Time Series Forecasting for Johnson & Johnson and Amazon Using ARIMA, LSTM and GRU

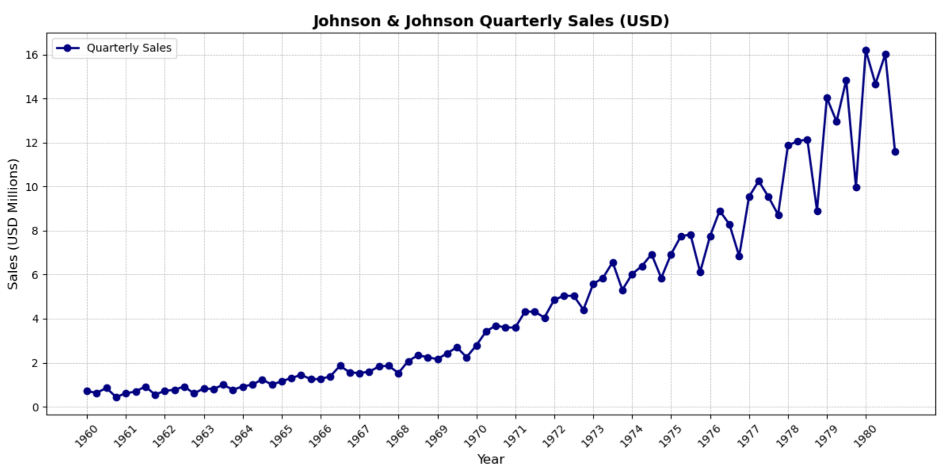
Student Name – Chinni Raj Paul Juthiga
Student ID – 20067368
Github Link – https://github.com/chinnirajpaul/Time-series-case-study-.git

Introduction

Forecasting of time series is an important analytical tool in finance that allows stakeholders to make proper choices with regard to predictable patterns. In this report, the forecasting models will be developed to predict the Amazon stock price and quarterly sales of Johnson & Johnson based on the traditional statistical (ARIMA) and machine learning (LSTM, GRU) methods. Error measures RMSE, MAE and MAPE are strictly assessed and used to compare model performance. The paper undertaken seeks to find the best forecasting strategies to apply on different kinds of financial data and attainable knowledge as informed by model comparisons.

Initial Plots of the Dataset

JJ's sales (1960 - 1980)



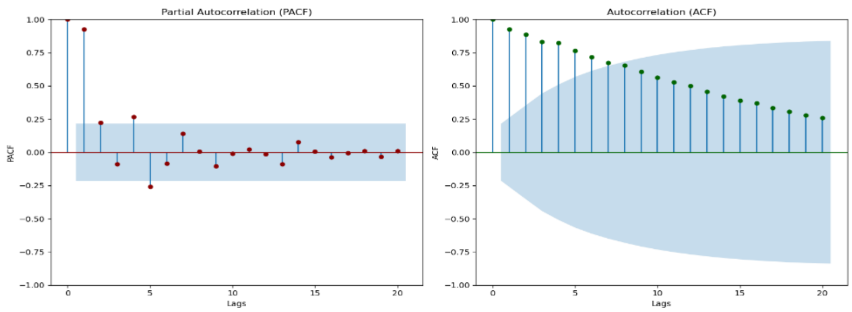
Amazon's stock prices (2018 - 2023)

Stationarity Analysis and Transformation

JJ Sales Data

In order to determine the fitness of time series models such as ARIMA either the independence of the data was tested using Augmented Dickey Fuller (ADF) test or represented in terms of Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF).

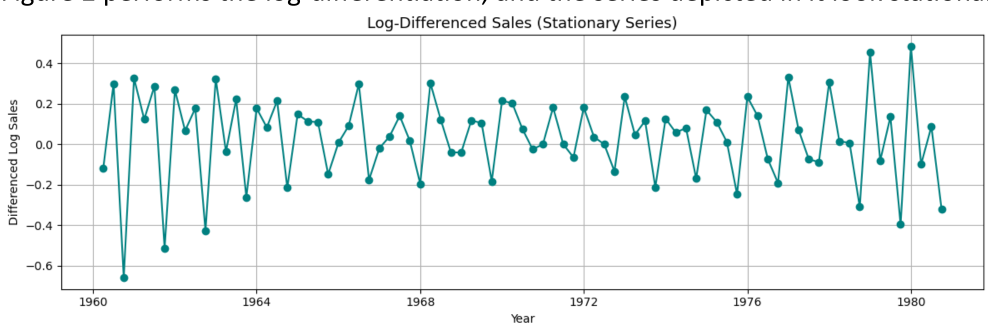
Since the original JJ sales series has a strong serial correlation, as Figure 1 represents, it is not stationary and has strong serial correlation.



