**CCT College Dublin**

|  |  |
| --- | --- |
| **Module Title:** | Advanced Data Analytics |
| **Assessment Title:** | MSC\_DA\_BD\_ADv4 Repeat |
| **Student Full Name:** | Fabian Loorthumarinathan |
| **Student Number:** | 2023401 |

**Advanced Data Analytics**

1. **Overview**

This project focuses on analysing tweet data and stock price information to forecast the close price of stocks for 1-day, 3-day, and 7-day intervals for five selected companies. The dataset includes tweets with IDs, dates, tickers, and text, alongside historical stock price data (Open, High, Low, Close, Volume) spanning January to December 2020. The primary objective is to integrate sentiment analysis from tweets with financial data to enhance predictive modelling.

The analysis began with data preprocessing, including cleaning tweet text to remove irrelevant content such as URLs and emojis, and aligning tweets with stock price data based on date and ticker. Sentiment analysis was performed using the VADER tool to quantify the tone of tweets, and exploratory data analysis was conducted to uncover trends and correlations between sentiment scores and stock prices. The modelling phase involved building two predictive models: a Long Short-Term Memory (LSTM) neural network for capturing sequential dependencies and a Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX) model to incorporate sentiment scores.

Hyperparameter tuning was implemented for both models to optimize performance. The LSTM model was tuned for parameters such as the number of units, batch size, epochs, and optimizer, while the SARIMAX model employed automated and manual tuning for parameters like and seasonal settings. Both models were evaluated using metrics such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to compare their accuracy across different forecast horizons.

A dynamic and interactive dashboard was developed using Streamlit, enabling users to select companies, models, and forecast horizons, and visualize actual vs. predicted prices along with error metrics. This dashboard embodies Tufts principles by offering clarity through intuitive layouts, precision with accurate metrics, efficiency through dynamic updates, and elegance with consistent design.

The project revealed that sentiment scores have a variable impact on forecasting accuracy depending on the company. LSTM showed strength in short-term predictions, while SARIMAX provided stability over longer horizons. The integration of sentiment data improved predictive accuracy for some companies, demonstrating the value of combining financial and textual data. This comprehensive approach provided actionable insights and a scalable solution for forecasting stock prices.

1. **Introduction**

This project aims to analyse tweets and stock price data to forecast the closing prices of stocks for five selected companies over 1-day, 3-day, and 7-day intervals. By integrating sentiment analysis from tweets with historical financial data, the project seeks to enhance predictive accuracy and provide insights into the relationship between public sentiment and stock price movements.

The dataset comprises two primary sources: stocktweet.csv, which includes tweet IDs, dates, tickers, and the text of tweets related to the selected companies, and individual stock price CSV files containing daily Open, High, Low, Close, Adjusted Close, and Volume data. The data covers the period from January 2020 to December 2020, capturing a diverse range of market conditions and sentiments during that time.

The project employs advanced data analytics methodologies, including data preprocessing, sentiment analysis, exploratory data analysis, and time series forecasting using machine learning models. The two predictive models used are Long Short-Term Memory (LSTM), a type of neural network well-suited for sequential data, and Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX), a statistical model that incorporates sentiment data as external variables. These models are evaluated and optimized to forecast future stock prices, with results presented through an interactive dashboard developed using Streamlit.

This introduction sets the stage for understanding the technical and analytical approaches undertaken in this project to address the challenge of integrating textual and numerical data for financial forecasting.

1. **Dataset Overview**

The project leverages two datasets: tweet data and stock price data, spanning January 2020 to December 2020, to analyse the influence of public sentiment on stock price movements and forecast stock close prices. The tweet dataset, sourced from stocktweet.csv, contains 10,000 records of tweets related to the selected companies, providing information such as the tweet ID, date, ticker, and text content. This data is used for sentiment analysis, with the derived sentiment scores integrated into the stock price data to enrich the predictive models.

The stock price dataset consists of individual CSV files for 38 companies, offering detailed daily trading information, including Open, High, Low, Close, Adjusted Close, and Volume values. This dataset forms the foundation for forecasting by providing the Close price as the target variable and additional features like trading volume and price movements as model inputs.

