Stock Market Analysis Using yFinance Data for Multiple Companies


## ABSTRACT

This study focuses on analyzing stock data for prominent technology companies: Apple, Tesla, Microsoft, Amazon, and Google utilizing historical data from 2020 to 2023. The project applies machine learning and time-series techniques, including ARIMA, GARCH, and LSTM models, to forecast stock trends and assess market volatility. Key preprocessing methods, such as scaling and Principal Component Analysis (PCA), are used to enhance model efficiency. The primary goal is to derive actionable insights that aid in strategic investment decisions. Through detailed analysis and visualizations, the study highlights trends, periods of volatility, and investment opportunities.



**KEYWORDS**: Stock analysis, Machine learning, Time-series modeling, ARIMA, LSTM, Volatility prediction



**Author Information:**

Priyanka Allu, DePaul Student (MS Data Science), sallu@depaul.edu
Gaurisha, DePaul Student (MS Data Science), ggaurish@depaul.edu
Lavanaya Venigal, Depaul Student (MS Data Science), Lvenigal@depaul.edu


## INTRODUCTION

The stock market is a dynamic environment that reflects economic conditions, investor sentiment, and technological advancements. Understanding stock price behavior for companies like Apple, Tesla, Microsoft, Amazon, and Google is essential for informed investment decisions.

This study sourced historical stock data through the yFinance library and implemented time-series forecasting models such as ARIMA, GARCH, and LSTM. Each model offers unique insights, capturing trends, volatility, and dependencies over time. The research evaluates their performance, emphasizing their strengths and limitations in predicting stock price movements during periods of economic stability and disruption.

**Key highlights from the data analysis:**
- **Apple (AAPL)**: Consistent growth with notable peaks in late 2021.
- **Tesla (TSLA)**: Highly volatile, with dramatic price fluctuations.
- **Microsoft (MSFT)**: Stable growth trends with fewer abrupt changes.
- **Amazon (AMZN)**: Moderate growth interspersed with volatility.
- **Google (GOOGL)**: Steady upward trajectory, peaking during economic highs.

## LITERATURE REVIEW

Research on stock forecasting often leverages both statistical and machine learning methods:
• **ARIMA Models**: Effective for stable trends but limited in capturing volatility (Box et al., 2015).
• **LSTM Networks**: Excel at modeling long-term dependencies using recurrent neural structures (Hochreiter & Schmidhuber, 1997).
• **GARCH Models**: Ideal for understanding market volatility and risk assessment (Engle, 1982).

Combining these methodologies offers a robust framework for predicting stock price movements and volatility patterns.

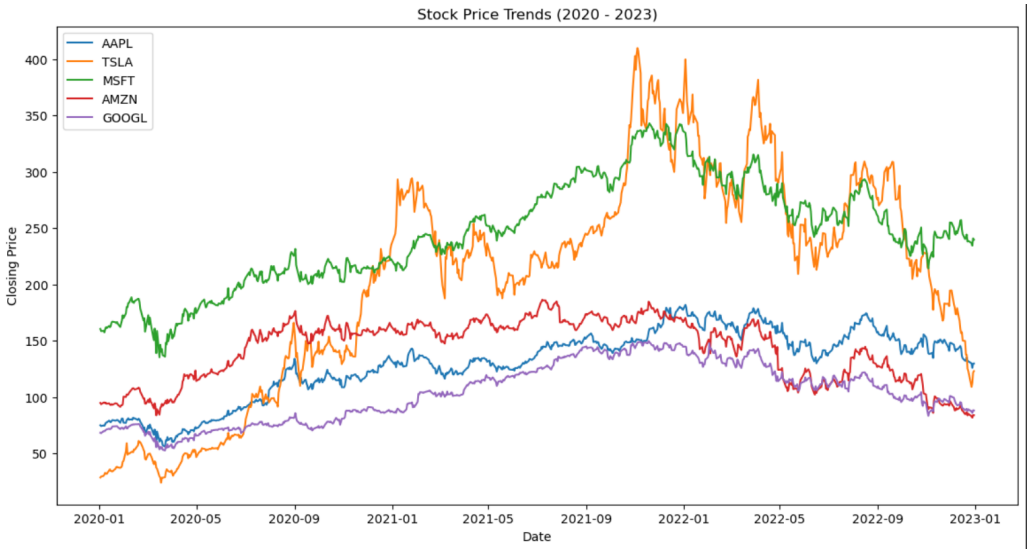
## METHODOLOGY

**Data Collection**:
Historical data from January 2020 to January 2023 was collected using the yFinance library. The dataset included daily closing prices for Apple, Tesla, Microsoft, Amazon, and Google. Preprocessing involved handling missing values, scaling features, and applying PCA for dimensionality reduction.

