City, University of London

MSc in Data Science

Department of Computer Science

Project Report

October 2023

Integrated Approach for Stock Price Prediction Using LSTM, Incorporating Technical Indicators, Fundamental Analysis, and Sentiments of News

Author: Deenadhayalan Ravi

Supervisor: Olga Galkin

Submitted: 16th October 2023

Declaration

By submitting this work, I declare that this work is entirely my own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the assessment instructions and any other relevant programme and module documentation. In submitting this work I acknowledge that I have read and understood the regulations and code regarding academic misconduct, including that relating to plagiarism, as specified in the Programme Handbook. I also acknowledge that this work will be subject to a variety of checks for academic misconduct.

Signed:

Deenadhayalan Ravi

Abstract

This study investigates integrating machine learning and natural language processing techniques for Apple stock price prediction. Long Short-Term Memory (LSTM) networks are utilized with varied feature combinations: technical indicators, financial ratios, and sentiment analysis of 10 years of financial news from The New York Times. For sentiment analysis, Large Language Models (LLM) like ChatGPT3.5 are compared against BiLSTM+Word2Vec and FinBERT using the financial_phrasebank dataset and unlabelled NYTimes corpus.

Results reveal LSTM with news sentiment outperforms LSTM with just technical and financial indicators. Among the NLP models, ChatGPT3.5 surpasses others in sentiment analysis accuracy. For stock price forecasting, ChatGPT3.5-derived sentiment combined with technical indicators yields the best performance - 2.76% MAPE and 90% predictions within +/-5% confidence band.

The key contributions are the in-depth examination of incorporating domain-specific news sentiments using state-of-the-art NLP, and the consistent improvement induced by hyperparameter optimization. Although promising results are achieved, expanding the experimental scope across more stocks and sectors is imperative to validate model versatility. Overall, the integrative methods provide unique insights, contributing significantly to the intricate domain of stock price prediction.

Key Words: Stock Price Prediction, Machine Learning model - LSTM, Sentiment Analysis, Large Language Model - ChatGPT3.5, 10 years of New York Times Apple News data

Table of Contents

List of Tables 6

List of Figures 6

1. Introduction and Objectives 9

1.1 Background and Motivation 9
1.2 Objectives of the Study 9
1.3 Significance and Beneficiaries of the Study 10
1.4 Research Questions 11
1.5 Project Overview 11
1.6 Structure of the Report 14

2. Context 15

2.1 Stock Price Prediction: An Overview 15
2.2 Machine Learning Models and Features used in Stock Price Prediction 15
2.3 Literature Survey on high-performance features: 19
2.3.1 Best Technical Indicators for stock price predictions 19
2.3.2 Best Financial Indicators for stock price prediction 21
2.3.3 Literature Exploration on Algorithms for Sentiment Analysis 22
2.4 Gaps in the Current Literature 24
2.5 Summary of the Literature Review 25

3. Methods 27

3.1 Introduction / Overview 27
3.2 Data collection and pre-processing 29
3.2.1 Data Collection, Sources Identification, and nature of data 29
3.2.2 Dataset preparation (A, B, C, D1, D2, D3) 37
3.2.3 Data Cleaning and Pre-processing Procedures for Sentiment Analysis 39
3.2.4 Post-processing of NLP model outputs to merge with the other features: 41
3.3 Feature Selection and Data splitting 42
3.4 Stock Price Prediction Models 47
3.4.1 Random Forest (RF) Regressor 47
3.4.2 Long Short-Term Memory (LSTM) Model 49

3.5.3 Hyper Parameter Optimized LSTM Model 52
3.5 Sentiment Analysis of Financial News 54
3.5.1 Bi-LSTM with Word2Vec 54
3.5.2 FinBERT 56
3.5.3 Large Language Model (LLM) – ChatGPT-3.5 57
3.6 Validation and Performance Metrics 58
3.6.1 NLP Models on labelled dataset 58
3.6.2 NLP Models on non-labelled dataset 60
3.6.3 Stock Price Prediction Models 62
4. Results 65
4.1 Sentiment Analysis Model's Performance 65
4.1.1 Senti Analysis 1 - Bi-LSTM (word2vec) on financial_phrasebank dataset 65
4.1.2 Senti Analysis 2 - FinBERT on financial_phrasebank dataset 67
4.1.3 Senti Analysis 3 - LLM on financial_phrasebank dataset 69
4.1.4 Senti Analysis 1.1, 2.1, and 3.1 all models on New York Times dataset 71
4.2 Stock Price Prediction Model's Performance 74
4.2.1 Baseline Model Performance: Output 1 74
4.2.2 LSTM with Dataset_A: Output 2 & 3 75
4.2.3 LSTM with Dataset_B: Output 4 & 5 77
4.2.4 LSTM with Dataset_C: Output 6 & 7 79
4.2.5 LSTM with Dataset_D1: Output 8 & 9 81
4.2.6 LSTM with Dataset_D2: Output 10 & 11 83
4.2.7 LSTM with Dataset_D3: Output 12 & 13 85
4.3 Consolidated Results 88
5. Discussion 89
5.1 Evaluation of Objectives 89
5.2 Comparing with the Literature Review: 97
6. Evaluation, Reflections, and Conclusions 98
6.1 Evaluation of the Study 98
6.2 Reflections on the Research Process 98
6.3 Conclusions 100
6.4 Recommendations for Future Work 101

List of Tables

Table 1: Best Model’s hyperparameters - consolidated results

Table 2: Stock Price Prediction - consolidated results

Table 3: Classification Report for NLP models on financial_phrase bank dataset

Table 4: Consolidated Classification Report for Sentiment Analysis

List of Figures

Figure 1: Project Overview – Main Task

Figure 2: Project Overview – Sub Task

Figure 3: General Methodology

Figure 4: Overview of Methodological Approach

Figure 5: Historical stock price data

Figure 6: Technical Indicators

Figure 7: Financial Statements

Figure 8: NLP – text pre-processing

Figure 9: General Flowchart for choosing a statistical test

Figure 10: Flowchart for choosing a statistical test.

