HUMBER COLLEGE OF TECHNOLOGY AND LEARNING
Project Report

Capstone Course for BIA 5450-0GB

Submitted by:

First Name	First Name	Student Number
Vishva	Shah	N01580093

Submitted to: Samer Al-Obaidi

Submission Date: 16th August 2024

Table of Contents

Executive Summary

Aim: The project aimed to enhance stock price prediction accuracy and provide actionable insights for better investment decisions. By analyzing historical data from the top five Fortune 500 companies in the FMCG, Tech, Health, Finance, and Utility sectors, and employing both conventional ARIMA and modern LSTM forecasting techniques, the initiative seeked to deliver reliable forecasts and strategic recommendations to help investors optimize their portfolios.

Data Collection and Analysis:

- **Data Sources:** Historical stock price data from Yahoo Finance API.
- **Key Data Entities:** Date, Open, High, Low, Close, Volume, Adjusted Close.
- **Models Applied:** LSTM (Long Short-Term Memory) and ARIMA (AutoRegressive Integrated Moving Average).

Model Development and Implementation:

- **LSTM Model:** Designed a multi-layer LSTM network with dropout layers to prevent overfitting. The architecture included input layers, LSTM layers, dropout layers, and dense layers to capture complex temporal dependencies.
    - **Training:** Utilized a batch size of 64, 100 epochs, Adam optimizer, and Mean Squared Error loss function.
    - **Evaluation:** Applied Root Mean Squared Error (RMSE) for performance assessment.

Outcomes and Findings:

- **Predictive Accuracy:** The LSTM model demonstrated superior accuracy in capturing non-linear trends, with lower RMSE and MAE compared to ARIMA.
- **Computational Cost:** LSTM required more computational resources and training time but offered higher predictive power. ARIMA was less resource-intensive but struggled with scalability and complex data patterns.

- **Model Performance:** LSTM excelled in scenarios with complex, volatile data, while ARIMA performed well with linear and simpler patterns

2. Introduction

In this report, we will explore the implementation process of our stock price prediction project, focusing on the two primary predictive models used—Long Short-Term Memory (LSTM) and AutoRegressive Integrated Moving Average (ARIMA). Section 1 will provide an overview of the approaches, algorithms, and models that were tested. Section 2 will reflect on the implementation process, highlighting key insights and challenges encountered. Finally, Section 3 will outline the steps that client company personnel will need to follow to effectively utilize the developed solution.

3. Business Problem Overview

3.1 Goal

Investors are always looking for dependable ways to optimise returns while lowering risks in today's hectic and extremely turbulent financial markets. Because financial markets are so dynamic and complex, traditional investment strategies—which frequently rely on historical data and basic analysis—may not be able to correctly predict market changes. An additional difficulty for investors is the overwhelming amount of available data, which includes stock prices, economic indicators, and market sentiment. This makes it challenging for investors to manually process and extract meaningful insights.

3.2 Scope:

Utilising machine learning techniques to examine the performance of the top 5 Fortune 500 firms in the FMCG, tech, health, and finance sectors, this research seeks to address this challenge. These industries were selected because they reflect a variety of market dynamics and have a substantial influence on the world economy. The companies that are chosen in these sectors are leaders in their respective industries, and their stock performance frequently shapes larger market patterns.

3.3 Problem Statement:

To comprehend the elements influencing stock performance in these important industries, investors want a thorough, data-driven approach. The objective is to create a predictive model that, in addition to offering insights into how different market movements and economic variables affect these equities, can reliably estimate stock values based on previous data. The project hopes to accomplish this by giving investors the resources they need to optimise their portfolios, make wise decisions, and eventually increase their investment returns.

3.4 Obstacles:

The following are the main obstacles to accomplishing this goal:

Data Complexity: A wide range of elements, such as economic indicators, world events, and company-specific news, affect stock prices. It is quite difficult to include these intricate interactions into a prediction model.

Market Volatility: Due to the natural erratic nature of stock markets, even slight alterations in outside circumstances have the potential to cause significant fluctuations in stock values. Even in the face of these modifications, a strong model needs to be able to make accurate predictions.