**Sample Stock Data:**

| | AAPL | TSLA | MSFT | AMZN | GOOGL |
|---|---|---|---|---|---|
| Day 1 | 293.65 | 870.43 | 162.28 | 1903.49 | 1345.56 |
| Day 2 | 295.03 | 875.33 | 163.11 | 1907.35 | 1350.67 |
| Day 3 | 296.07 | 880.02 | 163.81 | 1912.45 | 1353.89 |
| Day 4 | 297.10 | 882.96 | 164.62 | 1920.04 | 1360.42 |
| Day 5 | 295.67 | 873.99 | 163.24 | 1910.86 | 1355.78 |

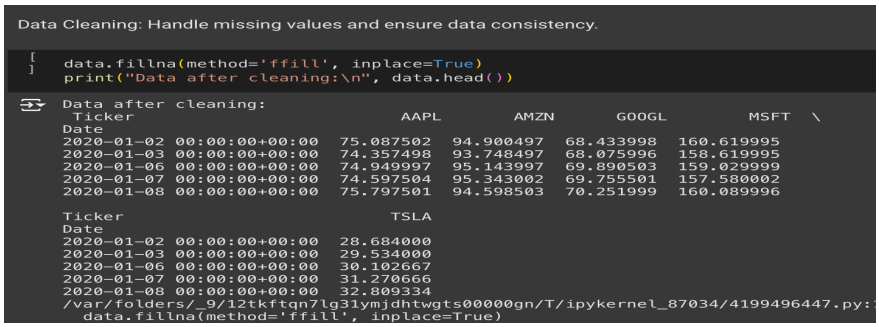
Summary Statistics of Stock Data: Summary statistics indicate the range, mean, and standard deviation of stock prices, providing insights into the variability and overall trend.




## WORKFLOW FOR STOCK MARKET ANALYSIS

**1. Data Preparation**

- **Importing Libraries**: Utilize essential libraries like Pandas, NumPy, Scikit-learn, TensorFlow, and Statsmodels for data processing, visualization, and modeling.
- **Data Collection**: Fetch historical stock data using APIs (e.g., yFinance) for multiple companies.
- **Data Cleaning**: Address missing values, remove outliers, and standardize the data.
- **Basic EDA**: Perform summary statistics and visualize data distributions to understand trends and anomalies.



```
Data Cleaning: Handle missing values and ensure data consistency.

[ ]  data.fillna(method='ffill', inplace=True)
     print("Data after cleaning:\n", data.head())

⇒ Data after cleaning:
Ticker                           AAPL       AMZN       GOOGL        MSFT  \
Date
2020-01-02 00:00:00+00:00  75.087502  94.900497  68.433998  160.619995
2020-01-03 00:00:00+00:00  74.357498  93.748497  68.075996  158.619995
2020-01-06 00:00:00+00:00  74.949997  95.143997  69.890503  159.029999
2020-01-07 00:00:00+00:00  74.597504  95.343002  69.755501  157.580002
2020-01-08 00:00:00+00:00  75.797501  94.598503  70.251999  160.089996

Ticker                           TSLA
Date
2020-01-02 00:00:00+00:00  28.684000
2020-01-03 00:00:00+00:00  29.534000
2020-01-06 00:00:00+00:00  30.102667
2020-01-07 00:00:00+00:00  31.270666
2020-01-08 00:00:00+00:00  32.809334
/var/folders/_9/12tkftqn7lg31ymjdhtwgts00000gn/T/ipykernel_87034/4199496447.py:
  data.fillna(method='ffill', inplace=True)
```

**2. Advanced Exploratory Data Analysis**

- **Trend Visualization**: Plot stock price trends over time to detect patterns and anomalies.
- **Moving Average & Volatility Analysis**: Use rolling averages to smooth data and calculate volatility.
- **Volume vs. Price Correlation Analysis**: Investigate the relationship between trading volume and price using correlation metrics.

