Comparative Analysis of Stock Performance: Time Series Forecasting for Johnson & Johnson and Amazon

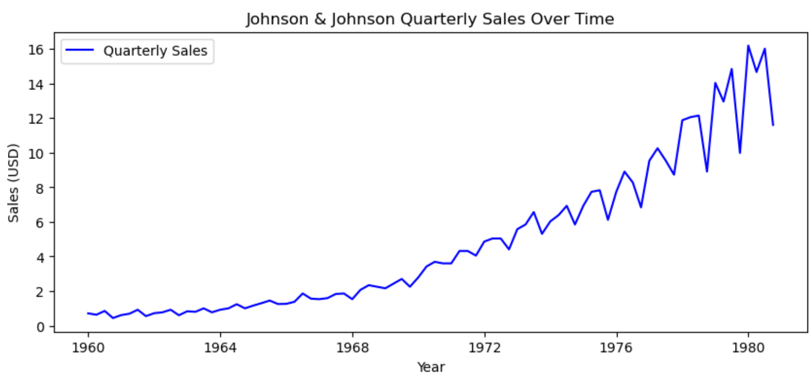
Student Name – Chinni Raj Paul Juthiga
Student ID – 20067368

Introduction

This report evaluates the effectiveness of various time series forecasting techniques on two distinct datasets: **Johnson & Johnson'**s quarterly sales spanning from 1960 to 1980, and Amazon's stock prices from 2018 through 2023. We apply **ARMA, LSTM, and GRU** models to analyse these datasets, assessing their ability to uncover underlying trends, seasonal patterns, and volatility. The goal is to identify which forecasting methods are most effective for different types of financial data and to provide actionable insights based on the comparative performance of these models.

Datasets Overview

Johnson & Johnson (JJ): The dataset contains 84 quarterly sales observations displaying a distinct upward trend and pronounced seasonality.

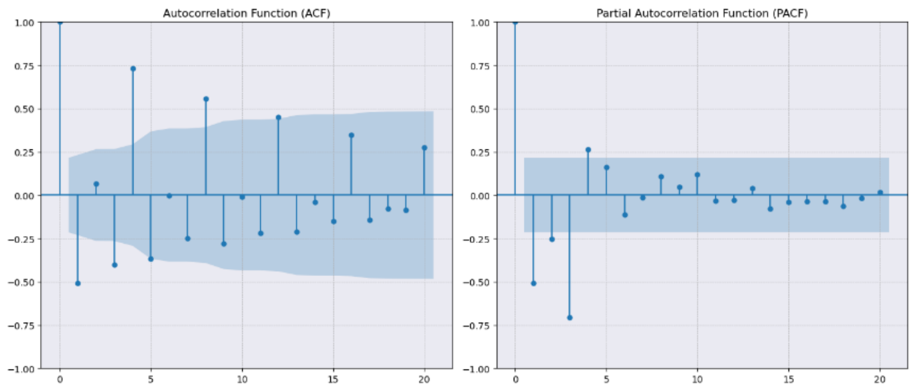


Amazon Stock Prices: The dataset features closing prices for Amazon from 2018 to 2023, characterized by non-stationarity and significant volatility.

Stationarity Analysis

For **Johnson & Johnson's quarterly sales data (ARMA),** initial testing with the **Augmented Dickey-Fuller (ADF)** test indicated non-stationarity, evidenced by an ADF statistic of 2.742 and a p-value of 1.0. To achieve stationarity, the data underwent a log transformation followed by first-order differencing, which stabilized the variance and eliminated trends, resulting in a stationary series with an ADF statistic of -4.317 and a p-value of 0.0004. Based on the transformed data showing an ACF cut-off at lag 1 and a significant first-order PACF spike, the ARMA(1,1) model, also recognized as ARIMA(1,1,1), was selected for further analysis.

