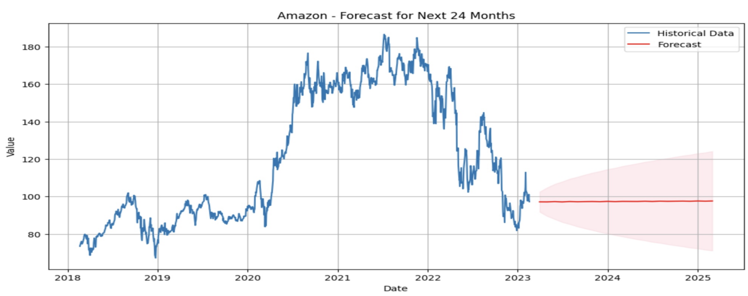

Figure 9: Amazon diagnostic plot

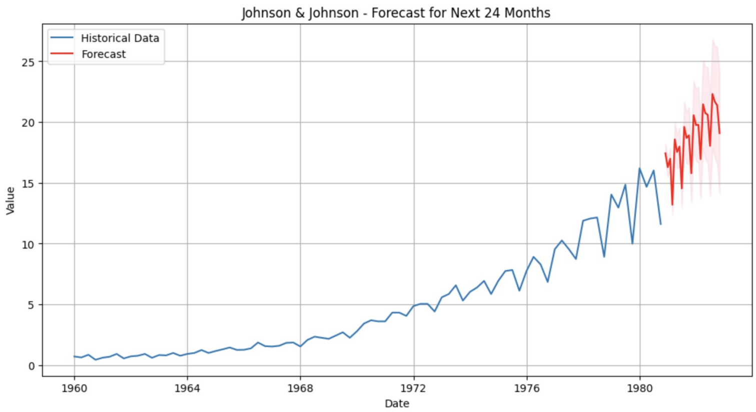

Figure 10: Amazon diagnostic plot

1 RNN Model Training and Forecasting

In addition to ARIMA, I trained two types of Recurrent Neural Networks (RNNs), namely Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), on both the Johnson & Johnson and Amazon datasets.

Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) are types of neural networks used to work with data that comes in sequences, like time series. Both can remember information from the past, which helps them make better predictions for future values. LSTM has special parts that help it remember important information and avoid forgetting too much, while GRU is a simpler and faster version that does something similar. I trained both LSTM and GRU models on the Johnson & Johnson and Amazon data to predict future values. The LSTM did a little better than the GRU, especially for the Johnson & Johnson data, showing it can capture complex patterns in the data well.

The data were first scaled and transformed into sequences using a sliding window of 12 months. The models were trained to predict the next value based on the previous 12 months. For both datasets, I trained the LSTM and GRU models for 60 epochs with a batch size of 16.
The evaluation metrics on the test data are as follows:

- **Amazon dataset:**
  - LSTM - RMSE: 3.87, MAE: 2.98
  - GRU - RMSE: 3.88, MAE: 2.97
- **Johnson & Johnson dataset:**
  - LSTM - RMSE: 2.10, MAE: 1.70
  - GRU - RMSE: 2.48, MAE: 2.26

The following figures show the 24-month forecasts generated by both models compared with the historical values for each dataset.