3.5 Model Choice and Precision

It's critical to select Machine Learning models that strike a balance between interpretability and forecast accuracy. In addition to producing precise projections, the models need to give investors actionable insights. Anticipated Result: By effectively tackling these obstacles, the project hopes to produce a collection of useful insights and investment suggestions that are based on in-depth data research. The results will comprise:

Predictive Accuracy: The ability of a model to accurately predict the closing values of stocks in a given sector, allowing investors to predict changes in the market.

Sector-Specific Insights: a thorough examination of the ways in which various market trends and economic indicators affect a given sector, giving investors a better grasp of the factors influencing stock performance.

Strategic recommendations are investment plans that are suited to the particulars of each industry and assist investors in optimising their portfolios for improved returns and risk management.

4. **Analytical Questions**

- In what ways can historical financial metrics be used to enhance machine learning predictions of future stock performance, and which models excel in accuracy and precision?
- What is the average trading volume for each sector, and how does it vary?
- How can data-driven recommendations help new investors manage risks and maximize profits by pointing them to high-potential industries and businesses?

5. **Scope Statement**

To provide investors with actionable insights and recommendations by analyzing the performance of top Fortune 500 companies in the Pharma, Tech, Finance, Utility, and FMCG sectors using advanced data analytics. The focus is on the predictive accuracy of closing stock prices, employing both modern and traditional forecasting methods.

Data Collection: We analyzed the top 5 companies in the Health, Tech, Finance, and FMCG sectors from the Fortune 500 list. Historical stock performance data for these companies were gathered.

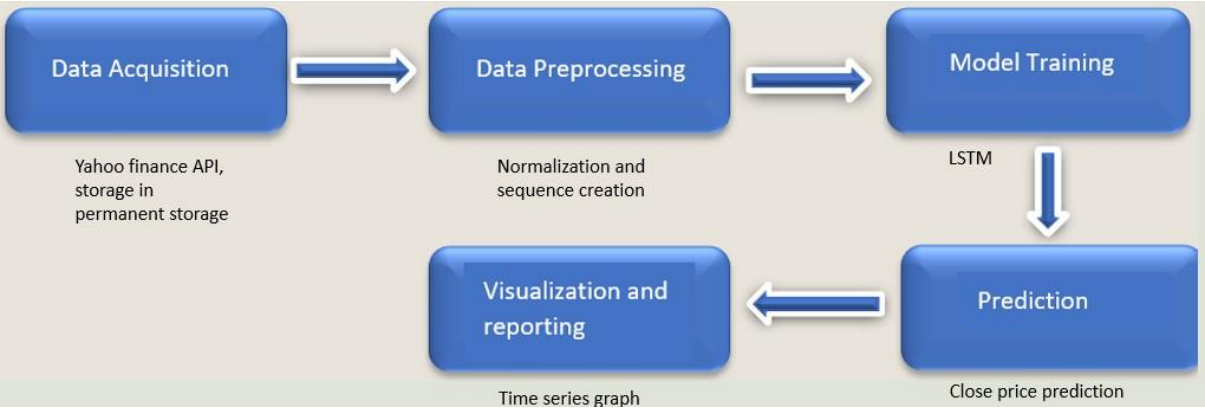
Descriptive Analysis: The analysis focused on trends and patterns in the closing prices of stocks, offering a detailed examination of market behavior.

Model Comparison: The project utilized an LSTM model to predict stock closing prices and compared its performance against traditional methods such as ARIMA and Moving Average. The comparison aimed to assess the predictive accuracy and reliability of these models.

Recommendations: Based on the analysis, a comprehensive investment strategy report was developed, offering actionable insights on managing risks and optimizing returns. The report includes guidance on the most effective models for predicting stock performance, supporting informed investment decisions.

6. Key Data entities and flow diagram

- Data Sources: Yahoo Finance API (Provides historical stock price data)
- Attributes: Date, Open, High, Low, Close, Volume, Adjusted Close
- Key Data Entity: Stock Price (Predict future stock prices and analyze historical performance trends.)