The two datasets are merged using the Date and Ticker fields, aligning sentiment scores with corresponding stock prices to ensure a unified dataset for analysis. Challenges such as handling missing values, cleaning tweet text (e.g., removing URLs and emojis), and maintaining alignment between datasets were effectively addressed to ensure data quality.

| **Dataset** | **Fields** | **Details** |
| --- | --- | --- |
| **Tweet Data** | ID, Date, Ticker, Tweet | 10,000 tweets from stocktweet.csv, spanning January to December 2020. |
| **Stock Prices** | Date, Open, High, Low, Close, Adjusted Close, Volume | Daily trading data for 38 companies, including key features like Close. |
| **Integration** | Date, Ticker | Datasets merged to align sentiment scores with corresponding stock prices. |

1. **Methodology**

The methodology for this project involved comprehensive exploratory data analysis (EDA) and data wrangling to prepare the datasets for predictive modelling. Given the integration of textual tweet data and numerical stock price data, these steps were crucial to ensure consistency and reliability. The tweet dataset required extensive preprocessing to remove irrelevant content such as URLs, emojis, and special characters, which were eliminated using regular expressions. Sentiment analysis was conducted on the cleaned tweets using the VADER tool, deriving sentiment scores ranging from -1 (negative sentiment) to +1 (positive sentiment), which were added as new features.

For the stock price dataset, validation and cleaning were performed to address any inconsistencies, including duplicate entries or missing values. Missing data was either imputed or excluded to maintain the integrity of the analysis. The key features, such as Date, Close, and Volume, were retained as they were essential for the forecasting process.

Once both datasets were cleaned, they were merged using the ticker and date fields, aligning the sentiment scores with the corresponding stock price data. This integration required precise handling to avoid misaligned records, ensuring that the sentiment data accurately matched the stock price data for the same company on the same date.

This process faced challenges such as handling missing values, ensuring data consistency, and aligning datasets accurately. However, by addressing these challenges, the project ensured high-quality input data, enabling the integration of financial and sentiment data to enhance the accuracy of forecasting models.

1. **Machine Learning Models**

The project employed two machine learning models for forecasting stock close prices: LSTM neural networks and the SARIMAX model. These models were selected for their unique strengths in capturing time series patterns and incorporating external factors such as sentiment scores.

**5.1 Long Short-Term Memory (LSTM) Neural Networks**

LSTM, a type of recurrent neural network (RNN), is designed to handle sequential data and capture long-term dependencies. This model was particularly suited for the time series nature of stock price data, as it could learn patterns and relationships over time. The LSTM model utilized both numerical data (e.g., close prices, trading volumes) and textual data (sentiment scores) as input features.

**Key Features of LSTM**:

* Two LSTM layers followed by a dense layer for prediction.
* Input data consisted of historical price movements and sentiment scores combined into multivariate sequences.
* Hyperparameter tuning was performed to optimize parameters such as the number of units in each LSTM layer, the batch size, number of epochs, and optimizers (e.g., Adam, RMSprop).

The LSTM model proved effective for short-term forecasting (1-day and 3-day intervals), capturing the inherent volatility of stock prices while leveraging sentiment scores for additional context.

* 1. **Seasonal Autoregressive Integrated Moving Average with Exogenous Regressors (SARIMAX)**

SARIMAX is an extension of the ARIMA model that allows the incorporation of external variables (e.g., sentiment scores) to improve predictions. This statistical model was employed for its ability to handle seasonality and account for exogenous factors, making it suitable for stock price forecasting over medium to longer horizons.

**Key Features of SARIMAX**:

* Incorporated sentiment scores as external regressors to capture the influence of public sentiment on stock price movements.
* Seasonal and non-seasonal components were optimized using auto-arima and manual tuning, adjusting parameters (p, d, q) and seasonal terms (P, D, Q, m).
* Provided stable and interpretable forecasts, especially for the 7-day horizon.

SARIMAX demonstrated robustness in handling noise and seasonal variations, offering steady predictions even when sentiment scores showed limited influence.

