Stock Market Analysis and Prediction Using LSTM: A Case Study on Apple Stock

Monsoon 2024

#### Submitted by

### Adeep Sharma (2210110116)

#### Under Supervision of

### Dr. Saurabh Shigwan Department of Computer Science Engineering



**Department of Computer Science Engineering, School of Engineering,**

**Shiv Nadar Institution of Eminence Deemed to be University, Delhi-NCR**

Abstract

The stock market, characterized by its complexity and volatility, presents an enduring challenge for accurate prediction. Recent advancements in deep learning, especially the emergence of Long Short-Term Memory (LSTM) networks, have enabled researchers to model intricate temporal dependencies within financial data. This report explores the use of LSTM models for predicting the adjusted closing prices of Apple Inc. (AAPL) stocks, leveraging historical data and diverse feature sets to enhance predictive accuracy.

Two distinct approaches were adopted for this study. **Approach 1** integrates a com- prehensive range of features, including the adjusted closing prices of major technology companies (e.g., META, AMZN, GOOGL, MSFT), macroeconomic indicators such as GDP and CPI, and sentiment analysis from social media. This multi-dimensional per- spective aims to capture the diverse factors influencing Apple’s stock prices. **Approach 2,** by contrast, employs a minimalistic strategy, relying solely on lagged historical prices of Apple’s stock to predict future values. Despite its simplicity, Approach 2 demonstrates superior evaluation metrics.

The findings reveal that while Approach 2 achieves marginally higher accuracy with an R-squared value of 0.9812, Approach 1’s holistic feature integration offers broader insights into the underlying factors driving stock price dynamics. This dual evaluation underscores the potential of LSTM networks in balancing simplicity and complexity for stock market prediction.

**Contents**

1. Introduction 7
2. Literature Review 9
3. Data Collection and Preprocessing 11
   1. [Stock Data 11](#_TOC_250020)
   2. [Macroeconomic Data 11](#_TOC_250019)
   3. [Sentiment Data 11](#_TOC_250018)
   4. [Dataset Overview 12](#_TOC_250017)
   5. [Handling Missing Data 13](#_TOC_250016)
   6. [Feature Scaling 13](#_TOC_250015)
   7. [Feature Engineering: Lag Features 14](#_TOC_250014)
      1. [Lag Features 14](#_TOC_250013)
      2. [Train-test-validation split 14](#_TOC_250012)
4. Methodology 15
   1. [LSTM Model Architecture 15](#_TOC_250011)
   2. [Model Training and Evaluation 16](#_TOC_250010)
5. Evaluation Metrics 17
   1. [Mean Absolute Error (MAE) 17](#_TOC_250009)
   2. [Mean Squared Error (MSE) 17](#_TOC_250008)
   3. [R-squared (*R*2) 18](#_TOC_250007)
   4. [Mean Absolute Percentage Error (MAPE) 18](#_TOC_250006)
   5. [Conclusion 19](#_TOC_250005)
6. Result and Discussion 20
   1. [Model Performance 20](#_TOC_250004)
      1. [Comparison of Evaluation Metrics 20](#_TOC_250003)
   2. [Graphical Analysis 21](#_TOC_250002)
      1. [Approach 1: Graphical Performance Analysis 21](#_TOC_250001)
      2. [Approach 2: Graphical Performance Analysis 23](#_TOC_250000)
7. Conclusion 25
8. Future Work 26

**List of Figures**

* 1. Sample of the dataset showing features and target variables. 12
  2. Actual vs Predicted Apple Stock Prices (Approach 1) 21
  3. Difference between GT and training data predicted stock price (Approach 1) 22
  4. Difference between GT and testing data predicted stock price (Approach 1) 22
  5. MAE Loss vs Epochs (Approach 1) 23
  6. Actual vs Predicted Apple Stock Prices (Approach 2) 23

**List of Tables**

* 1. Comparison of Evaluation Metrics for Both Approaches 21

**Chapter 1 Introduction**

The stock market is a dynamic and intricate system influenced by a wide range of fac- tors, from company performance and economic indicators to market sentiment and global events. Predicting stock prices is a challenging task that has drawn the attention of researchers, financial analysts, and investors alike. The advent of machine learning, par- ticularly deep learning methods, has provided powerful tools for extracting patterns and making predictions based on historical data. Among these methods, Long Short-Term Memory (LSTM) networks, a specialized form of recurrent neural networks, have proven highly effective for time series forecasting due to their ability to capture temporal depen- dencies and long-term patterns.

This report explores the application of LSTM networks to predict the adjusted closing prices of Apple Inc. (AAPL) stocks. Apple, being one of the most valuable and widely traded companies in the world, provides a rich dataset for analyzing stock price move- ments. The goal of this project is to develop predictive models that leverage historical stock data and additional economic and sentiment-based features to forecast future stock prices. By employing two distinct approaches, we aim to evaluate the effectiveness of different feature sets and methodologies in improving prediction accuracy.

**Approach 1** involves the incorporation of a comprehensive set of features that in- fluence stock prices. These include the adjusted closing prices of major tech companies such as Meta (META), Amazon (AMZN), Alphabet (GOOGL), and Microsoft (MSFT), as well as macroeconomic indicators such as GDP, CPI, unemployment rate, and inter- est rates. Sentiment analysis features derived from Twitter polarity and volume are also included. This multifaceted approach attempts to capture the broader context affecting Apple’s stock prices, making it a robust framework for understanding the interplay of various factors.

**Approach 2,** on the other hand, adopts a more minimalistic design by predicting Apple’s stock prices based solely on its lagged adjusted closing prices. This approach leverages the inherent temporal patterns within the stock’s historical data, aiming for simplicity and computational efficiency. While this method yields impressive evaluation metrics, it lacks the broader contextual understanding provided by Approach 1.

The evaluation metrics for both approaches highlight their strengths. Approach 1 achieves an R-squared value of 0.9690, indicating a strong correlation between the pre- dicted and actual stock prices. It incorporates multiple influencing factors, providing valuable insights into the complex dynamics of stock price movements. Approach 2, with

an R-squared value of 0.9812, demonstrates higher predictive accuracy, but its scope is limited to past price patterns. This report emphasizes the importance of considering both the breadth of features and the simplicity of the model in stock price prediction.