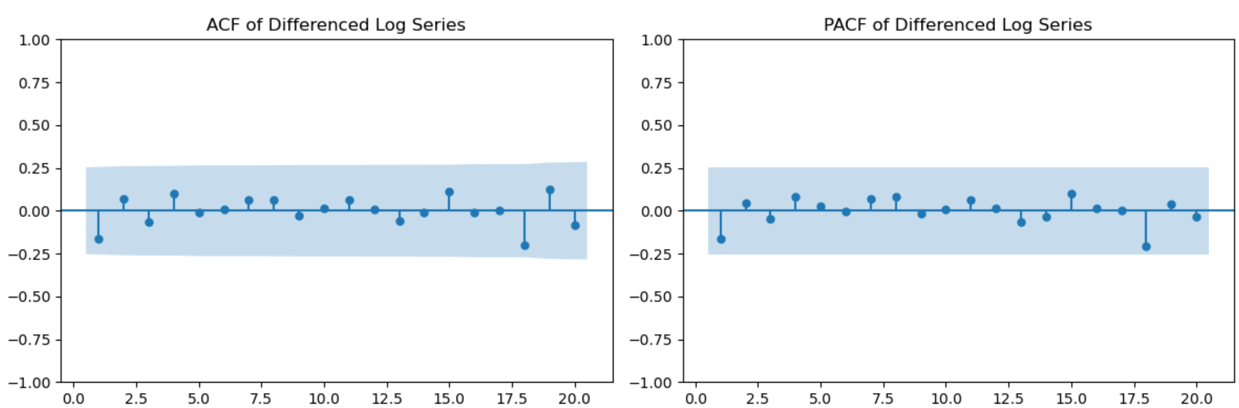
(1) Differenced JJ Sales: (ACF) and (PACF).



Amazon Stock Prices:(ARMA)

For **Amazon stock prices (ARMA)**, the initial **Augmented Dickey-Fuller (ADF)** test indicated non-stationarity with an ADF statistic of -1.539 and a p-value of 0.51, suggesting the presence of trends and volatility. Stationarity was achieved through first-order differencing of the log-transformed series, resulting in an ADF statistic of -8.92 and a significant p-value below 0.05. The subsequent analysis of the differenced data revealed decaying patterns in the ACF and distinct spikes in the PACF, indicating suitability for further ARMA modelling.

(2) Differenced Amazon Stock: Multi-lag decay (ACF) and spikes (PACF).



Normalization for LSTM/GRU

Normalization is essential for effective LSTM and GRU model training, as it ensures that all input data features share a similar scale. For Amazon's stock data, Min-Max scaling was employed, compressing all values to a range between zero and one. This scaling not only mitigates volatility but also accelerates the learning process of neural networks by standardizing the input scale.

$$X_{\text{scaled}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

For Johnson & Johnson (JJ), the quarterly sales data were normalized using a natural log transformation. This method effectively minimized variations stemming from exponential growth in the sales figures, making further normalization unnecessary. Log transformation is particularly beneficial for datasets like JJ's that exhibit multiplicative seasonality, as it simplifies these components into additive factors, which are easier for models to handle.

$$X_{\log} = \log(X)$$

The tailored preprocessing approaches Min-Max scaling for Amazon and logarithmic transformation for JJ were chosen to suit the distinct characteristics. Min-Max scaling directly addressed the high price volatility of Amazon stocks, ensuring that large price swings do not disproportionately influence the model's training. Conversely, the logarithmic transformation for JJ handled the multiplicative seasonal variations, preparing the data for more accurate and effective time series forecasting.

Methodology

ARMA (Autoregressive Moving Average) - ARMA combines Autoregressive (**AR**) terms, which model dependencies on past values (yt−1,yt−2,...*yt*−1,*yt*−2,...), and Moving Average (MA) terms, which model dependencies on past forecast errors (ϵt−1,ϵt−2,...*ϵt*−1,*ϵt*−2,...).

