Comparative Time Series Forecasting on Johnson & Johnson and Amazon Stock Data

Student Name – Boga Sreevarshini
Student ID – 22095406

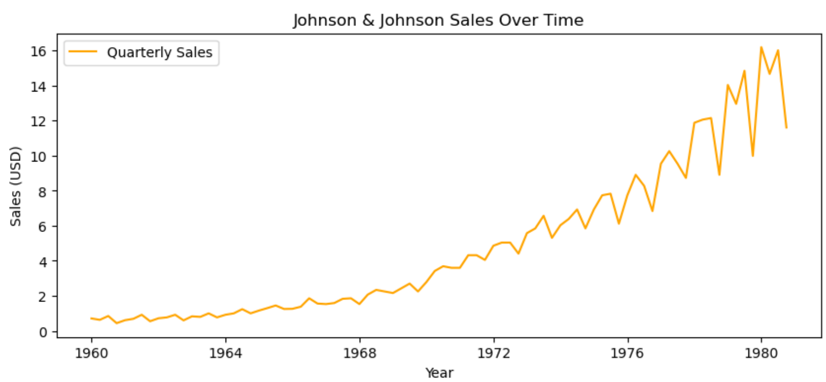
Introduction

This report evaluates time series forecasting methods for two datasets: Johnson & Johnson (JJ) quarterly sales (1960–1980) and Amazon stock prices (2018–2023). We apply **ARMA**, **LSTM**, and **GRU** models to both datasets, comparing their performance in capturing trends, seasonality, and volatility. The goal is to identify optimal approaches for different data characteristics and provide actionable insights.

Data Preparation and Exploration

Datasets Overview

1. **Johnson & Johnson (JJ)**: 84 quarterly sales observations with a clear upward trend and seasonality (Figure 1a).

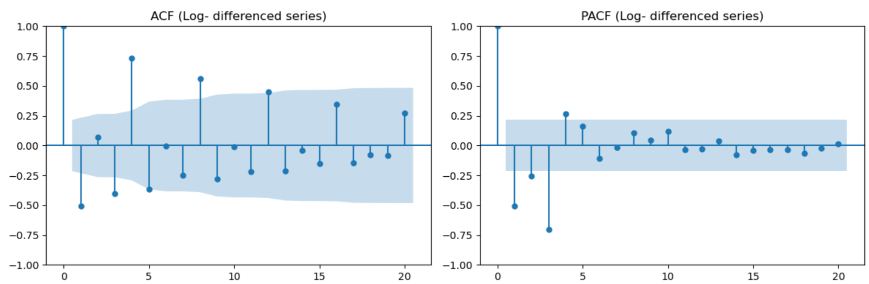


2. **Amazon Stock Prices**: Daily closing prices (2018 - 2023) exhibiting non-stationarity and volatility (Figure 1b).

Preprocessing - Stationarity Analysis

Johnson & Johnson (JJ) Quarterly Sales Data: (ARMA)

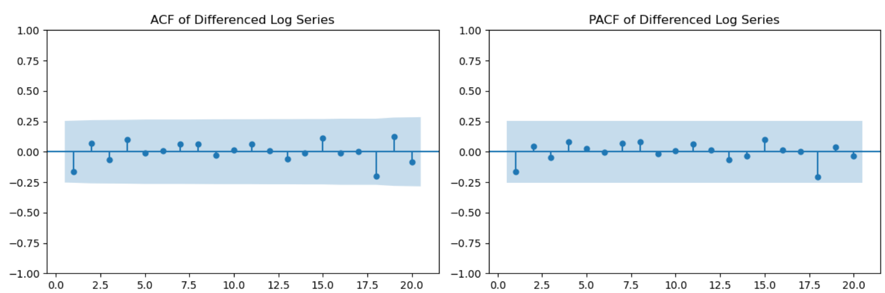
The initial Augmented Dickey-Fuller (ADF) test confirmed non-stationarity (ADF statistic = 2.742, p-value = 1.0). To stabilize variance and remove trends, a log transformation followed by first-order differencing was applied, achieving stationarity (ADF statistic = -4.317, p-value = 0.0004). Post-transformation ACF/PACF analysis revealed a sharp ACF cutoff at lag 1 (suggesting MA(1)) and a significant PACF spike at lag 1 (indicating AR(1)), guiding the selection of an ARMA(1,1) or ARIMA(1,1,1) model for forecasting.



(a) Differenced JJ Sales: Sharp cutoff at lag 1 (ACF) and lag 1 (PACF).