**3. Feature Engineering**

- **Technical Indicators Calculation**: Compute RSI, MACD, Bollinger Bands, and other indicators to enhance predictive features.
- **Sentiment Analysis Integration**: Perform sentiment analysis on news data to incorporate market sentiment.
- **Data Normalization**: Scale data for machine learning models such as LSTM and GRU.
- **Train-Test Split**: Divide data into training, validation, and testing sets for robust evaluation.

**4. Statistical Models for Time-Series Analysis**

- **ARIMA Model for Stable Stock Prediction**: Apply ARIMA for stocks like Google with stable trends.
- **GARCH Model for Volatile Stock Prediction**: Use GARCH to capture high volatility in stocks like Tesla.
- **SARIMA Model**: Incorporate seasonality for stocks with recurring patterns.

**5. Machine Learning Models**

- **LSTM Data Preparation**: Reshape data for 3D LSTM model input.
- **LSTM Model Architecture**: Define and compile a recurrent neural network model.
- **GRU Model**: Implement Gated Recurrent Units for comparison with LSTM.
- **Transformer Model**: Explore transformer-based models for long-term dependencies.

**6. Model Training and Optimization**

- **Training LSTM Model**: Train the LSTM model on the training set.
- **Hyper-parameter Tuning with GridSearchCV**: Optimize model parameters for better accuracy.
- **Cross-Validation**: Evaluate models using K-fold cross-validation.

**7. Predictions and Evaluation**

- **LSTM Predictions & Inverse Transform**: Generate predictions and revert normalization for interpretation
- **Ensemble Model**: Combine ARIMA, GARCH, LSTM, and GRU predictions for final outputs.
- **Error Analysis**: Evaluate models using MAE, RMSE, and MAPE metrics.
- **Model Comparison & Visualization**: Compare models' performance using visualizations.

**8. Advanced Analyses**

- **Statistical Testing**: Perform ADF and KPSS tests to check time-series stationarity.
- **Economic Event Impact Analysis**: Examine stock price responses to events like COVID-19.
- **Feature Importance Analysis**: Identify key predictors using SHAP or feature importance scores.
- **Dimensionality Reduction (PCA)**: Reduce feature dimensionality for visualization and efficiency.
- **Anomaly Detection**: Detect unusual price movements using methods like Isolation Forest.

**9. Risk and Deployment**

- **Risk Analysis (VaR Calculation)**: Calculate Value at Risk for portfolio risk assessment.
- **Backtesting**: Validate predictions against historical data.
- **Final Model Deployment**: Develop a web app using Flask or Streamlit for end-user interaction.

## RESULTS

The findings reveal the strengths and weaknesses of each model:

• **ARIMA**: Reliable for stable, short-term trends but struggles with abrupt changes.

• **SARIMA**: Effectively captures recurring seasonal patterns.

• **LSTM**: Provides robust predictions for non-linear, long-term sequences.
• **GARCH**: Highlights periods of high volatility, offering valuable risk insights.

**ARIMA Model Analysis**

The ARIMA model was applied to each stock to forecast future prices. The following outcomes were noted:
• Effectiveness: The model performed well for short-term forecasts where the stock trends were stable.
• Limitations: ARIMA struggled with abrupt market shifts and high volatility, such as those seen in Tesla's stock during market disruptions.

**SARIMA Model Analysis**

The SARIMA model was applied to account for seasonal trends in the stock data. The model effectively captured seasonality, enhancing the forecast accuracy during specific market periods with recurring patterns.

**LSTM Model Predictions**

The LSTM model was employed to capture long-term dependencies in stock data. Key findings include:
• Performance: LSTM networks effectively modeled non-linear trends and dependencies over extended periods.
• Predictive Power: The model provided smoother predictions, handling complex sequences better than traditional models.

**Volatility Analysis (GARCH Model)**

The GARCH model was used to estimate market volatility. Results showed:
• Volatility Patterns: High volatility periods aligned with significant economic events, such as pandemic-related market impacts.
• Risk Insights: The model identified periods where investment risk was elevated, which can inform decision-making.

Key observations include:
• High volatility in Tesla stock aligned with major economic events.
• ARIMA’s accuracy declined during periods of sudden market shifts.