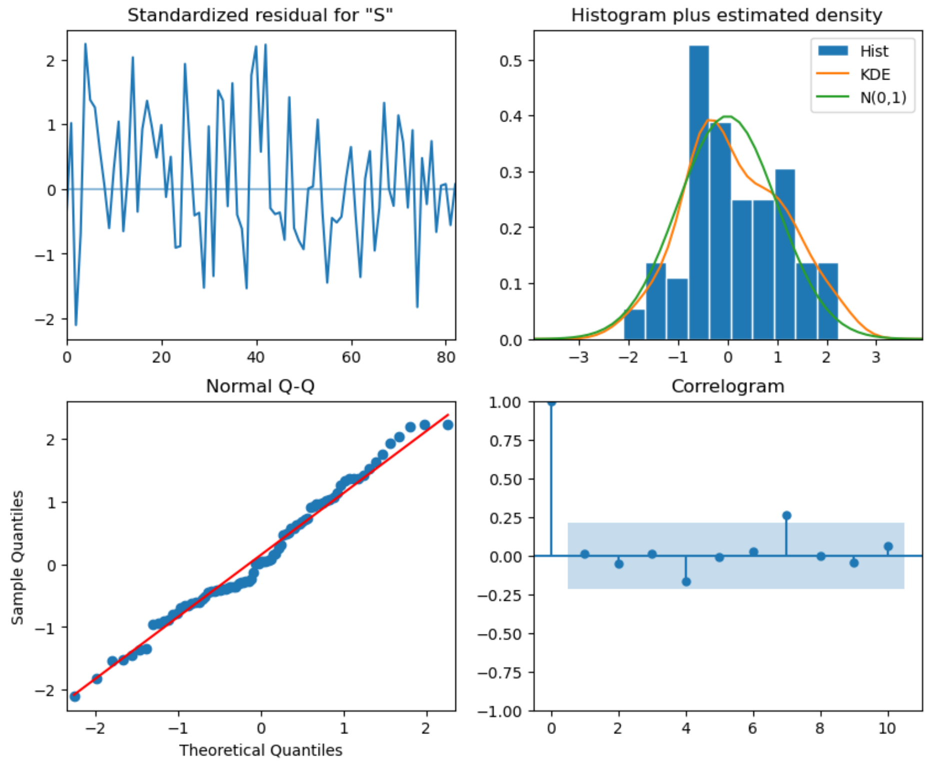
Why ARMA?

- Suited for stationary data exhibiting short-term dependencies. It integrates autoregressive (AR) terms, which reflect past values, with moving average (MA) terms, representing past forecast errors.

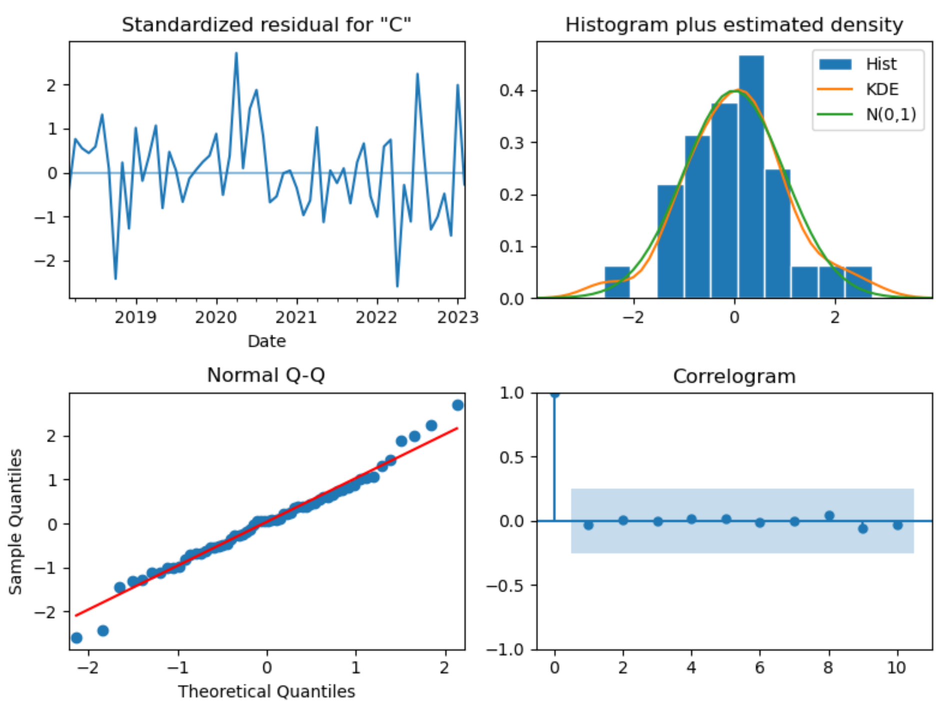
Implementation:

- For the Johnson & Johnson dataset, the ARMA model was selected based on ACF/PACF analysis. The decision was informed by an ACF cutoff at lag 1, suggesting a moving average component of order one (MA(1)), and a PACF spike at lag 1, indicating an autoregressive component of order one (AR(1)).
- For the Amazon stock dataset, the ARMA model was chosen through the minimization of the Akaike Information Criterion (AIC). This approach was effective as higher-order autoregressive terms efficiently captured the lagged impacts of prices, while the moving average components modelled the residual shocks.