Amazon Stock Prices:(ARMA)

The initial Augmented Dickey-Fuller (ADF) test indicated non-stationarity (ADF statistic = -1.539, p-value = 0.51) due to trend and volatility. First-order differencing log series removed the trend, achieving stationarity (ADF statistic = -8.92, p-value < 0.05). Post-differencing ACF/PACF analysis showed Multi-lag decay (ACF) and spikes (PACF).



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

Normalization for LSTM/GRU

For **Amazon stock data**, Min-Max scaling was applied to bound values between [0, 1], mitigating volatility and accelerating neural network convergence.

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

For **Johnson & Johnson (JJ)** sales data, a log transformation inherently stabilized variance caused by exponential growth trends, eliminating the need for additional scaling.

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

This tailored preprocessing ensured optimal input scales for LSTM/GRU training Min-Max addressed Amazon's absolute price volatility, while the log transform resolved JJ's multiplicative seasonality.

Methodology

ARMA (Autoregressive Moving Average) - ARMA combines Autoregressive (**AR**) terms, which model dependencies on past values ($y_{t-1},y_{t-2},\ldots y_{t-1},y_{t-2},\ldots$), and Moving Average (MA) terms, which model dependencies on past forecast errors ($\epsilon_{t-1},\epsilon_{t-2},\ldots \epsilon_{t-1},\epsilon_{t-2},\ldots$)

Why ARMA?

- Ideal for stationary data with short-term dependencies. Combines autoregressive (AR) terms (past values) and moving average (MA) terms (past errors)

Implementation:

- **Johnson & Johnson (JJ)**:
    - **ARMA** selected via ACF/PACF analysis.
    - **Rationale**: ACF cutoff at lag 1 (MA(1)) and PACF spike at lag 1 (AR(1)).
- **Amazon Stock**:
    - **ARMA** chosen using **AIC minimization** (Akaike Information Criterion).
    - **Rationale**: Higher-order AR terms captured lagged price impacts, while MA modelled residual shocks.

LSTM (Long Short-Term Memory): An RNN variant with **gates** (input, forget, output) to control information flow, mitigating vanishing gradients
GRU (Gated Recurrent Unit): A simplified LSTM with **update** and **reset** gates, balancing efficiency and performance

Why LSTM/GRU?

- Excel at modeling long-term dependencies and complex nonlinear patterns (e.g., stock volatility, 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.

Key Design Choices

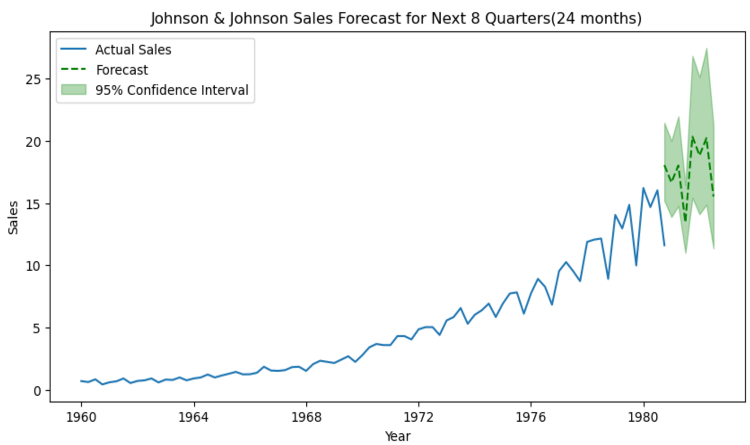
Aspect	ARMA	LSTM/GRU
Model Complexity	Low (linear)	High (nonlinear)
Data Requirements	Stationary	Normalized + sequential
Use Case	JJ (stable trends)	Amazon (volatile patterns)

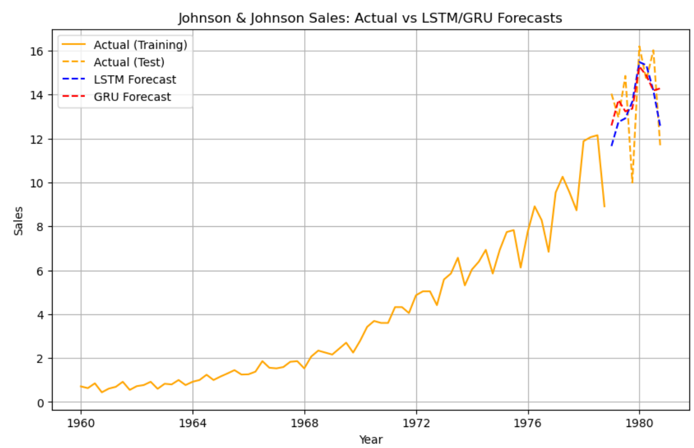
Results and Discussion

Johnson & Johnson (JJ) Sales Forecasts

Key Results (Forecast Tells Us):