By comparing these two approaches, we aim to balance the trade-offs between com- plexity and predictive performance. Approach 1, despite slightly lower evaluation metrics, underscores the value of incorporating diverse features to capture the multifactorial na- ture of stock price movements. Meanwhile, Approach 2 highlights the potential of simpler models in achieving high accuracy with fewer variables. This dual focus underscores the versatility of LSTM networks in stock market analysis and prediction, providing valuable insights for both researchers and practitioners in the field.

**Chapter 2 Literature Review**

Stock market prediction has been an area of intense research, with various models pro- posed to forecast stock price movements. Traditional statistical models, such as autore- gressive integrated moving average (ARIMA) and linear regression, have been widely used but often fail to capture the non-linear and complex patterns inherent in stock price move- ments. More recently, machine learning approaches, especially deep learning models, have gained popularity due to their ability to model such non-linearity and handle large, high- dimensional datasets [1]. Among these, Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), have been particularly successful in time series forecasting due to their capacity to retain information over long sequences, making them highly suitable for stock market prediction tasks [2].

LSTM networks have shown significant promise in predicting stock prices by leverag- ing historical stock data and other time series information. In particular, LSTM-based models are able to capture temporal dependencies and trends, which are crucial for pre- dicting future price movements. Studies such as by Fischer and Krauss [3] demonstrate that LSTM models can outperform traditional models in stock price prediction, showing more accurate forecasts when trained on time-series data from major stock indices. Addi- tionally, LSTM models have been applied successfully to specific stocks, including Apple, due to the high volume and richness of data available for companies of this size [4].

Moreover, combining sentiment analysis with LSTM models has proven to enhance the predictive power of stock price forecasts. The sentiment of news articles, financial reports, and social media data (such as Twitter sentiment) can offer valuable insights into public perception, which often correlates with stock market behavior [5]. Research by Xie et al. [6] highlights the benefits of integrating sentiment analysis with LSTM mod- els, demonstrating improved stock market prediction accuracy. For instance, sentiment analysis of Twitter data has been found to be highly indicative of short-term stock price movements, making it an essential feature in models for predicting stocks like Apple [7]. In the context of Apple Inc., stock price prediction using LSTM has been extensively studied, with researchers exploring both univariate and multivariate models. For instance, research by Shen et al. [8] used LSTM to predict Apple stock prices by integrating both historical stock data and sentiment scores derived from Twitter. Their results showed a notable improvement in prediction accuracy when sentiment data was included. Simi- larly, other studies have explored hybrid models that combine LSTM with other machine learning techniques such as Convolutional Neural Networks (CNN) for feature extraction,

further improving performance [9]. These studies underline the growing trend of using hybrid models to enhance predictive accuracy in stock market analysis.

While these studies show the potential of LSTM in stock price prediction, challenges remain in terms of model interpretability and the integration of diverse data types. The complexity of LSTM models, while effective in prediction, often leads to a ”black box” issue, making it difficult to understand the exact factors influencing the prediction [10]. Despite this, the success of LSTM models in stock price prediction, particularly for com- panies like Apple, offers a promising approach for financial analysts and traders seeking to leverage machine learning to predict market trends.

**Chapter 3**

**Data Collection and Preprocessing**

The data utilized in this project comprises a diverse set of features, including stock prices, macroeconomic indicators, and sentiment data, providing a comprehensive foundation for predicting Apple Inc.’s (AAPL) adjusted close prices.

# Stock Data

Historical adjusted close prices for Apple (AAPL) and major peer companies (Meta, Amazon, Google, and Microsoft) were sourced from Yahoo Finance. This dataset spans 1,338 entries from January 1, 2016, to August 30, 2019. The inclusion of stock data for peer companies highlights the significance of industry trends in predicting AAPL’s performance. For instance, shifts in Microsoft (MSFT) or Google (GOOGL) prices often correlate with macroeconomic or sector-specific events, enriching the predictive power of the model.

Incorporating these peer companies as features provides insights into competitive per- formance and market dynamics. Additionally, news data could enhance predictions by capturing the sentiment and narratives around tech-sector developments, mergers, regu- latory changes, and global events, which may directly or indirectly impact AAPL’s stock performance.

# Macroeconomic Data

Economic indicators such as GDP, CPI, unemployment rate, interest rate, industrial pro- duction, and retail sales were collected using the Federal Reserve Economic Data (FRED) API. These indicators offer a broader perspective on the economy, contextualizing fluctu- ations in stock values. For example, declining GDP or rising unemployment rates could signify economic downturns, influencing investor behavior and stock price trends.

# Sentiment Data

Sentiment analysis data from Twitter, sourced from a Kaggle dataset, adds another di- mension by capturing public opinion about Apple Inc. This dataset measures tweet

polarity and volume, reflecting how positive or negative public sentiment correlates with stock price movements. Social media activity provides real-time reactions to product launches, earnings reports, or major announcements, serving as a critical component for understanding short-term price volatility.

# Dataset Overview

The dataset includes the following features:

* **Stock prices:** Adjusted close prices for META, AMZN, GOOGL, MSFT, and AAPL.
* **Macroeconomic indicators:** GDP, CPI, unemployment rate, interest rate, indus- trial production, retail sales, and personal saving rate.
* **Sentiment data:** Twitter polarity and tweet volume.

The target variable is **AAPL** **Adj** **Close**, and the dataset spans **1,338 entries**, cov- ering stock market activity from **January 1, 2016**, to **August 30, 2019**. By combining financial, economic, and sentiment data, the model captures a multidimensional view of factors affecting AAPL stock prices, ensuring robust predictions.