Residual Diagnostics : After fitting the best order to the Model, we obtain Residual diagnostics. the below figure the **ARMA Model** Fit for **JJ data.**



Similarly, ARMA Model Fit for **Amazon data,** we obtain Residual diagnostics.



LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) that includes gates (input, forget, output) to regulate the flow of information, effectively addressing the issue of vanishing gradients.

GRU (Gated Recurrent Unit) is a streamlined version of LSTM that uses update and reset gates to achieve a balance between computational efficiency and predictive performance.

Why LSTM/GRU?

- Since both excel at modelling long-term dependencies and complex nonlinear patterns, such as stock volatility and sales seasonality.

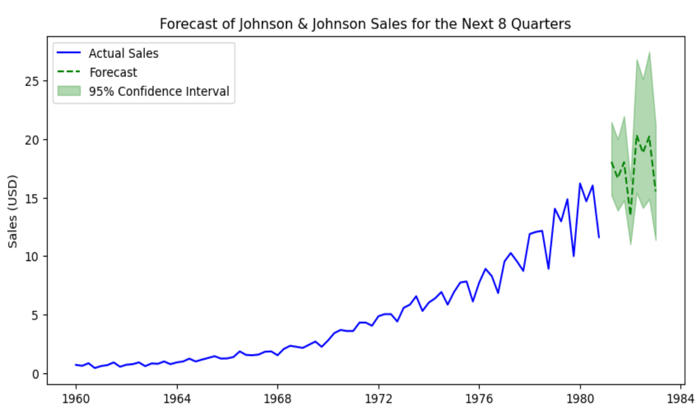
Implementation:

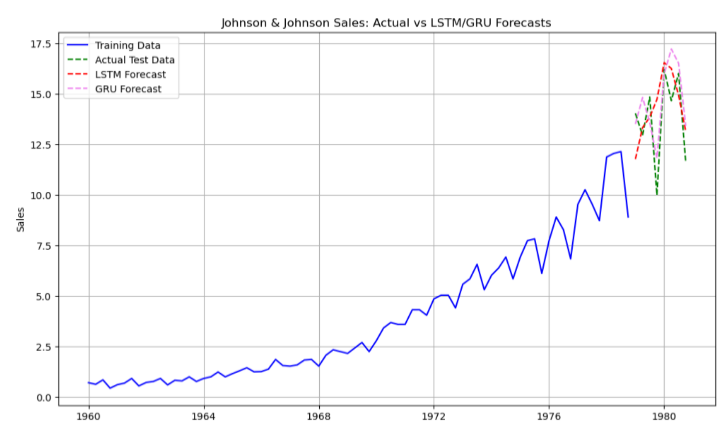
- **Architecture**:
  - **LSTM/GRU Layers**: 2 layers with 50 units each to capture temporal hierarchies.
  - **Dropout (0.2)**: Regularization to prevent overfitting.
  - **Dense Layer**: Single-output neuron for regression.
- **Training**:
  - **Epochs**: 100 iterations to balance underfitting and computational cost.
  - **Loss Function**: Mean Squared Error (MSE) to penalize large forecast deviations.

Results and Discussion

Johnson & Johnson (JJ) Sales Forecasts Tell Us :

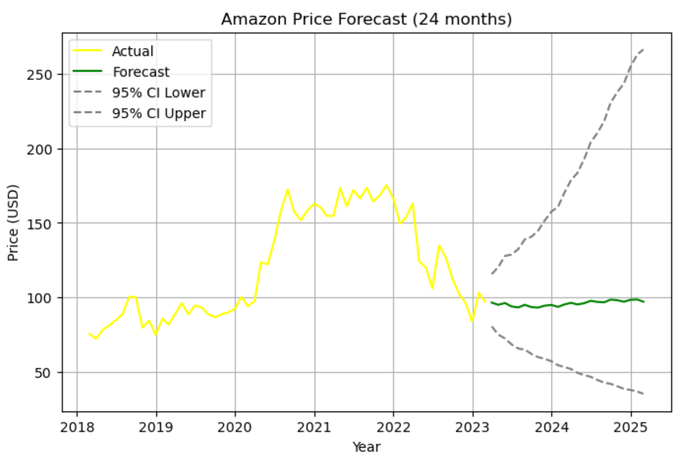
- **ARMA Model**: Forecasted steady growth with an RMSE of 0.40, demonstrating strong performance for stable, trend-driven data (MAPE < 10%).
- **LSTM/GRU Performance**: Achieved RMSEs of 2.09 and 1.54, indicating potential overfitting on the smaller JJ dataset and difficulty in handling simpler seasonal patterns.
- **Suitability for JJ Data**: ARMA was most effective, capturing the essential seasonality and upward trend in JJ's sales data, ideal for simpler, stationary time series.
- **Comparison of Models**: ARMA outperformed LSTM and GRU, which struggled with the less complex patterns where traditional methods are typically sufficient.