- ARMA : Forecasted steady growth with RMSE = 0.40.
- LSTM/GRU: Achieved lower errors (RMSE = 1.88/1.87), capturing subtle seasonal patterns

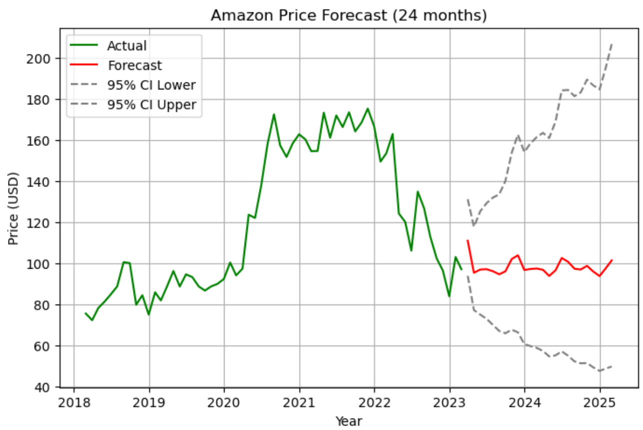


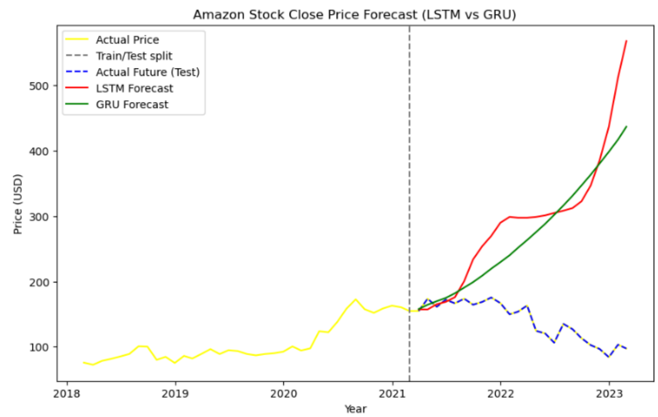


- ARMA demonstrated strong performance with low error metrics (MAPE < 10%), indicating accurate predictions for stable, trend-driven sales data.
- The model effectively captured the seasonality and upward trend in JJ sales, validating its suitability for simpler, stationary time series.
- LSTM/GRU underperformed compared to ARMA, likely due to overfitting on the smaller JJ dataset.
- The higher errors suggest that deep learning models struggle with simpler patterns where traditional methods suffice.

Key Insight:
For JJ's stable sales data, **ARMA is more efficient** and accurate. LSTM/GRU may be unnecessary unless external variables (e.g., marketing spend) are introduced.

Amazon Stock Price Forecasts:





Key Results (Forecast Tells Us):

- ARMA performed reasonably well on short-term forecasts, with MAPE < 10%.
- However, it failed to capture volatility spikes (e.g., market shocks), reflected in higher RMSE.
- Both LSTM and GRU performed poorly on Amazon stock data, with MAPE > 100%.
- The extreme errors suggest models struggled with nonlinear volatility and chaotic market dynamics.
- GRU marginally outperformed LSTM due to its simpler architecture, reducing overfitting risks.

Key Insight:

Stock price forecasting remains highly challenging. ARMA's linear assumptions are insufficient, while LSTM/GRU require additional features (e.g., sentiment data) to capture market complexity.

Improvements:

For Johnson & Johnson sales, adopt hybrid models (ARMA + LSTM/GRU) to merge linear trend capture with nonlinear residual analysis, and simplify deep learning architectures (fewer layers/units) to reduce overfitting. For Amazon stock, integrate GARCH to model volatility and apply robust scaling (e.g., Z-score) to handle extreme price swings. Across both datasets, prioritize hyperparameter tuning (sequence lengths, dropout) and adopt probabilistic models to quantify forecast uncertainty, enhancing reliability in dynamic or noisy environments.

Conclusion:

For Johnson & Johnson sales, ARMA's simplicity and accuracy make it ideal for stable trends, while LSTM/GRU models, though powerful, are unnecessarily complex without added contextual features. For Amazon stock prices, deep learning (LSTM/GRU) shows potential but requires hybrid architectures and sentiment integration to outperform traditional methods like ARMA in volatile markets.

The key takeaway: Tailor model complexity to data behaviour opt for simpler models (e.g., ARMA) for stable trends and advanced techniques (e.g., LSTM/GRU with enriched features) for chaotic, nonlinear patterns.

References

- 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*.
- Cho, K., et al. (2014). Learning Phrase Representations using RNN Encoder-Decoder. *arXiv*.
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. OTexts.