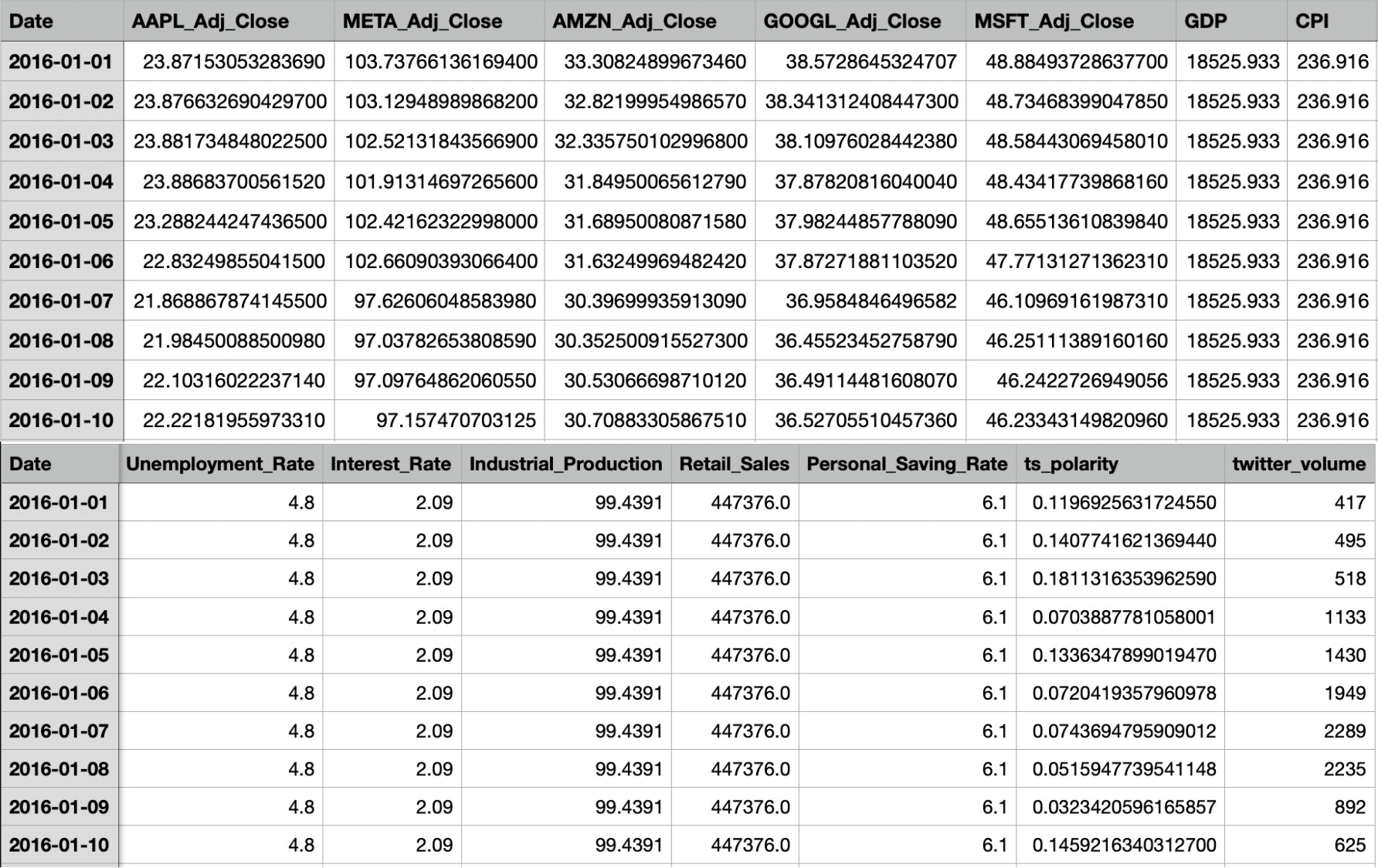


Figure 3.1: Sample of the dataset showing features and target variables.

# Handling Missing Data

Missing data is a common problem when working with real-world datasets. Missing data can introduce bias and reduce the model’s accuracy. To address this issue, several strategies can be employed:

* **Forward Fill:** For time-series data, forward filling replaces missing values with the last observed value.
* **Interpolation:** Linear interpolation can be used to estimate missing values based on surrounding data points.

Mathematically, for a time series *x*(*t*), the forward fill can be expressed as:

(

*x*(*t*) = *x*(*t*) if *x*(*t*) is not missing

*x*(*t* − 1) if *x*(*t*) is missing

In this project:

* **Stock Data:** Adjusted close prices for AAPL, META, AMZN, GOOGL, and MSFT were missing on weekends and holidays. To handle these gaps, Goel’s technique was employed, as stock prices generally follow concave curves. This technique estimates missing values by fitting a concave curve to the available data, ensuring smooth and realistic interpolation for missing days.
* **Macroeconomic Indicators:** Columns such as GDP, CPI, unemployment rate, interest rate, industrial production, and retail sales are updated at less frequent intervals (e.g., monthly or quarterly). Forward fill was applied to propagate the last available value, maintaining temporal consistency.

This approach helps maintain the integrity of the dataset by ensuring no missing values remain while preserving the inherent patterns of the data. Consequently, the LSTM model receives complete and coherent input data for training.

# Feature Scaling

Feature scaling is an essential step in preparing the data for machine learning models. Feature scaling is crucial for ensuring that all input features contribute equally to the model’s performance. In this project, all numerical features, including stock prices and macroeconomic indicators, were scaled using the Min-Max Scaling technique. The Min- Max scaling method scales the features to a specific range, typically between 0 and 1, ensuring that all features contribute equally to the model training process.

*x′* = *x* − min(*x*) max(*x*) − min(*x*)

Where:

* *x* is the original value,
* *x′* is the scaled value,
* min(*x*) and max(*x*) are the minimum and maximum values of the feature, respec- tively.

# Feature Engineering: Lag Features

Feature engineering involves creating new features that capture underlying patterns in the data. The techniques employed in this project is creating lag features:

## Lag Features

To capture temporal dependencies, lag features were created by shifting the stock price values for previous days. This technique allows the model to use past stock prices to predict future values. The lag features used in this project were created for a window of 10 days, which means that the stock prices for the past 10 days were used as input to predict the stock price for the next day.

*AAPLt−*1*, AAPLt−*2*, . . . , AAPLt−*10

where *AAPLt−i* represents the adjusted closing price of Apple stock at time *t* − *i*.

## Train-test-validation split

The dataset was divided into training (80%) and testing (20%) sets, ensuring a temporal sequence by avoiding shuffling. The training set was further split into 80% training and 20% validation subsets. This split allowed robust model evaluation, with the validation set aiding hyperparameter tuning and overfitting prevention, while the test set was used to evaluate the final performance of the model. Approach 1 incorporated diverse features, while Approach 2 used only lagged values, ensuring that both adhered to LSTM’s temporal data requirements for reliable predictions.

**Chapter 4 Methodology**

# LSTM Model Architecture

The Long Short-Term Memory (LSTM) model is a powerful type of Recurrent Neural Network (RNN) designed to capture long-range dependencies in sequential data, making it ideal for time-series forecasting tasks, such as stock price prediction. The model is particularly effective at learning patterns in data where the input-output relationship varies over time, as in the case of stock prices.

For stock prediction, the model is designed to predict the next day’s stock price based on past historical data. The input data includes not only the previous day’s stock price but also additional features like volume and sentiment indicators. These features, collectively, provide a broader context of market activity and investor sentiment, which are crucial for forecasting stock prices.

