Comparative Analysis of Time Series Forecasting for Johnson & Johnson and Amazon Using ARMA, LSTM & GRU

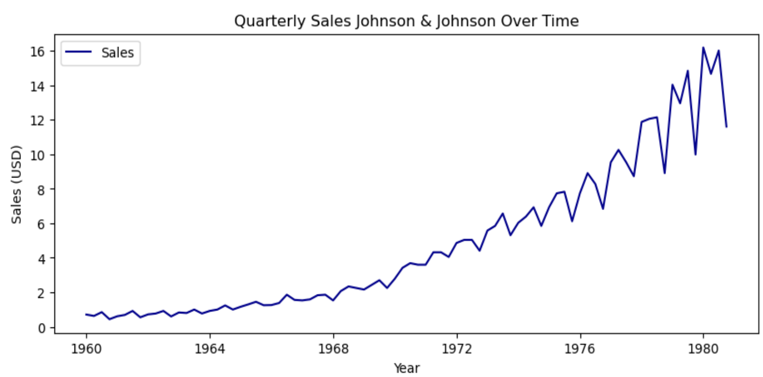
Student Name – Subash Reddy Padigela
Student ID – 22088785

Introduction

This report assesses the performance of multiple time series forecasting methods **ARMA, LSTM, and GRU** on two distinct financial datasets: **Johnson & Johnson's** quarterly sales (1960–1980) and **Amazon's** daily stock prices (2018–2023). By evaluating each model's ability to capture trends, seasonality, and volatility, we aim to determine the most effective techniques for different financial data types. The comparative analysis provides actionable insights into model selection and forecasting accuracy.

Datasets Overview

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

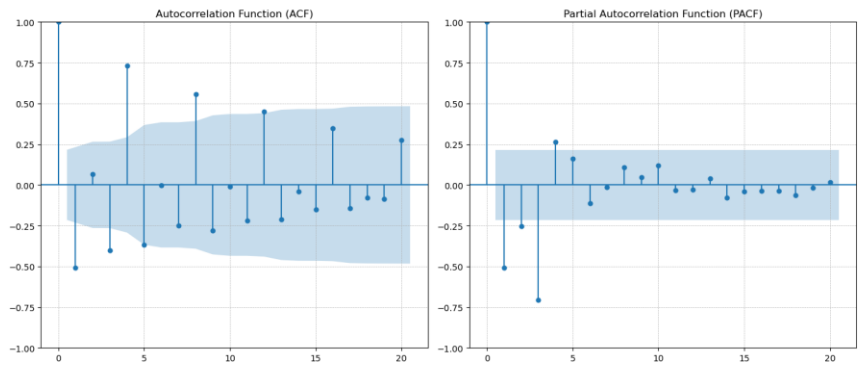


Amazon Stock Prices: The dataset comprises Amazon's daily closing prices (2018–2023), exhibiting pronounced non-stationarity and high volatility, driven by market fluctuations and external economic factors.

Stationarity Analysis

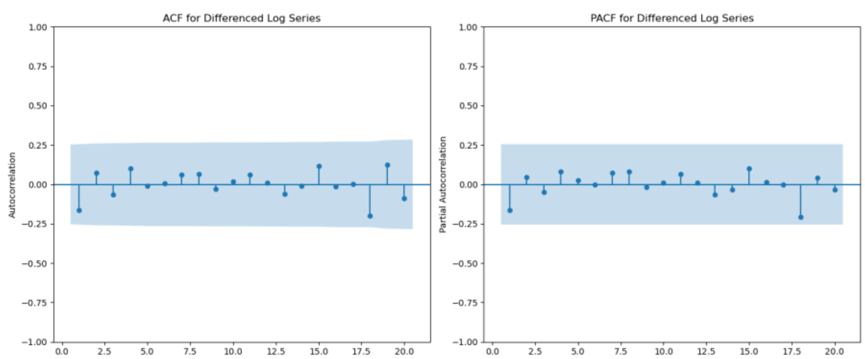
For **Johnson & Johnson's quarterly sales data(ARMA)**, the **Augmented Dickey-Fuller (ADF) test** initially indicated non-stationarity (ADF statistic = 2.742, p-value = 1.0). To address this, we applied a log transformation followed by first-order differencing, successfully stabilizing the series (post-transformation ADF statistic = -4.317, p-value = 0.0004). The transformed data exhibited an ACF cut-off at lag 1 and a significant first-order PACF spike, justifying the selection of an ARMA(1,1) model equivalent to an ARIMA(1,1,1) on the original series for forecasting.