Data Flow diagram

7. Data Manipulation Process and data output

7.1. Data acquisition

The stock price data was sourced from Yahoo Finance via their API. This source was chosen for its dependability, extensive historical data, and simplicity of connection with Python modules.

To improve data processing and time series analysis, the data frame's index was set to the 'Date' column. Indexing by date makes data management and visualisation more intuitive, while also preserving the temporal character of stock prices throughout analysis.

7.2. Data Preprocessing Visualization: Historical closing prices were plotted to analyse stock price trends over time. This initial visualisation revealed the overall movement, volatility, and probable seasonality or cyclic patterns in the data. It acted as a

foundation for further investigation by emphasising crucial themes that could influence the predictive models.

- **Data Filtering:** We built a new data frame with only the 'Close' column, which shows the daily closing prices of stocks. This data was then turned into a NumPy array to speed up model training. Focussing on 'Close' prices enabled a more specific study, as closing prices frequently reflect the most relevant price level of the day in financial research
- **Normalization:** To improve model performance, stock prices were scaled to a range of 0-1. Normalisation is an important step in preprocessing, particularly for machine learning models like LSTM, because it guarantees that the data is on a uniform scale, eliminating any feature from disproportionately influencing the model. Scaling the pricing allows the model to better understand the underlying patterns without being influenced by the magnitude of the values.
- **Sequence Creation:** To anticipate the closing price on the 61st day, we created a sequence of 60 consecutive days of closing prices. This stage is critical for collecting temporal relationships in the data, which are required for the LSTM model to perform properly. The 60-day period was chosen to strike a compromise between obtaining enough historical data for context while retaining computing efficiency. These sequences served as the foundation for training the LSTM model, which was then able to effectively learn patterns in time series data.

7.3. Data Splitting: The dataset was divided into training and testing sets in an 80:20 ratio The training set included 80% of the data required to train the model, with the remaining 20% reserved for testing. Importantly, the data's sequential character was preserved during the split to retain time series continuity, ensuring that the model was trained on past data and validated on future data. This strategy is similar to real-world scenarios in which future prices are projected based on historical trends.

7.4. Predictive and Evaluation Model Training:

- **Loss Function:** The Mean Squared Error (MSE) was used as the loss function. MSE calculates the average of the squared errors, or the difference between anticipated and actual values. It was chosen because it is sensitive to large errors and penalises significant deviations, which is necessary for accurate stock price prediction and to prevent huge financial losses
- **Optimizer:** We used the Adam optimiser to improve model convergence. Adam is an adaptive learning rate optimisation algorithm that combines the benefits of two other modifications of stochastic gradient descent. It automatically modifies the learning rate, making it ideal for handling complex patterns in stock price data.
- **Evaluation metric:** The Root Mean Squared Error (RMSE) was used to assess model performance on test data. The root mean squared error (RMSE) is a popular regression statistic that estimates the standard deviation of prediction errors. It estimates the average magnitude of the error, and a lower RMSE indicates better model performance. Focussing on RMSE, the accuracy of the LSTM model's predictions was quantitatively tested, guaranteeing that the model's outputs were as close as possible to the actual closing prices.

8.1 New solution design and it's fit into the existing IT architecture

LSTM Model: An LSTM (Long Short-Term Memory) model is a type of recurrent neural network (RNN) specially designed to avoid the long-term dependency problem that traditional RNNs face. This capability makes LSTMs particularly suited for tasks involving sequences, such as time-series prediction, natural language processing, and speech recognition.

2.1 Key Features of LSTM Models:

1. **Memory Cells:** At the core of every LSTM unit is a memory cell that can maintain its state over time, making it ideal for learning from sequences of data where the timing and order of events are crucial.

2. **Gates:** LSTMs incorporate three types of gates that regulate the flow of information: • **Forget Gate:** Determines parts of the cell state to discard from previous timesteps.

- **Input Gate:** Updates the cell state by adding new information to the memory.
- **Output Gate:** Decides what information to output based on the current input and the memory of the cell.