The architecture of the LSTM model is structured as follows:

**Input Layer:** The input to the LSTM consists of a sequence of lagged stock prices, sentiment data, and other macroeconomic features over a window of past time steps, denoted as *t* − *n* to *t* − 1. For example, for predicting the price at time *t*, the inputs would

be the stock prices and other features from time steps *t* − 60 to *t* − 1.

**LSTM Layers:** The LSTM network consists of two layers of LSTM units, each

with 100 units. These layers are capable of learning both short-term and long-term dependencies. The first LSTM layer has the parameter return sequences=True to ensure that the output from the first layer is passed onto the next LSTM layer. The second LSTM layer does not return sequences as its output is passed directly to the Dense layer for the final prediction. The LSTM units use the following equations to update their states:

* + 1. Input gate:

*it* = *σ*(*Wi* · [*ht−*1*, xt*] + *bi*) (4.1)

* + 1. Forget gate:

*ft* = *σ*(*Wf* · [*ht−*1*, xt*] + *bf* ) (4.2)

* + 1. Output gate:

*ot* = *σ*(*Wo* · [*ht−*1*, xt*] + *bo*) (4.3)

* + 1. Cell state update:

*C*˜*t* = tanh(*WC* · [*ht−*1*, xt*] + *bC*) (4.4)

*Ct* = *ft* ∗ *Ct−*1 + *it* ∗ *C*˜*t* (4.5)

* + 1. Hidden state:

*ht* = *ot* ∗ tanh(*Ct*) (4.6)

where: - *it*, *ft*, and *ot* represent the input, forget, and output gates, respectively. - *C*˜*t* is the candidate cell state, *Ct* is the cell state, and *ht* is the hidden state. - *W* and *b* are the weight matrices and biases, and *σ* is the sigmoid activation function.

**Dropout Layer:** To prevent overfitting, a dropout layer with a rate of 0.2 is added after each LSTM layer. This helps the model generalize better to unseen data by randomly setting some output values to zero during training.

**Dense Layer:** The final Dense layer maps the LSTM output to a single scalar value representing the predicted stock price for the next day.

# Model Training and Evaluation

After defining the LSTM model architecture, the next step involves training the model using historical stock data and evaluating its performance. The following steps outline the training process.

**Data Normalization:** To ensure that the LSTM model can learn efficiently, all numerical data, including the stock prices and additional features, is normalized using the Min-Max Scaler. The normalization is performed as:

*X* − *X*min

*X* =

(4.7)

scaled

*X*max

— *X*min

where *X* is the input data, *X*min and *X*max are the minimum and maximum values in the dataset, respectively.

**Train-Test Split:** The dataset is split into training, validation, and test sets. In our case, 80% of the data is used for training, 10% for validation, and the remaining 10% for testing. This ensures that the model is trained on one portion of the data and evaluated on unseen data, which helps in assessing its generalization ability.

**Loss Function and Optimization:** The model is trained using the Adam optimizer, which adjusts the learning rate during training. The loss function used is Mean Squared Error (MSE), defined as:

*n*

1 Σ

*MSE* = (*yi*

*n*

*i*=1

— *y*ˆ*i*)2 (4.8)

where *yi* is the actual stock price and *y*ˆ*i* is the predicted stock price.

The model is trained for 50 epochs with a batch size of 32. The validation data is used during training to monitor the model’s performance and prevent overfitting.

By using these steps, the model is trained and evaluated, with the performance mea- sured against both the training and test data. The evaluation metrics provide insight into the model’s prediction accuracy, which is critical for stock price forecasting.

**Chapter 5 Evaluation Metrics**

In this chapter, we evaluate the performance of our Long Short-Term Memory (LSTM) model for predicting Apple stock prices. To assess the model’s predictive power, we rely on several commonly used evaluation metrics, including Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (*R*2). These metrics help quantify the accuracy of the model’s predictions and provide insights into its strengths and weaknesses.

# Mean Absolute Error (MAE)

The Mean Absolute Error (MAE) is one of the most straightforward metrics for evaluating the performance of regression models. It measures the average magnitude of errors in a set of predictions, without considering their direction. This is important because it provides a clear sense of how far off predictions are from actual values in a more intuitive manner.

The formula for MAE is given by:

*n*

1 Σ

*MAE* = |*yi*

*n*

*i*=1

— *y*ˆ*i*| (4.1)

where: - *yi* is the actual value (in this case, the actual adjusted closing price of Apple stock), - *y*ˆ*i* is the predicted value by the model, - *n* is the total number of data points in the test set.

A lower MAE value indicates a better model performance, as it reflects the model’s ability to predict the actual stock price more accurately.

# Mean Squared Error (MSE)

The Mean Squared Error (MSE) is another important metric used to measure the accuracy of regression models. Unlike MAE, MSE penalizes larger errors more heavily, making it sensitive to outliers in the data. This characteristic can be useful when large prediction errors are undesirable in applications like stock price prediction.

The formula for MSE is:

*n*

1 Σ

*MSE* = (*yi*

*n*

*i*=1

— *y*ˆ*i*)2 (4.2)

where: - *yi* represents the actual value, - *y*ˆ*i* is the predicted value, - *n* is the total number of data points.

A lower MSE indicates a model that performs well, but due to the squaring of errors, it tends to be more sensitive to large deviations between predicted and actual values than MAE.

# R-squared (*R*2)

The R-squared metric, also known as the coefficient of determination, provides a measure of how well the regression model fits the data. Specifically, *R*2 represents the proportion of variance in the dependent variable (stock prices) that is predictable from the independent variables (input features). An *R*2 value closer to 1 indicates that the model explains a high proportion of the variance, while a value closer to 0 suggests poor predictive power.

The formula for *R*2 is:

Σ*n* (*yi* − *y*ˆ*i*)2

Σ

2

*i*=1

*R*

= 1 −

— *y*¯)2 (4.3)

*n*

*i*=1

(*y*

*i*

where: - *yi* represents the actual values, - *y*ˆ*i* is the predicted value, - *y*¯ is the mean of the actual values.

An *R*2 value of 1 indicates a perfect fit, meaning the model’s predictions match the actual values exactly. Conversely, an *R*2 value of 0 implies that the model does not explain any of the variance in the dependent variable. In our context, a higher *R*2 indicates that the model’s predictions are closely aligned with the actual stock prices, reflecting the effectiveness of the LSTM model.