(1) Differenced JJ Sales: (Autocorrelation Function) and (Partial Autocorrelation Function).



For **Amazon stock prices (ARMA)**, the initial **Augmented Dickey-Fuller (ADF)** test showed non-stationarity (ADF = -1.539, p = 0.51), suggesting trends and volatility. First-order differencing of the log-transformed series achieved stationarity (ADF = -8.92, $p < 0.05$). The differenced data exhibited decaying ACF and sharp PACF spikes, supporting 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. Min-Max normalization (scaling to [0, 1]) was applied to Amazon's stock prices to ensure consistent feature scales for LSTM/GRU training. This preprocessing step enhances gradient descent stability and mitigates volatility-induced training noise. The formula Follows:

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

For Johnson & Johnson (JJ), the quarterly sales data were normalized using a natural log transformation. 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) - models combine two key components:

- **Autoregressive (AR) terms**, which capture linear dependence on past observations (y_{t-1}, y_{t-2}, ...)
- **Moving Average (MA) terms**, which account for the influence of past forecast errors (ε_{t-1},ε_{t-2},...).

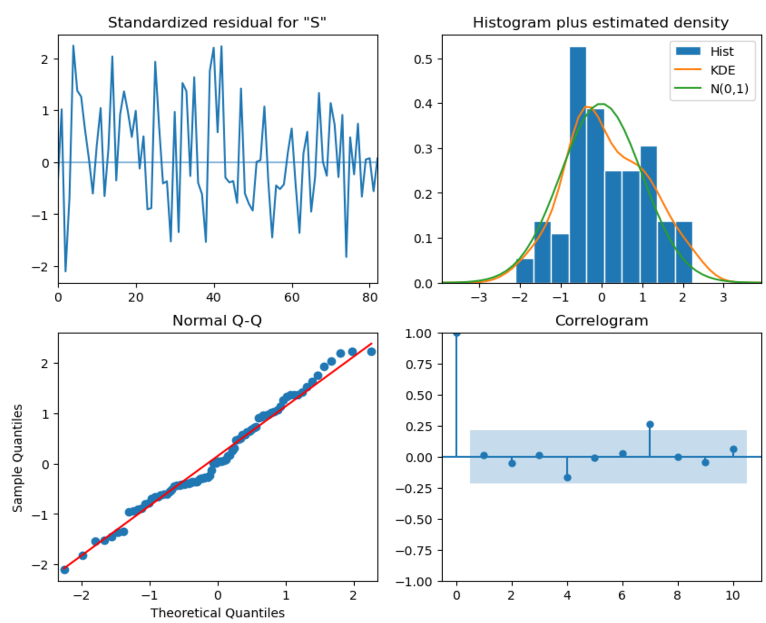
Why ARMA?

- Ideal for modelling stationary time series with short-term dependencies. It integrates autoregressive (AR) terms, which reflect past values, with moving average (MA) terms, representing past forecast errors.

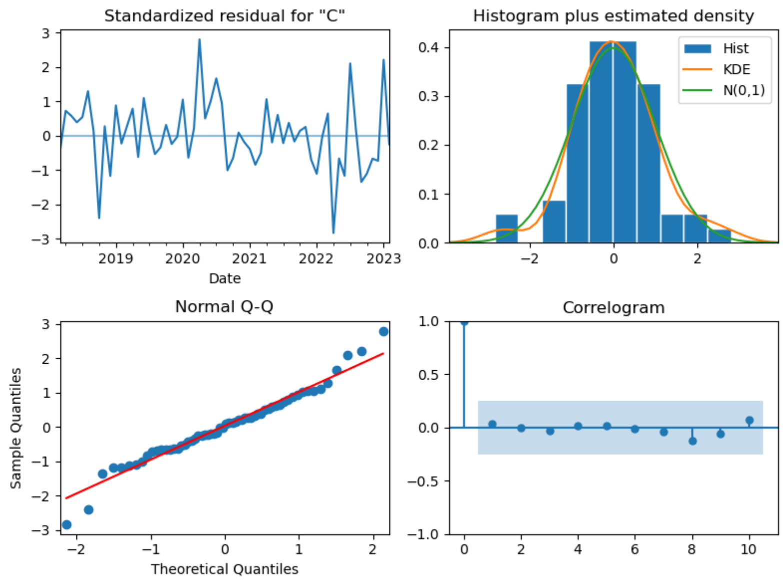
Implementation:

- 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 (MA) components captured the impact of residual shocks and unexpected fluctuations in the data.

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.

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 Tell Us :

Traditional Methods Prevail

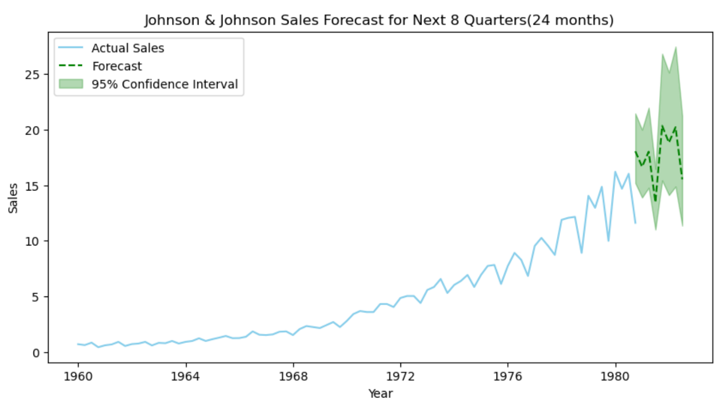
- The ARMA (1,1) model delivered exceptional accuracy (RMSE: 0.40, MAPE <10%), efficiently capturing JJ's steady growth and seasonal patterns. Its simplicity proved ideal for this well-structured dataset.

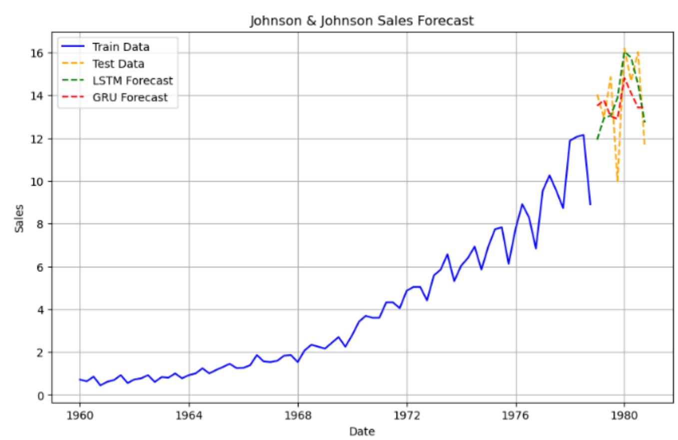
Neural Networks: Overengineered Solution?

- While LSTM (RMSE: 1.86) and GRU (RMSE: 1.77) demonstrated technical feasibility, their performance lagged behind ARMA. The higher errors suggest these complex architectures struggled with JJ's relatively predictable patterns - a classic case of "using a sledgehammer to crack a nut."

Key Takeaway

- For traditional business metrics like JJ's sales data, simpler statistical models often outperform sophisticated deep learning approaches. The results reinforce that model selection should be driven by data characteristics rather than technical novelty.