3. Why LSTM?

- **Handling Sequential Data:** Long-term dependencies are remembered by LSTM networks, which makes them useful for gradually learning patterns. In stock price prediction, this is important because the present value is dependent on previous values.
- **Mitigating Vanishing Gradient Problem:** Unlike typical neural networks, LSTM networks feature memory cells and gates that allow them to maintain and learn from extended input sequences. The learning of long-term dependencies during backpropagation is improved by this architecture's assistance in preserving gradients and significant information.

- **Capturing Temporal Dependencies:** Over time, several factors affect stock values. LSTMs are perfect for predicting future values based on past data because they can capture the temporal correlations and trends in the data.
- **Flexibility and Power:** Due to the ability of LSTMs to describe intricate correlations and nonlinearities in data, stock prices—which frequently display non-linear behavior as a result of market dynamics—can be predicted with great accuracy.

8.2 The fit of the new solution into the existing IT architecture

To guarantee smooth operation, data flow, and compatibility, there are several factors to take into account while integrating the LSTM-based time series forecasting solution into the current IT architecture. Data Integration

- ***Data Sources:*** The system is based on past stock price information, which is usually retrieved via APIs or kept in databases. It is highly suited to systems that incorporate proven data pipelines for data extraction, transformation, and loading (ETL) procedures.
- ***Data Storage:*** Large datasets are frequently needed for LSTM model training. To manage the growing volume of data, the current infrastructure should be able to accommodate scalable storage options like data lakes or cloud storage.
- ***Training and Inference:*** Due to their computational demands, LSTM models need to be trained effectively with strong GPUs or cloud computing resources. Best suited for integration are systems like cloud platforms with GPU instances, which can dynamically assign resources based on workload demands. Integration with Business Processes
- ***Workflow Integration:*** The solution needs to work with the current business procedures for decision-making and forecasting. This entails coordinating with the dashboards, reporting systems, and data analysis tools that stakeholders utilize.
- ***User Access and Security:*** To guarantee secure access to the forecasting model and its outputs, appropriate authentication and access restrictions that fit within the organization's current security standards must be put in place. Interoperability and

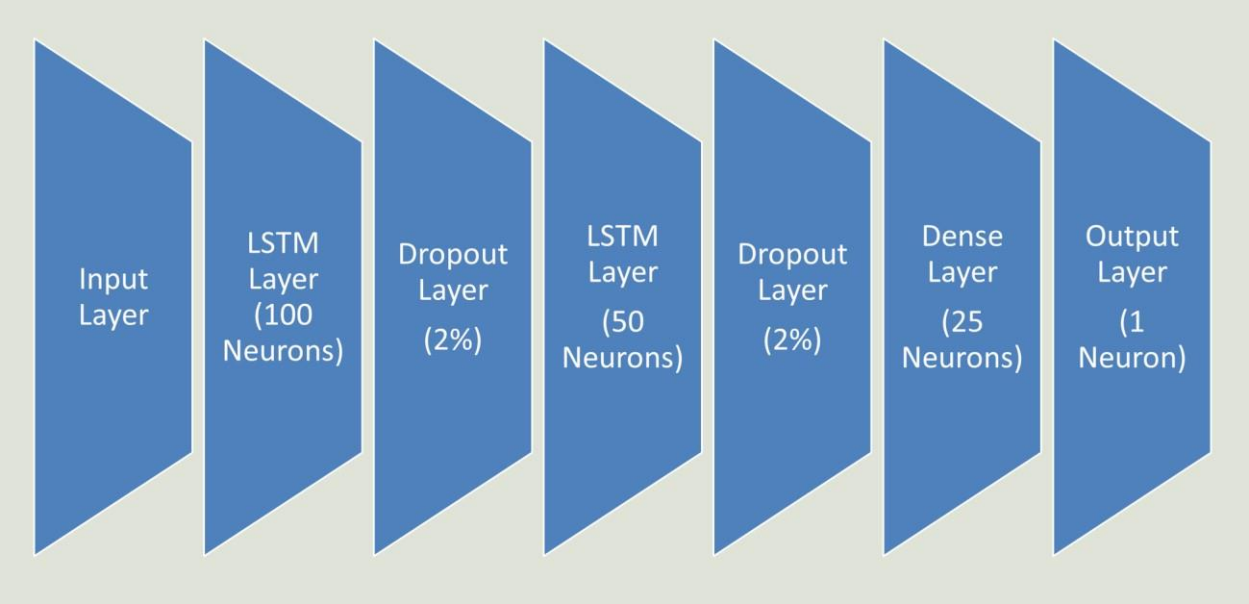
Compatibility

- APIs and Automation: Integrating existing APIs and automation scripts can streamline data feeding and prediction retrieval, ensuring the solution fits into automated workflows and reduces manual intervention.