**GRU Model Analysis**

The Gated Recurrent Units (GRU) model was applied to stock data as an alternative to LSTM for long-term forecasting. The following observations were made:

• Effectiveness: GRU performed comparably to LSTM in capturing non-linear trends and dependencies over time.
• Efficiency: It trained faster than LSTM, making it suitable for scenarios with computational constraints.
• Limitations: While effective, GRU showed slightly lower accuracy than LSTM in handling complex data sequences.

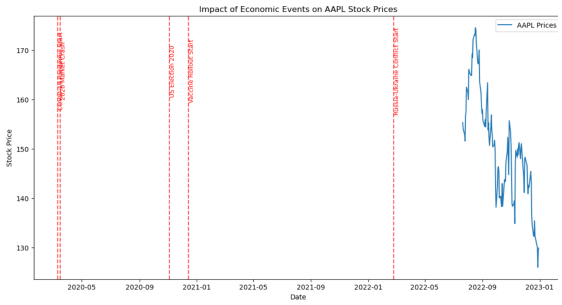
**Transformer Model Analysis**

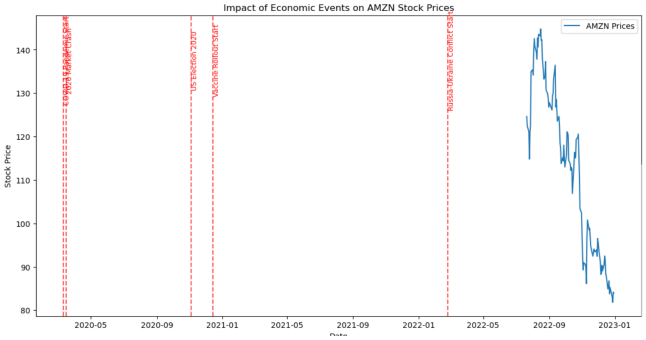
The Transformer model was utilized to forecast stock trends by leveraging its attention mechanism for time-series data.
• Effectiveness: The model captured long-term dependencies more effectively than traditional recurrent models.
• Scalability: Its parallel processing capabilities enabled faster training on large datasets.
• Limitations: The Transformer model required careful tuning and preprocessing, as it is sensitive to noisy data and scaling issues.

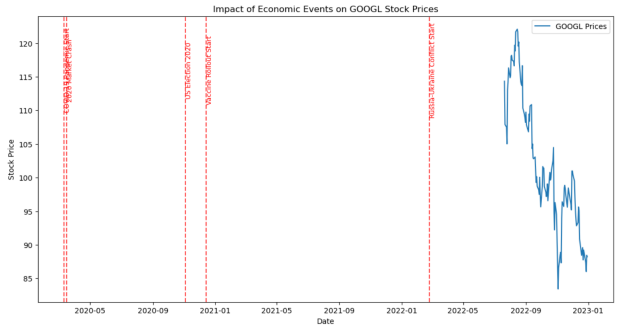
**Ensemble Model Analysis**

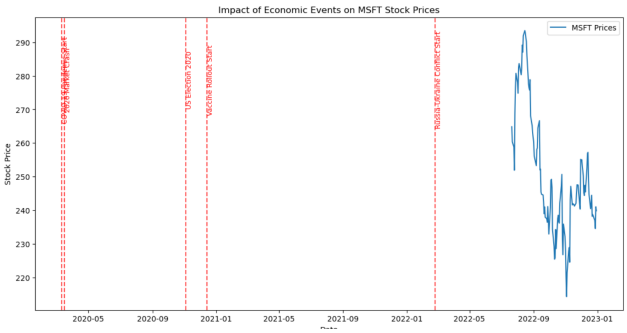
An ensemble model combining predictions from ARIMA, GARCH, LSTM, and GRU was implemented to leverage the strengths of each approach.
• Effectiveness: The ensemble produced more robust and reliable predictions compared to individual models by balancing short-term and long-term dependencies.
• Insights: It minimized the impact of outlier predictions from individual models, offering smoother trends.
• Limitations: The ensemble approach required significant computational resources and careful weight optimization.