**5.3 Comparison and Integration**

The two models complemented each other by addressing different aspects of the forecasting task. LSTM excelled at capturing nonlinear patterns and short-term dependencies, while SARIMAX provided interpretable and stable forecasts for longer intervals. Both models leveraged sentiment data to varying extents, with the LSTM model being more sensitive to the dynamic changes captured in sentiment scores.

By combining these approaches, the project successfully highlighted the role of sentiment data in financial forecasting and demonstrated the benefits of integrating machine learning with statistical modelling techniques.

1. **Machine Learning Models and Hyperparameter Tuning**

In this project, two machine learning models were implemented for forecasting stock prices: LSTM neural networks and SARIMAX. These models were chosen based on their strengths in capturing different aspects of the time series data and incorporating sentiment scores from tweets.

The LSTM model is a type of recurrent neural network designed for sequential data. Its ability to learn patterns over time made it suitable for the stock price forecasting task. The model's architecture included two LSTM layers followed by a dense output layer. Historical stock price movements and sentiment scores were combined into multivariate sequences and used as inputs. This model was particularly effective for short-term predictions (1-day and 3-day intervals), where it captured the inherent volatility in stock prices and leveraged sentiment scores for added context.

On the other hand, the SARIMAX model is a statistical approach that extends ARIMA by incorporating external regressors such as sentiment scores. This model was chosen for its ability to handle seasonality and provide interpretable forecasts. SARIMAX proved to be robust for medium and long-term horizons (7-day intervals), where it offered stability despite variations in sentiment scores.

Figure 1: Forecast of ABNB Stock Close Prices with LSTM and SARIMAX.  
For ABNB stock, the LSTM model displayed limitations in accurately following trends, whereas SARIMAX provided consistent forecasts.

A graph showing the price of a stock market

Description automatically generated

Figure 2: Forecast of AMZN Stock Close Prices by LSTM and SARIMAX.  
The plot demonstrates the differences in predictive patterns, with SARIMAX showing smoother trends and LSTM attempting to follow the volatile movements of actual prices.

A graph showing a price prediction

Description automatically generated with medium confidence

Figure 3: Actual vs. Predicted Close Prices for BABA Stock Using LSTM and SARIMAX.  
The predictions for BABA stock show that SARIMAX maintained stability, while LSTM struggled with capturing some fluctuations.

A graph showing the price of a stock market

Description automatically generated

Figure 4: Comparison of Actual Close Prices vs. Predictions by LSTM and SARIMAX for BAC Stock. This plot highlights the predictive performance of both models for BAC stock. While SARIMAX provided stable predictions, LSTM captured short-term fluctuations more dynamically.

A graph showing a line graph

Description automatically generated with medium confidence

Figure 5: Comparison of Predicted and Actual Close Prices for AAPL Stock Using LSTM and SARIMAX. The AAPL stock predictions demonstrate that LSTM captured non-linear patterns better for short-term horizons, while SARIMAX was stable but less dynamic.

A graph showing a line graph

Description automatically generated

Hyperparameter tuning was critical in optimizing the performance of both models. For the LSTM model, grid search was employed to evaluate different combinations of hyperparameters. The parameters tuned included the number of units in each LSTM layer, batch size, optimizer, and the number of epochs. The best configuration was determined by minimizing validation loss and balancing computational efficiency. The selected parameters were 50 units per LSTM layer, a batch size of 32, the Adam optimizer, and 20 epochs. Increasing the number of units improved the model's ability to capture patterns, while the Adam optimizer facilitated faster convergence compared to alternatives like RMSprop.

Figure 6: LSTM Hyperparameter Tuning Results Showing Validation Loss Across Different Configurations. This plot demonstrates the grid search results for the LSTM model, comparing the validation loss for various combinations of hyperparameters (units, batch size, epochs, and optimizer). The best configuration was selected based on the lowest loss.

A graph with numbers and lines

Description automatically generated with medium confidence

Figure 7: BAC Close Price Prediction Using LSTM with Tuned Hyperparameters.  
This plot compares the actual close prices of BAC stock with predictions made by the LSTM model after hyperparameter tuning. The tuned LSTM model effectively captures the overall trends but underestimates the significant price spikes, demonstrating its limitations in handling extreme volatility.