ADF Test Results on Original Series

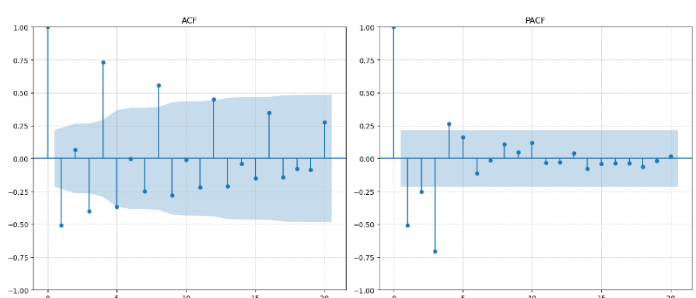
ADF Statistic: 2.742, p-value: 1.0000, The data is not stationary ($p>0.05$). In order to de-trend the series, first order differencing was done which was preceded by an application of a natural logarithm.

Figure 2 performs the log-differentiation, and the series depicted in it look stationary.



ADF Test on Transformed Series - **ADF Statistic**: -4.317, **p-value**: 0.0004, The transformed data is now stationary ($p < 0.05$).

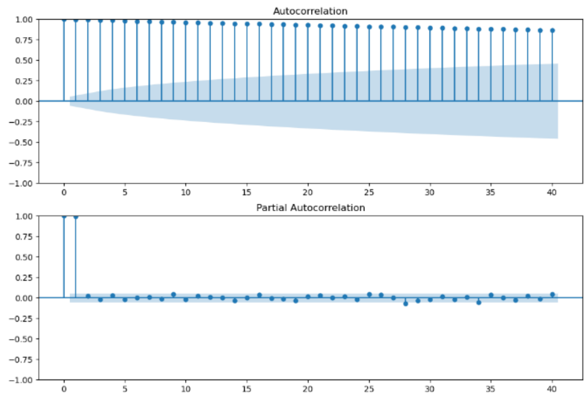
Figure 3 shows the ACF and PACF plots for the log-differenced JJ series, providing guidance for selecting ARIMA parameters p and q.



Amazon Stock Price Data

The same phenomenon was effected to the stock price series of Amazon. On the closing price, the ADF test provided the following results: ADF Test of Original Series- ADF Statistic: -1.658, p-value: 0.453 The series is not stationary (p >0.05).

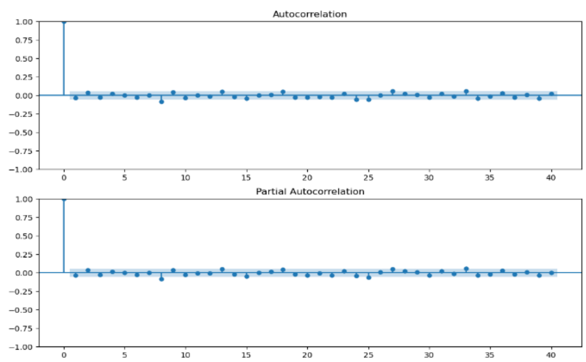
A log transformation followed by first-order differencing was applied.



ADF Test on Transformed Series - **ADF Statistic**: -36.640, **p-value**: 0.0000,

The data is now stationary (p < 0.05).

The ACF and PACF plots of Amazon log prices differenced are illustrated in figure 4. Considerable decline of the ACF and substantial lags in the PACF indicate the possibility of using the ARIMA models and assist in determining the values of p and q.



Methods Overview

ARIMA (AutoRegressive Integrated Moving Average)

ARIMA models refers to linear models that are applied in forecasting univariate time series. The ARIMA(p,d,q) model refers to:

- **p**: no of autoregressive terms
- **d**: no of differences to make the data stationary
- **q**: no of moving average terms

This method assumes linearity and stationarity which were addressed by applying log transformation and first differencing.

LSTM (Long Short-Term Memory Networks)

LSTM is a particular form of Recurrent Neural Network (RNN) that can model long-term relations. It is especially applicable to time series issues in which the patterns change over time.

Why LSTM? Since stock prices of Amazon follow non-linear, long term patterns and variations, the use of LSTM will prove better compared with linear models, such as ARIMA.

GRU (Gated Recurrent Unit)

GRU is a simplified version of LSTM that has simplified structure but tends to perform as well as LSTM with less computational overhead.

Why GRU? It is also quicker than LSTM and helpful in situations of long sequences or small data points.