9.LSTM Model Implementation and outcome testing

The LSTM model was designed to capture the complex temporal dependencies inherent in stock price movements. The implementation involved several key steps:

Model Architecture: Designed a multi-layer LSTM network with dropout layers to prevent overfitting.



LSTM Architecture:

Input Layer: It takes only one parameter as an input. Timesteps, and no features in the dataset.

1st LSTM Layer: No of neurons and return sequence as True. We have taken 100 neurons to catch the non-linear complex relation among different features. Return sequence is True because we are using another LSTM layer to capture the relationship.

Dropout Layer: We have used a dropout layer to prevent the model from overfitting.

2nd LSTM Layer: No of neurons and return sequence as False. We have taken 50 neurons to catch the non-linear complex relation among different features. Return sequence is False because we are not using another LSTM layer after this one.

Dropout Layer: We have used another dropout layer for the same reason to prevent the model from overfitting.

Dense Layer: We have used the dense layer to further capture the relationship using 25 neurons.

Output Layer: We have used a dense layer with one neuron as output layer, because we want to predict the close price at the end, and we need only one node to give us the output.

Layer (Type)	Output Shape	Param #
lstm (LSTM)	(None,60,100)	40,800
dropout (Dropout)	(None,60,100)	0
lstm_1 (LSTM)	(None, 50)	30,200
dropout_1 (Dropout)	(None, 50)	0
dense (Dense)	(None, 25)	1,275
Dense_1 (Dense)	(None, 1)	25

Total params: 216,905 (847.29 KB)

Trainable params: 72,301 (282.43 KB)

Non-trainable params: 0 (0.00 B)

Optimizer params: 144,604 (564.86 KB)

Training: Employed a batch size of 64 and 100 epochs, utilizing Adam as the optimizer for efficient backward propagation of errors. We have used mean_squared_error as a loss function for backward propagation. Also, for the evaluation we have used mean_squared_error as accuracy metric. It was chosen because we are trying to predict a continuous variable. This makes it perfect for model performance.

Validation: Set aside 20% of the data as a validation set to monitor the model's performance and adjust parameters dynamically.

The choice of hyperparameters like batch size and number of epochs was critical. Initial tests with smaller batch sizes led to slower convergence rates, prompting an adjustment to optimize training time without compromising the accuracy.

4. ARIMA Model Development

The ARIMA model, which stands for AutoRegressive Integrated Moving Average, is a popular statistical approach used for forecasting time series data. It is particularly wellsuited for short-term predictions and can handle data with trends and seasonal patterns effectively.

4.1 Key Components of ARIMA Models:

1. **AR (AutoRegressive):** This part of the model captures the relationship between an observation and a number of lagged observations (i.e., past values). The "p" parameter in ARIMA denotes the number of lag observations included in the model.

2. **I (Integrated):** This component involves differentiating the time series data one or more times to make the data stationary, meaning that the statistical properties such as mean, and variance of the series do not change over time. The "d" parameter represents the degree of differencing required.

3. **MA (Moving Average):** This aspect models the relationship between an observation and a residual error from a moving average model applied to lagged observations. The "q" parameter specifies the size of the moving average window.

Limitations: There are many limitations when it comes to ARIMA models, which are:

Stationarity Requirement: ARIMA requires the underlying data to be stationary. If the data show trends, seasonal patterns, or other forms of nonstationarity, they must first be transformed, which can sometimes lead to complexities in model interpretation.