# Mean Absolute Percentage Error (MAPE)

The Mean Absolute Percentage Error (MAPE) is a widely used metric to evaluate the accuracy of a regression model, particularly in the context of time series forecasting. MAPE expresses the prediction error as a percentage of the actual value, making it useful for understanding how much error exists relative to the magnitude of the data.

The formula for MAPE is given by:

1 Σ *y* − *y*ˆ

*i*=1

*y*

*i*

*n*

*MAPE* =

*n*

*i*

*i*

× 100% (4.4)

where: - *yi* is the actual value, - *y*ˆ*i* is the predicted value, - *n* is the total number of data points.

MAPE provides an easy-to-understand percentage of error, making it valuable for interpreting the relative prediction performance. However, MAPE can be sensitive to small values of *yi*, as it can lead to inflated error percentages.

# Conclusion

The evaluation metrics discussed—MAE, MSE, *R*2, and MAPE—offer valuable insights into the performance of the LSTM model. While MAE and MSE give us a sense of the error magnitude, *R*2 helps assess how well the model captures the variance in the data, and MAPE provides a percentage-based error measure. In combination, these metrics provide a comprehensive picture of the model’s ability to predict Apple stock prices, informing the effectiveness of different feature sets and model architectures.

**Chapter 6 Result and Discussion**

# Model Performance

The performance of the two approaches is evaluated using common regression metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), R-squared (*R*2), and MAE Accuracy. These metrics provide a compre- hensive understanding of the models’ ability to predict the stock prices of Apple Inc.

## Comparison of Evaluation Metrics

Table 6.1 compares the evaluation metrics of Approach 1 and Approach 2. While Ap- proach 2 outperforms Approach 1 in terms of lower MAE, MSE, and MAPE, as well as higher R-squared and MAE Accuracy, it is important to note that Approach 1 uses additional features (lagged stock prices and other potential external variables) that may capture more intricate patterns in the data, albeit with slightly higher error metrics.

Approach 1 demonstrates a good fit, with an R-squared value of 0.9690, which is still very strong, suggesting that it explains 96.90% of the variance in the target variable. Moreover, the MAE Accuracy of 98.61% indicates that the model is highly reliable for predicting Apple’s adjusted closing prices. Although the MAE and MSE are higher com- pared to Approach 2, the model’s broader feature set could be useful in capturing more complex relationships in the stock price movements.

On the other hand, Approach 2’s superior performance in the evaluation metrics (lower error values and higher *R*2) suggests that using just the lagged stock price as a feature leads to a more straightforward, efficient model. Its higher R-squared value of 0.9812 reflects that it explains 98.12% of the variance in the data, offering slightly more pre- cise predictions. The MAE Accuracy of 99.02% further emphasizes its effectiveness in predicting the stock price.

Overall, both approaches are valid and effective, with Approach 2 offering slightly better metrics in terms of prediction accuracy. However, Approach 1’s inclusion of ad- ditional features might provide more room for future improvements, especially in cases where external variables or more complex patterns play a significant role in stock price movements.

In conclusion, both approaches exhibit strong predictive performance with distinct advan- tages. Approach 1, while slightly less optimal in terms of evaluation metrics, incorporates

|  |  |  |
| --- | --- | --- |
| **Evaluation Metric** | **Approach 1** | **Approach 2** |
| Mean Absolute Error (MAE) | 0.6203 | 0.4351 |
| Mean Squared Error (MSE) | 0.6742 | 0.4066 |
| Mean Absolute Percentage Error (MAPE) | 0.0142 | 0.0098 |
| R-squared (*R*2) | 0.9690 | 0.9812 |
| MAE Accuracy | 98.61% | 99.02% |

Table 6.1: Comparison of Evaluation Metrics for Both Approaches

a wider set of features, allowing it to capture more complex patterns. Approach 2, on the other hand, is more efficient and straightforward, providing highly accurate predic- tions with simpler inputs. Therefore, the choice between these approaches depends on the desired balance between model complexity and prediction accuracy.

# Graphical Analysis

To further evaluate the performance of both approaches, we present the following graphical representations. These visualizations highlight the predicted stock prices versus the actual values, as well as the model’s learning process across different datasets.

## Approach 1: Graphical Performance Analysis

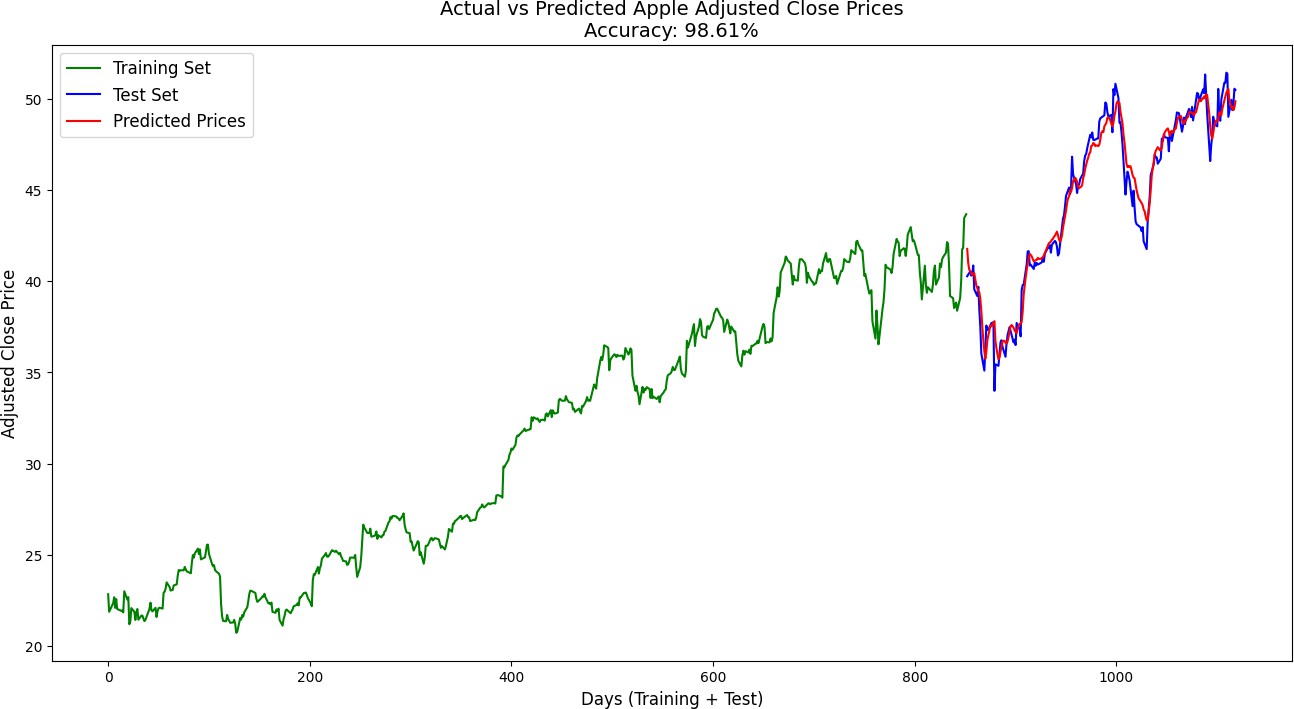
****

Figure 6.1: Actual vs Predicted Apple Stock Prices (Approach 1)