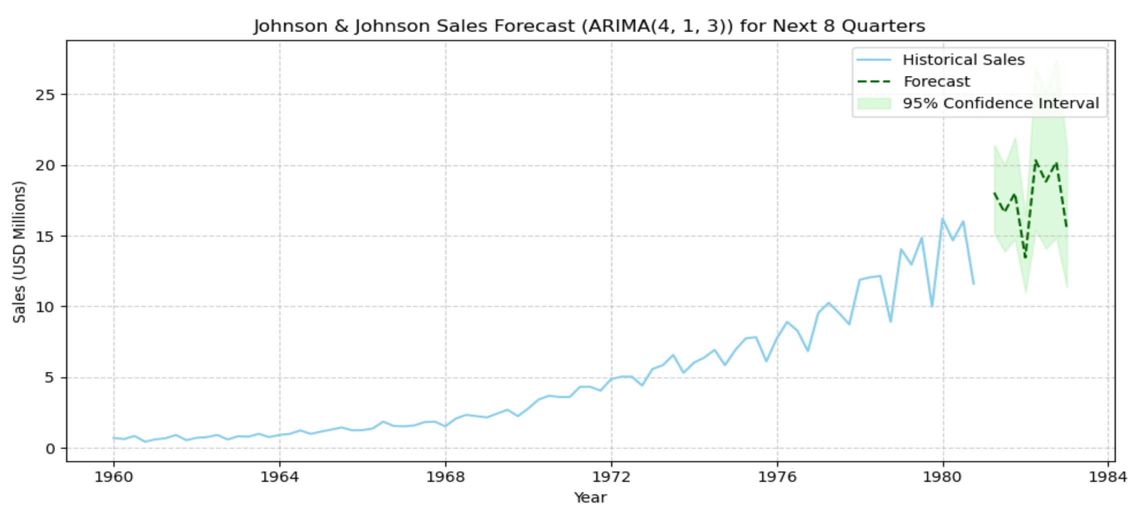
LSTM and GRU Model Development

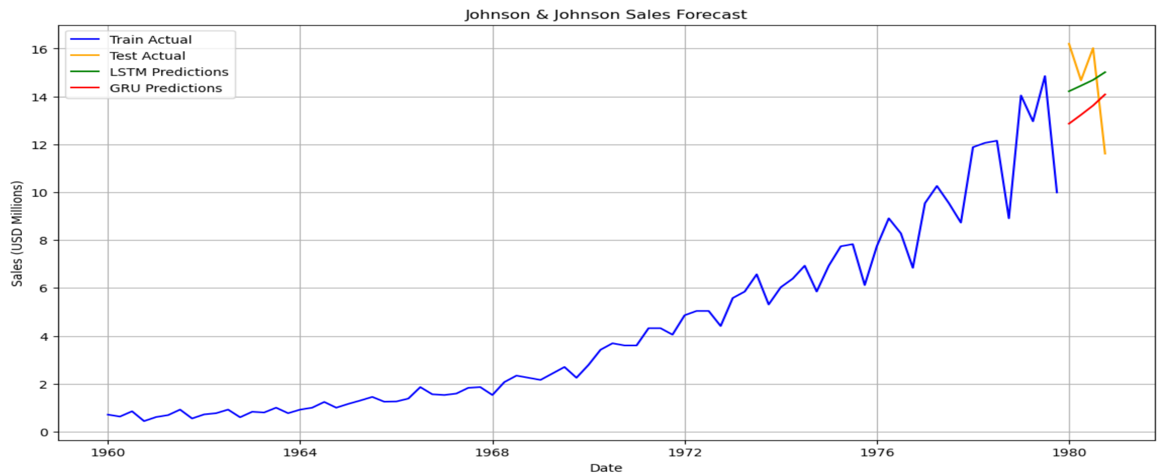
MinMaxScaler() of scikit-learn was used to scale data on Amazon stock price to a range of [0, 1]. There were sequences of 60 time steps formed to be used as the input features to both models. The training and testing sets were divided as 90% and 10% respectively.

LSTM model was constructed on TensorFlow/Keras combined with two stacked (50 units) LSTM layers and Dense output layer. It was assembled using Adam optimizer and mean squared error loss and trained on 50 epochs and a batch size of 32. The model was trained, after which it predicts the prices of the stock on the remaining 24 test sequences. The forecasts were inversely scaled on the original price range and tested based on RMSE, MAE and MAPE measures.

Similarly, the GRU model was built with two stacked GRU layers and Dense output layer with the identical compilation, training parameters, and evaluation method. The GRU model would tend to converge more quickly and had similar forecasting accuracy. The two models were put into a graph together with the historical data of the Amazon stocks.