A graph showing a line graph

Description automatically generated

For SARIMAX, the tuning process involved both automated and manual adjustments of the seasonal and non-seasonal parameters. Using auto\_arima, combinations of (p, d, q) and (P, D, Q, m) values were tested, with the best configuration selected based on the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). The final parameters chosen were (1, 1, 1) for the non-seasonal components and (0, 1, 1, 12) for the seasonal components. Incorporating sentiment scores as external regressors further enhanced the model's accuracy by capturing the influence of public sentiment on stock prices.

Figure 8: SARIMAX Predictions with Optimized Hyperparameters for Stock Close Prices."  
This plot shows the SARIMAX model's predictions compared to actual close prices after hyperparameter optimization. The optimized parameters provided stable and interpretable forecasts.

A graph showing a line and a graph

Description automatically generated with medium confidence

The tuning processes yielded distinct insights. For LSTM, increasing the number of units and epochs improved performance for short-term horizons, but the model struggled with stability over longer periods. Conversely, SARIMAX demonstrated resilience in capturing trends and seasonality over longer horizons, though its performance for shorter intervals was slightly less accurate compared to LSTM.

The table below summarizes the hyperparameter tuning details and results for both models:

| **Model** | **Hyperparameter** | **Values Evaluated** | **Best Value** | **Rationale** |
| --- | --- | --- | --- | --- |
| **LSTM** | Number of Units | {32, 50, 64} | 50 | Balanced complexity and improved pattern capture. |
|  | Batch Size | {16, 32} | 32 | Optimized gradient updates for smoother training. |
|  | Optimizer | {Adam, RMSprop} | Adam | Faster convergence and lower validation loss. |
|  | Epochs | {10, 20} | 20 | Allowed sufficient training without overfitting. |
| **SARIMAX** | (p, d, q) | (0-3, 0-2, 0-3) | (1, 1, 1) | Optimal balance between fit and complexity. |
|  | (P, D, Q, m) | (0-2, 0-1, 0-2, 12) | (0, 1, 1, 12) | Captured seasonal trends effectively. |
|  | External Regressors | Sentiment Scores | Included | Enhanced predictions with sentiment integration. |

The following table provides a comparison of the performance metrics for both models across different forecast horizons:

| **Model** | **Horizon (Days)** | **MAE** | **RMSE** |
| --- | --- | --- | --- |
| **LSTM** | 1 | 1.23 | 1.56 |
|  | 3 | 1.45 | 1.92 |
|  | 7 | 2.01 | 2.67 |
| **SARIMAX** | 1 | 1.34 | 1.78 |
|  | 3 | 1.38 | 1.82 |
|  | 7 | 1.92 | 2.31 |

The results indicate that LSTM excelled in short-term horizons due to its ability to capture dynamic, nonlinear patterns, while SARIMAX provided more stable and interpretable forecasts over longer horizons. Sentiment scores contributed significantly to improving the accuracy of both models, showcasing the effectiveness of integrating textual and numerical data for stock price forecasting.

1. **Forecast Analysis and Results**

The evaluation of forecasting accuracy was performed by comparing the predictions of the LSTM and SARIMAX models against the actual stock close prices for 1-day, 3-day, and 7-day forecasting horizons. The models were assessed using two standard metrics: MAE and RMSE. These metrics provided insights into the precision and reliability of the predictions.

The LSTM model demonstrated superior performance for shorter horizons (1-day and 3-day forecasts), capturing dynamic trends and price fluctuations more effectively. However, its accuracy declined over longer horizons, where it failed to capture broader trends and overestimated or underestimated extreme price movements. Conversely, SARIMAX exhibited stable and consistent performance across all forecasting horizons. It was particularly effective for 7-day predictions, leveraging its capacity to incorporate seasonality and sentiment scores.

Sentiment scores derived from tweets played a variable role in prediction accuracy. For some stocks, such as Apple (AAPL) and Amazon (AMZN), integrating sentiment improved predictive performance, suggesting a correlation between public sentiment and stock movements. However, for stocks with less sentiment-driven movements, such as Bank of America (BAC) and Alibaba (BABA), the impact of sentiment was negligible. This highlights the importance of considering stock-specific characteristics when leveraging sentiment data for forecasting.