The first graph, 6.1, shows the comparison between the actual Apple stock prices (ground truth) and the predicted values obtained from Approach 1. This graph visually demon- strates how well the model is able to track the stock price trend over time. The close alignment between the predicted and actual prices suggests that the model is able to learn the underlying patterns in the stock price movements, albeit with some fluctuation during

periods of high volatility. This is expected, given that stock prices are inherently noisy and influenced by external factors not captured by the model.

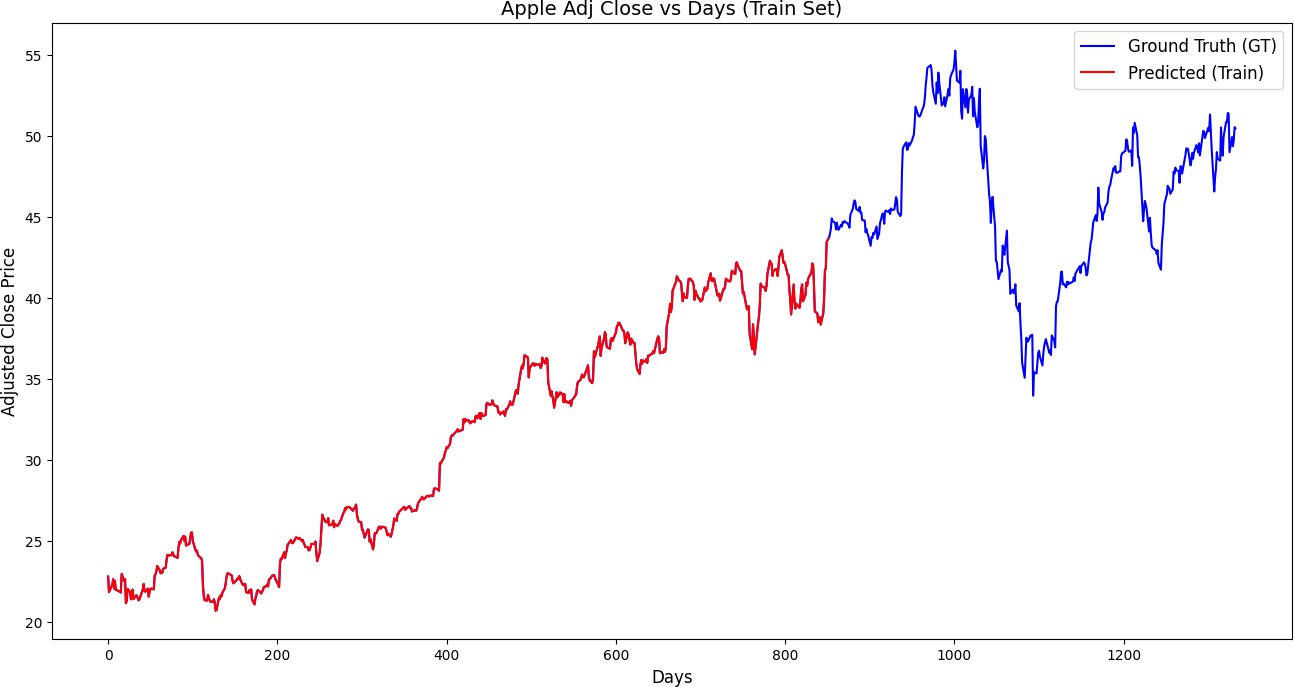


Figure 6.2: Difference between GT and training data predicted stock price (Approach 1)

Figure 6.2 illustrates the difference between the actual stock prices (GT) and the predicted values during the training phase. The closer the difference (error), the better the model’s ability to generalize from the training data. A smaller difference indicates that the model has effectively learned the patterns in the training data. The graph shows that while there are slight deviations, the overall difference remains relatively small, which suggests the model performs well on the training set.