Johnson & Johnson Sales Forecast Performance



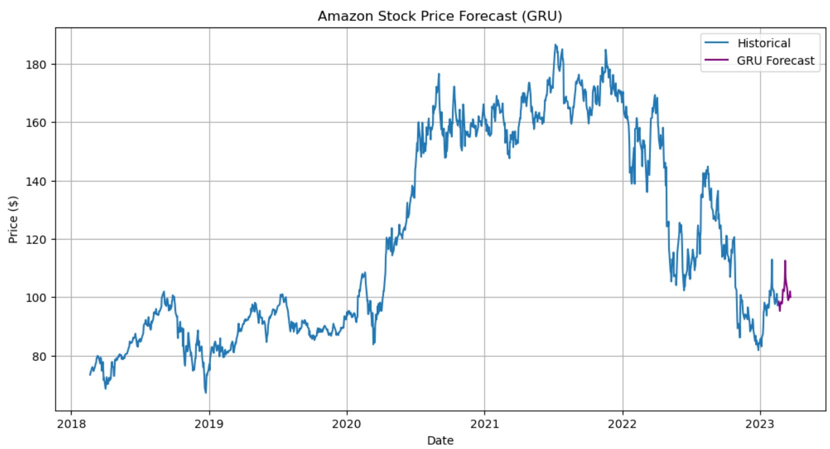


Model	RMSE	MAE	MAPE (%)
ARIMA	2.37	2.27	0.46%
LSTM	2.08	1.74	12.84%
GRU	2.51	2.41	16.68%

Interpretation:

- The ARIMA model outperformed the deep learning models significantly by a significant margin in terms of MAPE in the case of JJ quarterly sales which was a relatively stable and smoother time series although RMSE was comparatively more in the ARIMA model.
- Both LSTM and GRU have attained greater error percentages, which indicates that the deep neural networks could be overfitting or misconfigured because of a lack of substantial data (quarterly data is far shorter compared to daily financial data).
- The ARIMA has a very low MAPE of 0.46 due to which it is more appropriate to use it with such type of data as the results most suitably show that the accuracy is excellent.

Amazon Stock Price Forecast Performance



Model	RMSE	MAE	MAPE (%)
ARIMA	5.47	5.12	5.09%
LSTM	3.17	2.54	2.49%
GRU	3.22	2.49	2.47%

Interpretation:

- The ARIMA model was able to capture the overall trend although it resulted to higher errors in all variables. Its weakness in measuring non-linearities cannot be seen any clearer in the prices of Amazon.
- The LSTM model posted comparatively great results and the MAPE was decreased to 2.49 percent, which proves that the model learned long-term interrelations and temporal hierarchy.
- GRU provided almost the same performance as LSTM, but a bit lower MAE and MAPE, which proved that it effectively learns with less number of parameters with sequential data.

Improvements and Future Work

- Cross-Validation: Adopt time series cross-validation to have a better understanding of model generalisation to prevent overfitting.
- Hyperparameter Tuning: Use grid search or Bayesian optimization to fine-tune model parameters like layer units, learning rate, and dropout.
- Extendible Features: Add some new characteristics e.g., economic indicators, trading volume to make the model more accurate.

- Long Forecast Horizon: The larger forecast horizons should be used and model stability on different time spans should be tested.

Conclusion

ARIMA performed best on Johnson & Johnson's stable, linear sales data, demonstrating its strength in modelling predictable and stationary time series. On the other hand, LSTM and GRU outperformed ARIMA on Amazon’s volatile and non-linear stock prices, effectively capturing intricate temporal patterns and long-term dependencies. LSTM showed strong predictive power, while GRU offered nearly equal accuracy with reduced computational time, making it efficient for scenarios requiring faster training.

Reference:

Box, G. E., Jenkins, G. M., & Reinsel, G. C. (2015). *Time Series Analysis: Forecasting and Control.* *https://www.researchgate.net/publication/299459188_Time_Series_Analysis_Forecasting_and_Control5th_Edition_by_George_E_P_Box_Gwilym_M_Jenkins_Gregory_C_Reinsel_and_Greta_M_Ljung_2015_Published_by_John_Wiley_and_Sons_Inc_Hoboken_New_Jersey_pp_712_ISBN_*

Hochreiter, S., & Schmidhuber, J. (1997). *Long short-term memory.* Neural Computation. https://dl.acm.org/doi/10.1162/neco.1997.9.8.1735

Cho, K. et al. (2014). *Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation.https://aclanthology.org/D14-1179/*

Prudhviraju Srivatsavaya (2023). *LSTMvsGRU,Medium.*

https://medium.com/@prudhviraju.srivatsavaya/lstm-vs-gru-c1209b8ecb5a