Amazon Stock Price Forecasts Tell Us:

1. ARMA Model: Reliable in Calm Waters

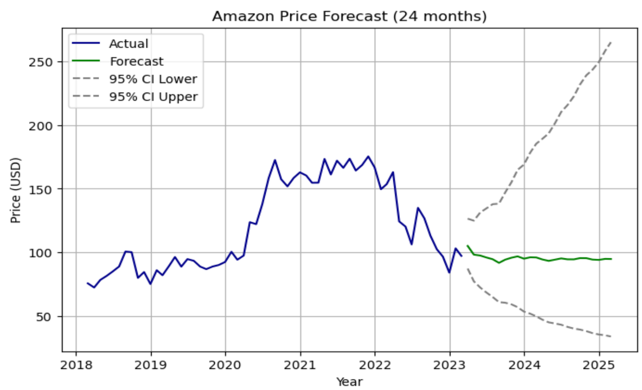
- Demonstrated strong accuracy (MAPE <10%) during stable market conditions
- Struggled with volatility spikes, with RMSE increasing significantly during financial shocks
- Limitations exposed in rapidly changing market environments

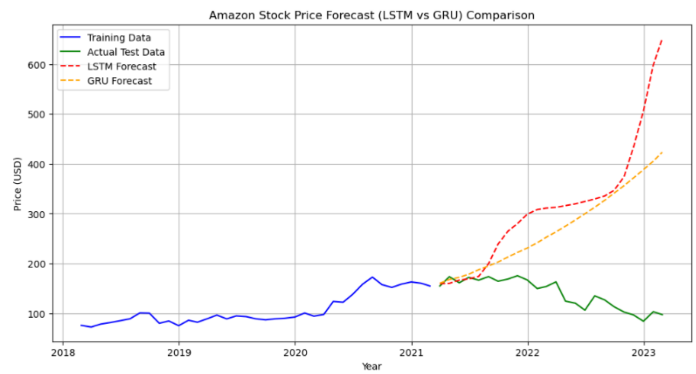
2. Deep Learning Challenges

- LSTM/GRU performance disappointing (MAPE >100%)
- Failed to capture nonlinear volatility and complex market dynamics
- GRU slightly outperformed LSTM, suggesting simpler architectures may handle volatility better

3. Critical Insight

- No single model dominated across all market conditions
- ARMA's interpretability valuable for stable periods
- Deep learning's potential remains unrealized for this volatile asset
- Hybrid approaches may be worth exploring for future research





Improvements:

1. Johnson & Johnson Sales Forecasting

- Hybrid Approach: Combine ARMA (linear trends) with simplified LSTM/GRU (nonlinear patterns) to balance accuracy and complexity.
- Prevent Overfitting: Streamline deep learning architectures to avoid excessive parameterization on smaller datasets.
- Optimal Baseline: ARMA remains the gold standard for stable, seasonal data deep learning adds marginal gains without external variables.

2. Amazon Stock Price Forecasting

- Volatility Modelling: Integrate GARCH for dynamic risk assessment, paired with Z-score normalization to stabilize extreme price swings.

Deep Learning Refinements:

- Hybrid Architectures: Merge LSTM/GRU with attention mechanisms or exogenous data (sentiment, macros).
- Probabilistic Forecasting: Use quantile loss or Bayesian layers to quantify uncertainty in volatile regimes.

3. Universal Best Practices

- Hyperparameter Optimization: Rigorous tuning (e.g., Bayesian optimization) to align model capacity with data complexity.

Right-Fit Modelling:

- Stable Data (e.g., JJ): Leverage interpretable models (ARMA, ETS).
- Volatile Data (e.g., AMZN): Deploy adaptive hybrids (GARCH + LSTM) with external inputs.

Conclusion:

- **Johnson & Johnson Sales**: ARMA models are well-suited due to their simplicity and reliable performance under stable conditions. In contrast, LSTM/GRU models may be unnecessarily complex without additional contextual inputs.
- **Amazon Stock: LSTM/GRU** models show promise for forecasting in volatile markets but require enhancements such as hybrid structures or incorporation of external factors (e.g., sentiment or macroeconomic indicators) to effectively manage high variability.
- **Overall Strategy**: Align model complexity with data characteristics opt for simpler models for stable, predictable series, and adopt advanced models for dynamic, nonlinear datasets where complexity adds value.

References

- Cho, K., et al. (2014). Learning Phrase Representations using RNN Encoder-Decoder. *arXiv*.
- Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). *Time Series Analysis: Forecasting and Control*. Wiley.
- 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*.
- Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*. OTexts.