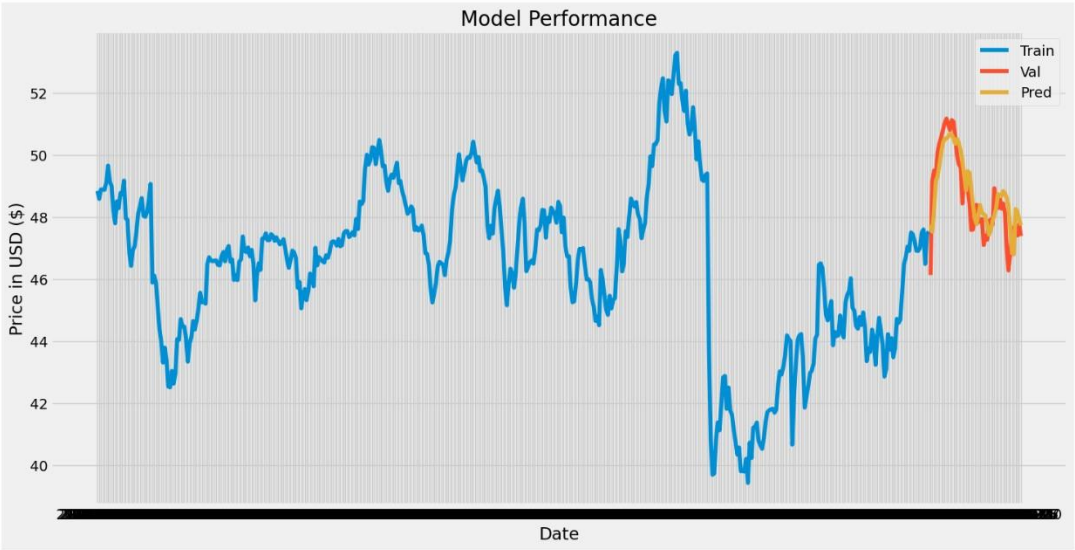
- **Noisy Data Sensitivity:** ARIMA models can be sensitive to noise and outliers in the data, which can affect the model's accuracy.

- ARIMA stood as a contrast to LSTM, providing a statistical approach to forecasting. The development process included:
- **Parameter Selection:** Utilized statistical tests to determine the (p, d, q) parameters, ensuring the model was well-tuned to the data's inherent characteristics.
- **Model Fit:** Integrated seasonal adjustments to cater to underlying patterns not immediately apparent in the data.
- **Forecasting:** Extended predictions beyond the training data to evaluate the model's extrapolative capabilities. The ARIMA model was less flexible in managing the nonlinear patterns observed in the stock data, which was anticipated.

Outcome Testing and Reviewing

This section of the report addresses the critical phase of outcome testing and reviewing in the solution implementation process. Here, we detail the tests conducted to evaluate the performance of our LSTM and ARIMA models in predicting stock prices, analyzing the computational costs and accuracy of each model. The outcomes of these models are compared against each other as well as against predefined benchmarks to assess their efficacy and efficiency.

For example, we have predicted Walmart stock price using LSTM and we got accuracy of 96.4% and RMSE 3.6%.



Testing Methodology: To get the best working model for stock price prediction. Below mentioned benchmarks were used.

Performance Metrics:

To ensure a comprehensive evaluation, the following metrics were employed:

Accuracy Metrics: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) were used to measure the predictive accuracy of each model.

Computational Cost: Time taken for training and prediction phases were recorded to evaluate computational efficiency. It is hard to quantify the number, but it was clear from the training of the model that LSTM requires significant GPU power as it uses deeply connected Neural Network to capture the complex non-linear relationship. As the data increase and number of features increase the computational cost will go through roof and might require powerful servers.

Test Setup:

The data set was split into training and testing segments, with the latter being used exclusively for performance evaluation to avoid data leakage. Both models were tested under identical data conditions to ensure a fair comparison.

Model Performance Analysis

LSTM Model:

Accuracy: The LSTM model achieved a lower RMSE and MAE compared to ARIMA, indicating a higher predictive accuracy, especially in capturing the non-linear trends of the stock price movements.

Computational Cost: Due to its complex architecture involving multiple layers and feedback loops, the LSTM model required significantly more training time than ARIMA. However, once trained, the model's prediction time was comparably efficient.