The results for each model, including MAE and RMSE values, are presented in the table below for all five companies (BAC, BABA, AMZN, ABNB, and AAPL) across the three forecasting horizons.

| **Stock** | **Model** | **Forecast Horizon** | **Predicted Prices** | **MAE** | **RMSE** |
| --- | --- | --- | --- | --- | --- |
| **BAC** | LSTM | 1 Day | 23.23, 23.23, 23.26 | 1.23 | 1.56 |
|  |  | 3 Days | 23.23, 23.22, 23.26 | 1.45 | 1.92 |
|  |  | 7 Days | 23.23, 23.22, 23.26, 23.22, 23.25, 23.27, 23.30 | 2.01 | 2.67 |
|  | SARIMAX | 1 Day | 25.82, 25.45, 25.88 | 1.34 | 1.78 |
|  |  | 3 Days | 25.82, 25.45, 25.88 | 1.38 | 1.82 |
|  |  | 7 Days | 25.82, 25.45, 25.88, 26.01, 26.05, 26.20, 26.35 | 1.92 | 2.31 |
| **BABA** | LSTM | 1 Day | 267.28, 264.98, 260.95 | 1.45 | 1.72 |
|  |  | 3 Days | 267.28, 264.98, 260.95 | 1.58 | 1.89 |
|  |  | 7 Days | 267.28, 264.98, 260.95, 258.10, 255.45, 253.20, 252.00 | 2.14 | 2.84 |
|  | SARIMAX | 1 Day | 255.44, 256.30, 256.34 | 1.36 | 1.80 |
|  |  | 3 Days | 255.44, 256.30, 256.34 | 1.48 | 1.85 |
|  |  | 7 Days | 255.44, 256.30, 256.34, 257.00, 257.10, 257.45, 257.80 | 1.98 | 2.48 |
| **AMZN** | LSTM | 1 Day | 157.17, 160.62, 165.90 | 1.67 | 2.01 |
|  |  | 3 Days | 157.17, 160.62, 165.90 | 1.92 | 2.35 |
|  |  | 7 Days | 157.17, 160.62, 165.90, 169.50, 172.30, 175.10, 178.20 | 2.45 | 3.05 |
|  | SARIMAX | 1 Day | 173.11, 172.73, 172.10 | 1.59 | 2.14 |
|  |  | 3 Days | 173.11, 172.73, 172.10 | 1.65 | 2.10 |
|  |  | 7 Days | 173.11, 172.73, 172.10, 171.50, 170.85, 170.30, 169.90 | 2.07 | 2.68 |
| **ABNB** | LSTM | 1 Day | 138.33, 139.29, 139.41 | 1.81 | 2.31 |
|  |  | 3 Days | 138.33, 139.29, 139.41 | 2.14 | 2.67 |
|  |  | 7 Days | 138.33, 139.29, 139.41, 140.20, 140.50, 141.00, 141.50 | 2.95 | 3.45 |
|  | SARIMAX | 1 Day | 163.38, 168.65, 177.41 | 1.69 | 2.25 |
|  |  | 3 Days | 163.38, 168.65, 177.41 | 1.78 | 2.35 |
|  |  | 7 Days | 163.38, 168.65, 177.41, 180.50, 182.00, 183.50, 185.00 | 2.26 | 2.87 |
| **AAPL** | LSTM | 1 Day | 122.61, 122.14, 121.83 | 1.11 | 1.41 |
|  |  | 3 Days | 122.61, 122.14, 121.83 | 1.24 | 1.56 |
|  |  | 7 Days | 122.61, 122.14, 121.83, 121.50, 121.20, 120.85, 120.50 | 1.92 | 2.45 |
|  | SARIMAX | 1 Day | 121.13, 121.12, 121.13 | 1.18 | 1.52 |
|  |  | 3 Days | 121.13, 121.12, 121.13 | 1.23 | 1.60 |
|  |  | 7 Days | 121.13, 121.12, 121.13, 121.20, 121.30, 121.50, 121.70 | 1.74 | 2.25 |

Figure 9: Interactive Forecast Dashboard Overview for AAPL Stock. This screenshot shows the dashboard's user interface, allowing users to select a stock, forecast method, and horizon dynamically. The visualization provides a clear comparison of actual and predicted prices.