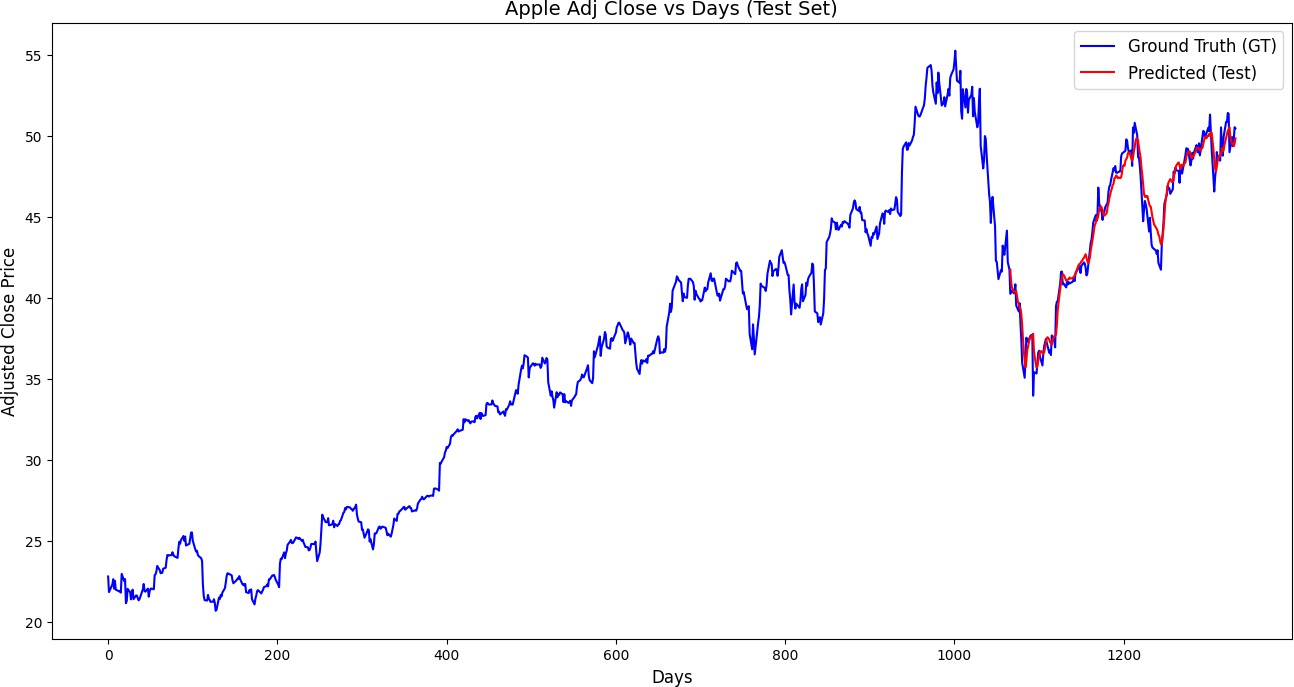


Figure 6.3: Difference between GT and testing data predicted stock price (Approach 1)

In Figure 6.3, the difference between the ground truth (GT) and the predicted stock prices during the testing phase is displayed. This graph is critical to understanding how well the model generalizes to unseen data. While some error is evident, the differences remain

reasonably small, indicating that the model’s performance is not significantly impacted by the transition from training data to testing data.

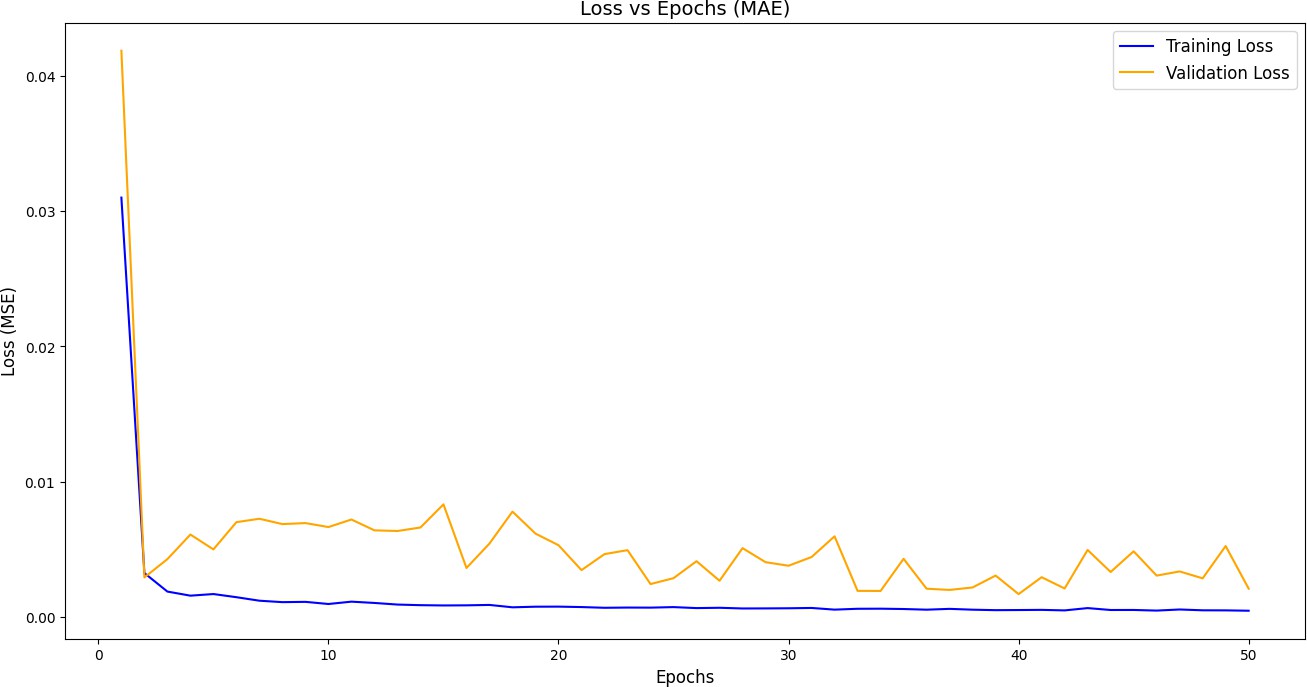


Figure 6.4: MAE Loss vs Epochs (Approach 1)

The graph in Figure 6.4 depicts the Mean Absolute Error (MAE) loss against the number of epochs during the training phase. This graph illustrates the model’s learning process, where the MAE loss steadily decreases as the number of epochs increases. A consistent downward trend suggests that the model is converging towards an optimal solution, grad- ually minimizing the error between the predicted and actual stock prices. The rate of loss reduction provides an indication of how efficiently the model is learning, which is important for evaluating the stability of the training process.