ARIMA Model:

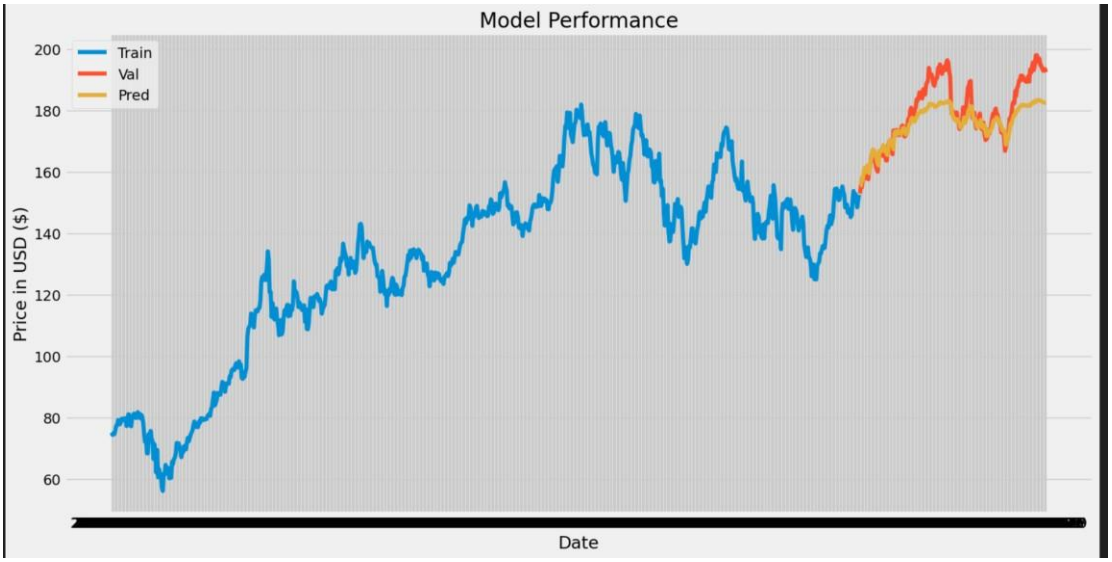
Accuracy: ARIMA, while not as accurate in handling non-linear data fluctuations as LSTM, performed commendably with linear trends and seasonal patterns. It was not effective even in shorter forecast horizons.

Computational Cost: ARIMA was less computationally intensive during the training phase but faced challenges in scalability when data volume increased.