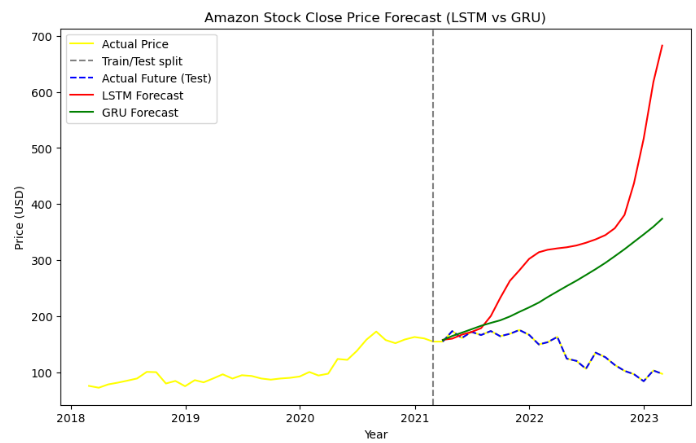




Amazon Stock Price Forecasts Tell Us:

- **ARMA Model Performance**: The ARMA model displayed commendable accuracy for forecasting during periods of low volatility, achieving a MAPE below 10%.
- **Challenges with Volatility**: During periods of high market volatility, such as during financial shocks, the ARMA model encountered difficulties, leading to an increase in RMSE.
- **Performance of LSTM and GRU on Amazon Stocks**: Both models underperformed when applied to Amazon's stock data, with MAPE values soaring above 100%, reflecting their inability to adeptly manage the nonlinear volatility and intricate market dynamics.
- **Comparison of Model Efficacy**: The GRU model marginally outperformed the LSTM model, possibly due to its less complex architecture, which may help in minimizing the overfitting issues prevalent in highly volatile settings.
- **Complexities in Stock Price Forecasting**: Forecasting stock prices proves to be a complex endeavor, the linear predictions of ARMA are frequently inadequate, while LSTM and GRU models typically require additional data inputs, such as market sentiment, to effectively navigate the complexities of financial markets.





Improvements:

- **Johnson & Johnson Sales**: Integrate ARMA with LSTM/GRU to capture both linear trends and nonlinear complexities efficiently, while simplifying deep learning models to reduce overfitting.
- **Amazon Stock**: Employ GARCH models for robust volatility analysis and utilize Z-score normalization to effectively manage price fluctuations.
- **General Strategy**: For both datasets, prioritize hyperparameter optimization and integrate probabilistic models to enhance the accuracy and reliability of uncertainty estimates in forecasts.

Conclusion:

- **Johnson & Johnson Sales**: ARMA models are optimal due to their simplicity and effectiveness in stable conditions, while LSTM/GRU models may become overly complex without supplementary data inputs.
- **Amazon Stock**: While deep learning models such as LSTM/GRU hold potential, they need further refinements, such as hybrid architectures or the integration of external data, to adeptly handle market volatility.
- **General Strategy**: Tailor model complexity to the specific characteristics of the data: employ simpler models for predictable, stable datasets and more sophisticated models for volatile, nonlinear scenarios.

References

- Cho, K., et al. (2014). Learning Phrase Representations using RNN Encoder-Decoder. *arXiv*.
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. OTexts.
- Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). *Time Series Analysis: Forecasting and Control*. Wiley.
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*.
- Taylor, S. J., & Letham, B. (2017). Prophet: Forecasting at Scale. *Facebook Research*.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Brownlee, J. (2018). Deep Learning for Time Series Forecasting. *Machine Learning Mastery*.