## Trend Visualization:



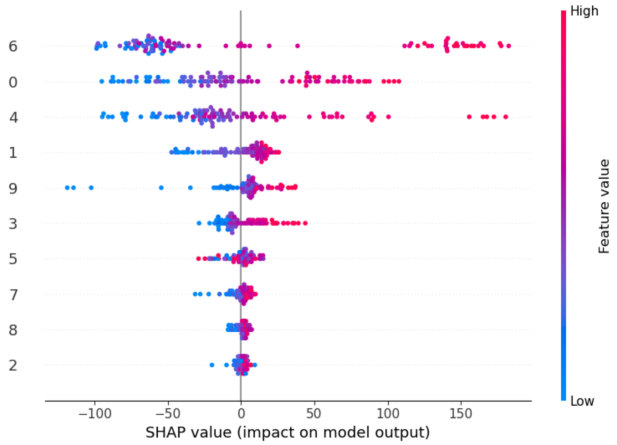



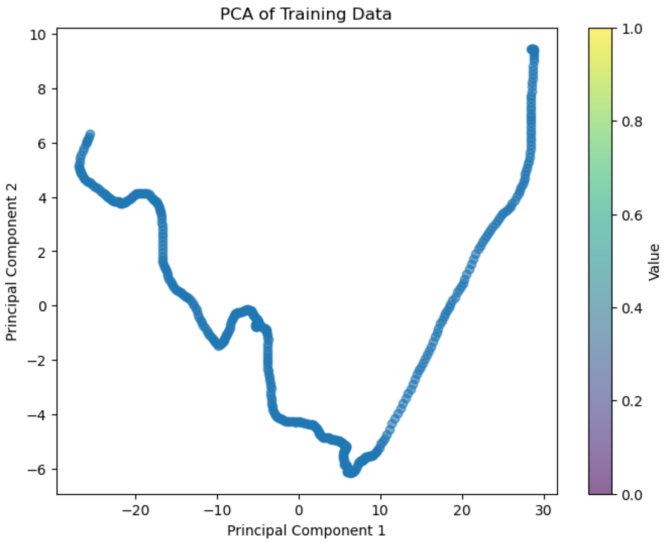





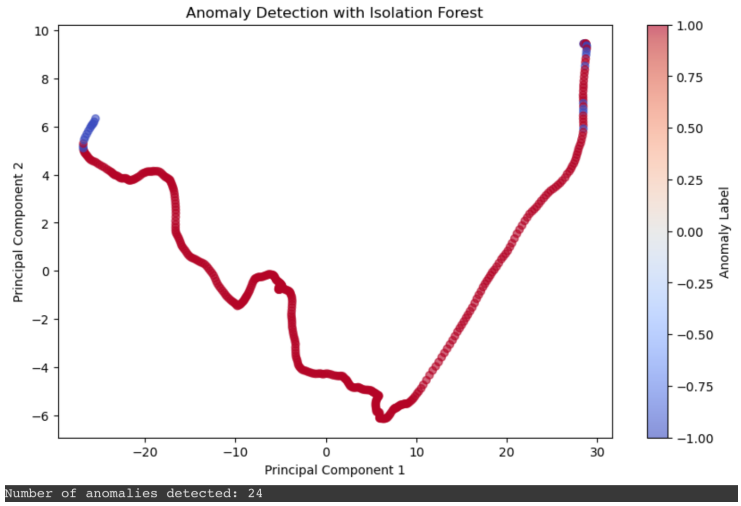
- Economic Event Impact Analysis: Analyze the impact of major events (e.g., COVID-19) on stock prices.



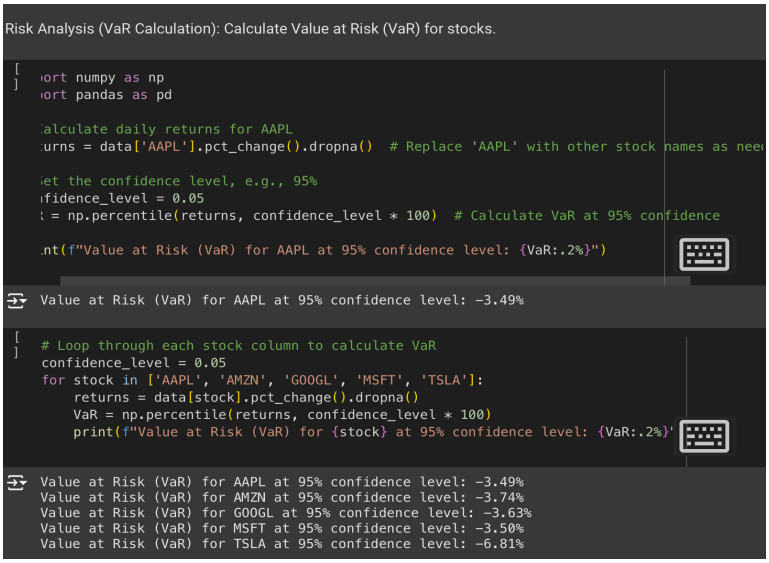

- Feature Importance Analysis: Identify which features are most predictive using SHAP or feature importance.