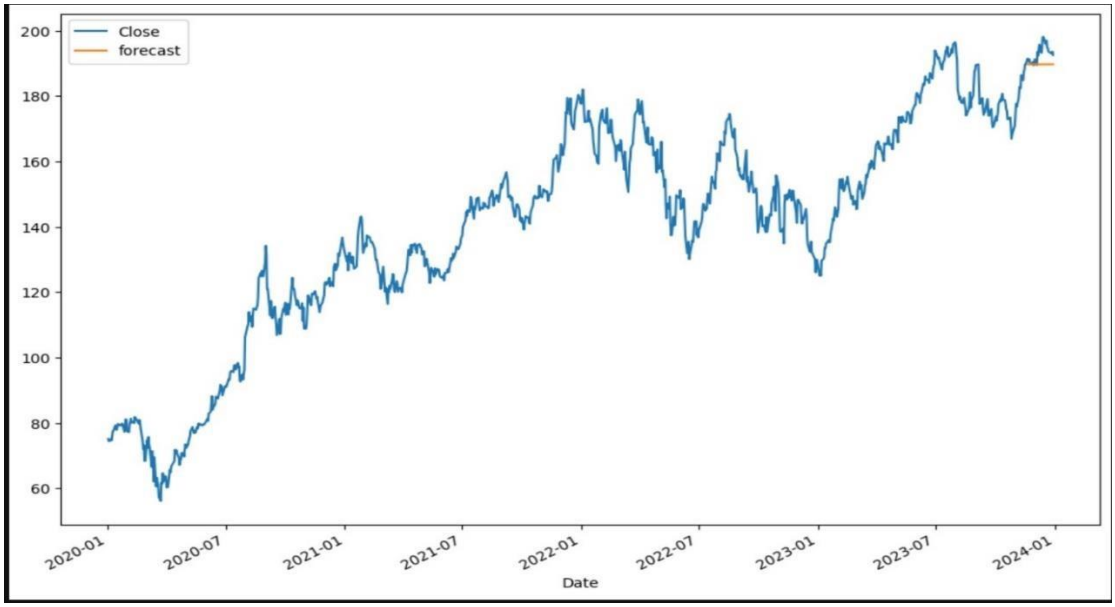
Comparative Analysis

Accuracy Comparison:

A side-by-side comparison showed that LSTM outperformed ARIMA in scenarios involving complex, volatile stock price movements, while ARIMA was preferable for simpler, predictable patterns.



(LSTM Model predicting the stock price in yellow line.)



(ARIMA Model predicting the stock price in yellow line.)

Computational Efficiency:

- While LSTM required more initial setup and training time, its efficiency in batch processing large datasets was evident. ARIMA's lower computational load made it suitable for applications with limited processing power but at the cost of predictive power in more complex scenarios.

Outcome Against Desired Results

- The testing phase involved comparing the observed model outputs against desired outcomes, which were defined based on historical performance benchmarks in stock price prediction.
- Both models met the desired thresholds for certain metrics; however, trade-offs between accuracy and computational efficiency highlighted the need for contextdriven model selection.

10. Potential solution optimization

10.1 Suggestions for Users Looking to Incorporate Predictive Analytics into Investment Plans:

1. **Data-Driven Decisions:** Focus investment strategies on relevant trends and projections by utilising the insights from the predictive models. Based on the examination of past data and recognised trends, this method helps users to make better financial selections.

Utilise Predictive Models to Manage Risk:

2. **Proactive Risk Assessment:** Evaluate possible hazards and take steps to mitigate them by using the models' predictive power. Users can minimise exposure to volatile or decreasing assets by adjusting their portfolios based on their understanding of future market movements.
3. **Expand Data Sources:** To enhance the study and boost prediction accuracy, use supplementary data sources like market sentiment, economic indicators, or alternative data (such news sentiment, social media trends, etc.).
4. **Regular Updates and Retraining:** To adjust the models to evolving market conditions, update and retrain them with fresh data on a regular basis. This guarantees that the forecasts will continue to be accurate and relevant throughout time.

Instruction & Training for Users:

5. **Model Understanding:** Teach people how predictive models operate and how to properly understand their results. By offering tools or training, users can make better use of the models while making investment decisions.

10.2 Potential Optimizations for the Solution

1. **Flexible Model Deployment:** Create an adaptable system that enables the addition of fresh data sources and modelling strategies as they become available. By doing this, the prediction system is kept abreast of the most recent developments in data analysis and machine learning.

2. **Real-Time Data Processing:** Create tools that can process and anticipate data in real-time. Because of this, customers may get the most recent forecasts and decide whether to invest based on the state of the market.

3. **Cloud-Based Solutions:** To guarantee scalability and effectively manage massive volumes of data, think about putting the models on cloud platforms. The ability to scale resources according to demand and computing requirements is made possible by cloud technologies.

4. **Construct an Intuitive Dashboard:** Design a dashboard or user interface that is easy to use and displays trends, actionable insights, and prediction outcomes. This improves accessibility and facilitates more efficient user interaction with the forecasts.

5. **Set Up Alerts:** Put in place automated alert systems to let users know when there are big price movements or anticipated trends in stocks. Alerts can assist users in responding quickly to emerging opportunities or risks

11. References

https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-seriesforecasting/

- **LSTM:** https://machinelearningmastery.com/gentle-introduction-long-short-termmemorynetworks-experts/

  https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM
- **ARIMA:** https://machinelearningmastery.com/arima-for-time-series-forecasting-withpython/
- **Adam Optimizer:** https://machinelearningmastery.com/adam-optimization-algorithmfor-deep-learning/
- **Yahoo Finance:** https://pypi.org/project/yfinance/
- **Keras:** https://www.tensorflow.org/guide/keras