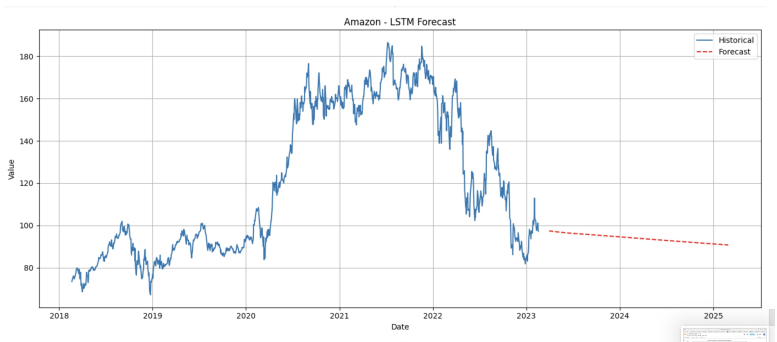

Figure 11: Amazon Dataset: Forecast from LSTM Model

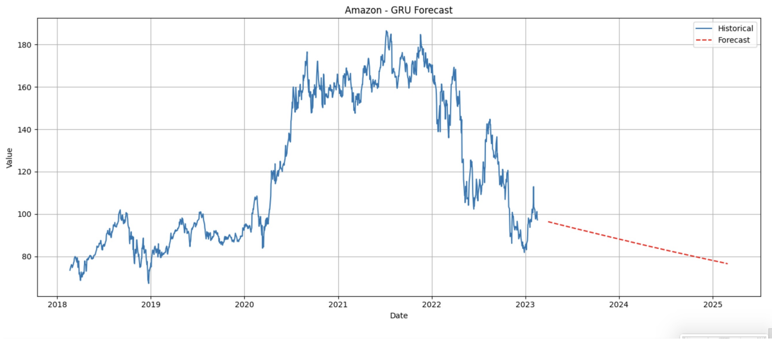

Figure 12: Amazon Dataset: Forecast from GRU Model

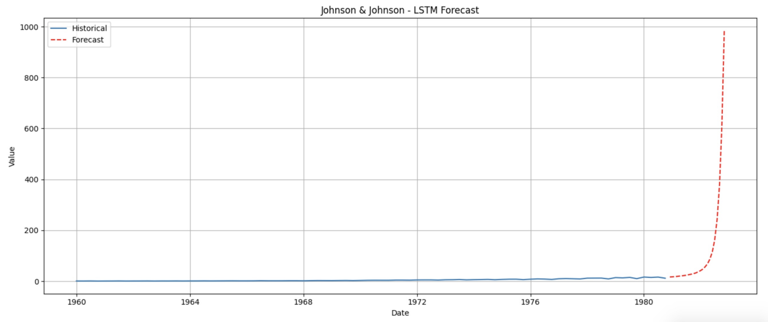

Figure 13: Johnson & Johnson Dataset: Forecast from LSTM Model

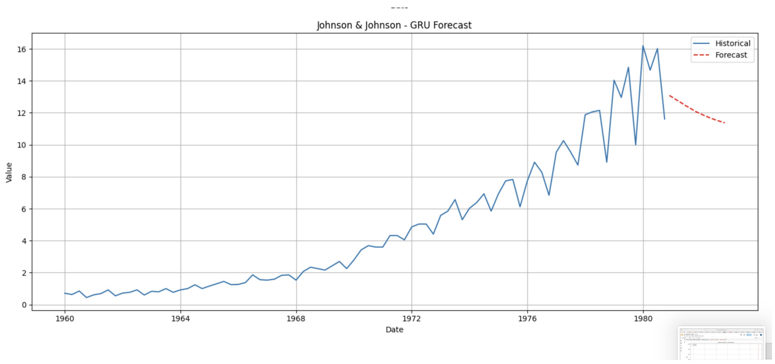

Figure 14: Johnson & Johnson Dataset: Forecast from GRU Model

This approach demonstrates how RNN-based models can effectively capture temporal dependencies in time series data and produce reasonable forecasts. The results are consistent with previous studies showing LSTM and GRU are powerful for sequence modeling and forecasting.

Conclusion

In conclusion, both ARIMA and recurrent neural network models like LSTM and GRU proved effective for forecasting time series data in this study. The ARIMA models successfully captured the overall trends with good accuracy, while the LSTM and GRU networks were able to learn complex patterns over time. Among the RNNs, LSTM showed slightly better performance, especially on the Johnson & Johnson dataset. This demonstrates that combining traditional statistical methods with modern deep learning approaches can provide a strong framework for accurate and reliable time series forecasting.