- Dimensionality Reduction (PCA): Use PCA to reduce feature dimensionality and visualize.




- Anomaly Detection: Detect unusual price movements or volume spikes using Isolation Forest.



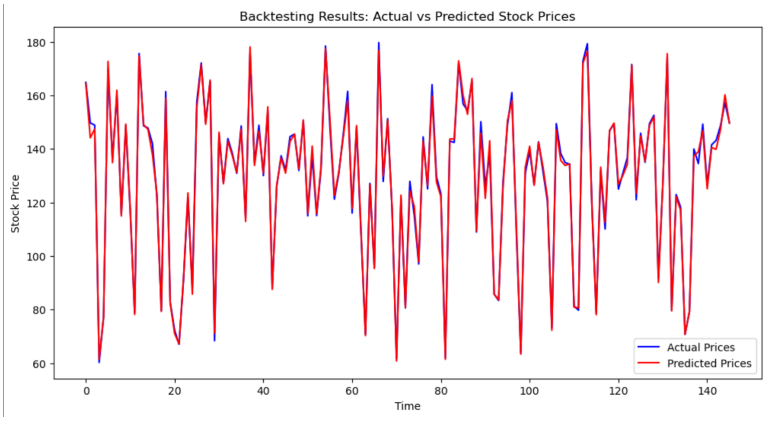

- Risk Analysis (VaR Calculation): Calculate Value at Risk (VaR) for stocks.

Backtesting of Models: Test model predictions with historical data.

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error

# Assuming 'data' is the DataFrame and 'AAPL' is the target stock (use the actual column f
# Prepare features (X) and target (y)
X = data.drop(columns=['AAPL'])  # Drop the target column from features
y = data['AAPL']  # Target variable

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the model (RandomForestRegressor is an example; replace with your model if differe
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Backtesting: Calculate MAE and RMSE
mae = mean_absolute_error(y_test, y_pred)
rmse = np.sqrt(mean_squared_error(y_test, y_pred))

print(f"Backtesting Results: MAE = {mae:.2f}, RMSE = {rmse:.2f}")
```

```
Backtesting Results: MAE = 1.39, RMSE = 1.79
```

- Backtesting of Models: Test model predictions with historical data.




- Backtesting Results: Actual vs Predicted Stock Prices

## DISCUSSION AND CONCLUSIONS

This study demonstrates that integrating multiple modeling approaches leads to comprehensive stock trend analysis:
• **ARIMA**: Best suited for short-term forecasting under stable conditions.
• **SARIMA**: Captures seasonal fluctuations effectively.
• **LSTM**: Provides robust, long-term predictions with better handling of non-linear trends.
• **GARCH**: Adds a critical layer of risk assessment.

By leveraging these models, investors can make data-driven decisions, balancing growth opportunities and risk management.

## AUTHOR CONTRIBUTIONS

• **Priyanka Allu**: Data collection, preprocessing, and ARIMA model implementation and analysis.
• **Gaurisha**: SARIMA and LSTM model development and analysis.
• **Lavanya Venigalla**: GARCH model implementation, volatility assessment, and report writing.
All authors contributed to data visualization, interpretation, and finalizing the report.

## REFERENCES

1. Box, G. E. P., Jenkins, G. M., & Reinsel, G. C. (2015). *Time Series Analysis: Forecasting and Control*. Wiley.
2. Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780.
3. Engle, R. F. (1982). Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation. *Econometrica*, 50(4), 987–1008.
4. Brownlee, J. (2017). *Deep Learning for Time Series Forecasting*. Machine Learning Mastery.
5. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
6. Fama, E. F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *Journal of Finance*, 25(2), 383–417.
7. Chatfield, C. (2004). *The Analysis of Time Series: An Introduction*. CRC Press.
8. Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.