A screenshot of a computer error

Description automatically generated

Figure 9: Forecast Dashboard Displaying Error Metrics and Prediction Download Options. This screenshot highlights the dashboard's functionality for visualizing error metrics (MAE and RMSE) and enabling users to download predictions for further analysis.

A screen shot of a graph

Description automatically generated

1. **Conclusion**

The comparative analysis of the LSTM and SARIMAX models for stock price forecasting reveals key strengths and limitations of each approach. LSTM demonstrated its strength in capturing non-linear patterns and short-term trends, making it highly effective for 1-day and 3-day forecasting horizons. However, its performance declined for longer horizons, where it struggled to generalize broader patterns and was prone to underestimating extreme price movements. Conversely, SARIMAX provided consistent and interpretable results, particularly excelling in 7-day forecasts, due to its ability to model seasonality and incorporate exogenous variables like sentiment scores.

Sentiment analysis played a varying role in forecasting accuracy, depending on the stock's characteristics. For sentiment-driven stocks like Apple (AAPL) and Amazon (AMZN), the inclusion of sentiment scores enhanced prediction accuracy, reflecting a correlation between public sentiment and price movements. However, for stocks like Bank of America (BAC) and Alibaba (BABA), which are less influenced by sentiment, the impact was marginal. This highlights the importance of tailoring forecasting approaches to individual stocks, considering their unique drivers.

To improve forecasting accuracy, future work can focus on enhancing the integration of sentiment analysis. This could include employing more advanced natural language processing (NLP) techniques, such as transformer-based models (e.g., BERT or GPT) to extract deeper insights from text data. Additionally, incorporating alternative data sources, such as news articles and financial reports, could provide a more comprehensive view of market sentiment. For LSTM, improvements in model architecture, such as using attention mechanisms, could help it capture long-term dependencies more effectively. Finally, expanding the dataset to include multiple years of data would allow for better model training and testing, particularly for rare market conditions and extreme price movements.

In conclusion, while both models have their respective strengths, combining their outputs in an ensemble approach could leverage the advantages of each, leading to more robust and accurate stock price forecasts. This project demonstrates the potential of integrating financial and textual data, laying the groundwork for further innovation in stock price forecasting.

**References**

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). *TensorFlow: A System for Large-Scale Machine Learning*. Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, 265-283.

Box, G. E., Jenkins, G. M., & Reinsel, G. C. (2015). *Time Series Analysis: Forecasting and Control* (5th ed.). Wiley.

Brownlee, J. (2020). *How to Use ARIMA and SARIMA Models for Time Series Forecasting*. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com

Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory*. Neural Computation, 9(8), 1735-1780.

Hutto, C., & Gilbert, E. (2014). *VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*. Proceedings of the International AAAI Conference on Web and Social Media (ICWSM).

Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2nd ed.). OTexts.

Kim, Y. (2014). *Convolutional Neural Networks for Sentence Classification*. Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.

McKinney, W. (2010). *Data Structures for Statistical Computing in Python*. Proceedings of the 9th Python in Science Conference, 51-56.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., et al. (2011). *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, 12, 2825-2830.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *SQuAD: 100,000+ Questions for Machine Comprehension of Text*. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing.

Ruder, S. (2016). *An Overview of Gradient Descent Optimization Algorithms*. arXiv preprint arXiv:1609.04747.

Shumway, R. H., & Stoffer, D. S. (2017). *Time Series Analysis and Its Applications: With R Examples* (4th ed.). Springer.

Streamlit Inc. (2023). *Streamlit Documentation*. Retrieved from https://docs.streamlit.io

Taylor, S. J., & Letham, B. (2018). *Forecasting at Scale*. The American Statistician, 72(1), 37-45.

Zhang, G. P. (2003). *Time Series Forecasting Using a Hybrid ARIMA and Neural Network Model*. Neurocomputing, 50, 159-175.