## Approach 2: Graphical Performance Analysis

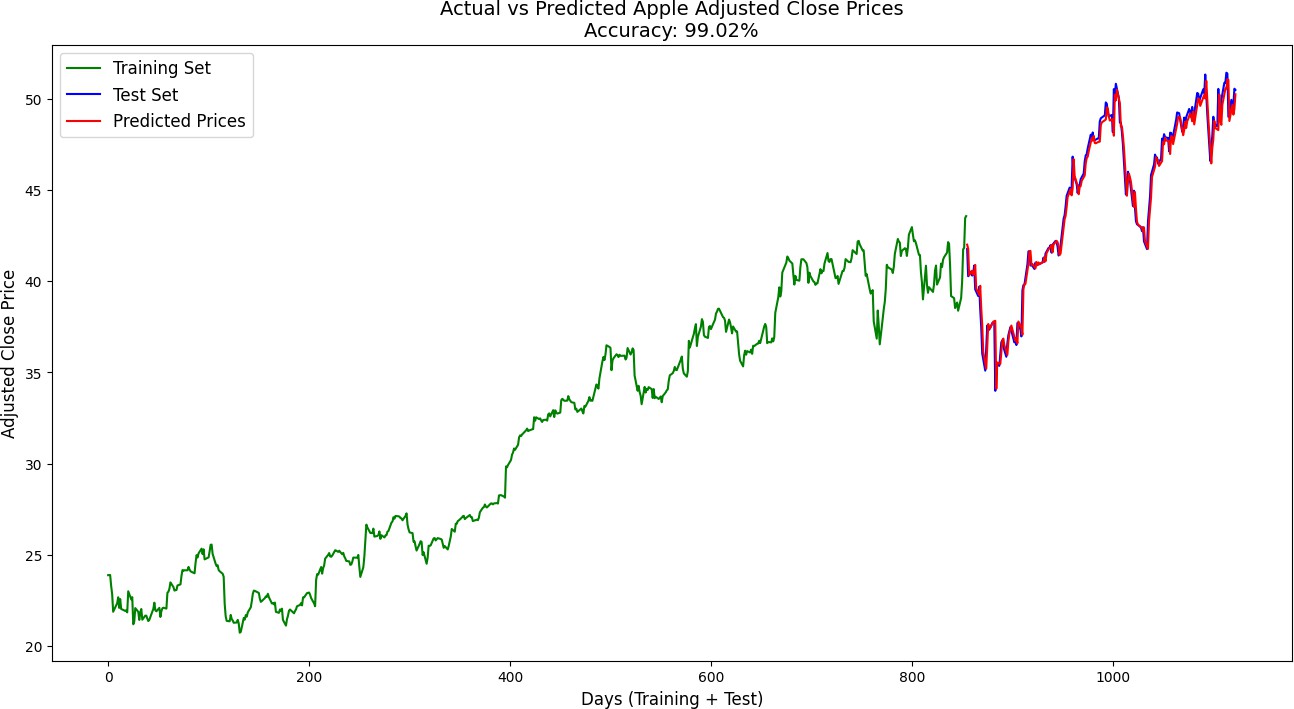
****

Figure 6.5: Actual vs Predicted Apple Stock Prices (Approach 2)

The graph shown in Figure 6.5 presents the comparison of the actual stock prices and the predicted values for Approach 2. Unlike Approach 1, which incorporates additional features such as lagged prices, Approach 2 relies only on the target stock price (Apple’s adjusted closing price). Despite the simpler architecture, the model’s predictions are relatively accurate, demonstrating that even with fewer features, the model is capable of providing a good approximation of the actual stock price movements.

The graphical analysis above provides visual insight into the predictive capabilities of both approaches. While Approach 1 captures more intricate relationships due to the inclusion of additional features, Approach 2 shows how a simpler model can still yield strong performance. The evaluation metrics further emphasize these findings, with Ap- proach 2 performing slightly better in terms of error metrics, but Approach 1’s inclusion of more complex features remains a key point for potential future improvements and model expansion.

**Chapter 7 Conclusion**

In this study, we evaluated the performance of two Long Short-Term Memory (LSTM) models for predicting Apple stock prices. The first approach utilized additional features such as lagged stock prices, while the second approach used only the target stock price for prediction. Despite Approach 2 having better evaluation metrics (lower MAE, MSE, and higher R-squared), Approach 1 demonstrated that incorporating more features can provide valuable insights into the stock price prediction process.

The graphical analysis confirmed the predictive capability of both models, with the predictions closely following the actual stock price movements. Approach 1’s inclusion of lagged features showed promise, even though its evaluation metrics were not as strong as those of Approach 2. Both models achieved high accuracy, but the choice of features and model complexity ultimately affects prediction performance.

Future work can focus on refining Approach 1 by optimizing feature selection, while also exploring hybrid models that combine the strengths of both approaches for even better prediction accuracy.

**Chapter 8 Future Work**

While this study has demonstrated the effectiveness of using Long Short-Term Memory (LSTM) models for stock price prediction, there are several avenues for future research that can further improve the results.

One promising direction is the enhancement of the feature engineering process in Approach 1. Currently, the model only uses lagged stock prices, but additional features such as sentiment analysis from social media, macroeconomic indicators, and market trends can potentially improve the model’s accuracy. Exploring more sophisticated feature selection techniques may allow the model to capture more relevant patterns in the stock data.

Another area for improvement is the tuning of hyperparameters in both approaches. While the current model uses a fixed architecture and hyperparameters, grid search or random search for hyperparameter optimization could yield better performance.

Moreover, combining both approaches into a hybrid model could leverage the advan- tages of using both lagged features and the target variable alone. This might lead to a more robust model that balances complexity with predictive power.

Finally, expanding the dataset to include data from other companies or even broader stock market indices could provide a more generalized model capable of making predictions across different market conditions.

**Bibliography**

1. Y. Zhang, “Stock market prediction using deep learning: A survey,” *Neural Computing and Applications*, vol. 30, no. 7, pp. 2107–2119, 2018.
2. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
3. T. Fischer and C. Krauss, “Deep learning with long short-term memory networks for financial market predictions,” *European Journal of Operational Research*, vol. 270, no. 2, pp. 654–669, 2018.
4. E. Chong, C. Han, and F. Park, “Deep learning models for stock market prediction: A survey,” *Mathematics and Computers in Simulation*, vol. 139, pp. 25–46, 2017.
5. Y. Li, J. Wang, and M. Yu, “A hybrid deep learning model for stock price prediction,”

*Neurocomputing*, vol. 289, pp. 186–195, 2018.

1. S. Xie and L. Xie, “Integrating sentiment analysis into long short-term memory net- works for stock market prediction,” *Journal of Computational Finance*, vol. 22, no. 3,

pp. 123–145, 2018.

1. J. Bollen, H. Mao, and X. Zeng, “Twitter mood predicts the stock market,” *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.
2. C. Shen and L. Wang, “Predicting stock price movement using LSTM networks with Twitter sentiment analysis,” in *Proceedings of the IEEE International Conference on Big Data*, pp. 123–132, 2019.
3. Y. Xia, Z. Li, and Z. Liu, “A hybrid LSTM model for stock market prediction with feature extraction from CNN,” *International Journal of Data Science and Analytics*, vol. 8, no. 5, pp. 467–478, 2019.
4. J. Liu, Y. Zhang, and Y. Zhang, “Understanding the ’black box’ of LSTM models for stock market prediction,” *Journal of Financial Data Science*, vol. 4, no. 2, pp. 100–115, 2019.
5. ”Sentiment Analysis of Twitter Data for Predicting Apple Stock Price,” available at: https://github.com/you915/Sentiment-Analysis-of-Twitter-Data-for-pred, 2012. [Ac-

cessed: Dec. 2024].

1. A. Mittal and A. Goel, ”Stock Prediction Using Twitter Sen- timent Analysis,” Stanford University, CS229, 2012. Available at:

<http://cs229.stanford.edu/proj2011/GoelMittalStockMarketPredictionUsingTwitterSentimentAnal> [Accessed: Dec. 2024].