Figure 11: Correlation matrix of Dataset_C

Figure 12: Correlation with close price for Dataset_C

Figure 13: Random Forest Regressor – working flowchart

Figure 14: LSTM Architecture

Figure 15: Sliding and Expanding window cross validation

Figure 16: Close Price vs Cumulative sentiment over time - example

Figure 17:Performance Metrics 2 - example 1

Figure 18: Performance Metrics 2 - example 2

Figure 19: Classification Report – Bi-LSTM with word2vec

Figure 20: Class Distribution - True sentiments (left) and Predicted Sentiments (right) – Bi-LSTM word2vec

Figure 21: Confusion Matrix – Bi-LSTM with word2vec

Figure 22: Classification Report - FinBERT

Figure 23: : Class Distribution - True sentiments (left) and Predicted Sentiments (right) - FinBERT

Figure 24: Confusion Matrix - FinBERT

Figure 25: Classification Report – ChatGPT3.5

Figure 26: Class Distribution - True sentiments (left) and Predicted Sentiments (right) – ChatGPT3.5

Figure 27: Confusion Matrix - ChatGPT3.5

Figure 28: Class distribution - Bi-LSTM with word2vec (left), FinBERT(center), LLM-ChatGPT3.5(right)

Figure 29: Close Price vs Cumulative sentiment over time - Bi-LSTM with word2vec

Figure 30: Close Price vs Cumulative sentiment over time – FinBERT

Figure 31: Close Price vs Cumulative sentiment over time - ChatGPT3.5

Figure 32: Experiment 1 – Model Prediction vs True Value

Figure 33: Exp-2 Expanding Window CV performance Figure 34: Exp-2 Train loss vs Val loss

Figure 35: Experiment 2 – Model Prediction vs True Value

Figure 36: Experiment 3 – Model Prediction vs True Value

Figure 37: Exp-4 Expanding Window CV performance Figure 38: Exp-4 Train loss vs Val loss

Figure 39: Experiment 4 – Model Prediction vs True Value

Figure 40: Experiment 5 – Model Prediction vs True Value

Figure 41: Exp-6 Expanding Window CV performance Figure 42: Exp-6 Train loss vs Val loss

Figure 43: Experiment 6 – Model Prediction vs True Value

Figure 44: Experiment 7 – Model Prediction vs True Value

Figure 45: Exp-8 Expanding Window CV performance Figure 46: Exp-8 Train loss vs Val loss

Figure 47: Experiment 8 – Model Prediction vs True Value

Figure 48: Experiment 9 – Model Prediction vs True Value

Figure 49: Exp-10 Expanding Window CV performance Figure 50: Exp-10 Train loss vs Val loss

Figure 51: Experiment 10 – Model Prediction vs True Value

Figure 52: Experiment 11 – Model Prediction vs True Value

Figure 53: Exp-12 Expanding Window CV performance Figure 54: Exp-12 Train loss vs Val loss

Figure 55: Experiment 12 – Model Prediction vs True Value

Figure 56: Experiment 13 – Model Prediction vs True Value

Figure 57: Results of RF Regressor vs LSTM on Dataset_A

Figure 58: Results of Parametric Study

Figure 59: Results of non-optimized vs optimized LST models

Figure 60: Examination of Influence of Sentiment Analysis on Stock Price Prediction

1. Introduction and Objectives

1.1 Background and Motivation

This chapter delves into the rationale, background, and driving forces behind my selection of this subject matter for my dissertation. I have garnered experience and knowledge in the sales and marketing domain of mechanical products for the last six years. In my initial years, without a solid understanding of stock markets, I ventured into investing. Despite some gains, I incurred more losses, prompting me to retreat and reflect on the importance of knowledge in navigating such complex terrains as stock trading. The intricate nature of stock trading, with its dependence on multifarious factors and its susceptibility to volatile market dynamics, sparked my curiosity about stock price prediction. The unpredictable movements and the myriad of influencing elements make stock price prediction a complex, yet intriguing field of study.

Now, having pursued a course in Data Science and acquired a substantial understanding of various concepts, I found myself revisiting this unsolved, mystifying question of predicting stock prices. I delved into how academic concepts and computational skills could be leveraged, combined with my business acumen, to unravel the complexities of stock price prediction. My exploration led me to identify two pivotal areas for effective prediction: first, understanding the factors influencing stock pricing, and second, honing the computational skills essential for analysis and prediction.

1.2 Objectives of the Study

In this study, we aim to scrutinize the interplay between the identified domains of influence and computational analysis. The specific objectives are as follows:

- Assess the performance a neural network model, Long Short-Term Memory - LSTM, in predicting the stock prices of Apple Inc.
- Investigate the impact of hyperparameter optimization on the performance of these models.
- Investigate the impact of the following elements in the company’s stock price prediction.
    - ➢ Technical indicators - which are statistical methods of prediction based on fundamental historical data.
    - ➢ Financial statements - released by the company annually which provide insights about the company’s financial position.
    - ➢ News articles - news about the company in the daily newspaper.

- Sentiment Analysis Performance: Evaluate the effectiveness of the below mentioned three Natural Language Processing (NLP) models in determining the sentiments of news articles from newspapers.
    - ➢ Bi-LSTM with word2vec
    - ➢ FinBERT
    - ➢ LLM – ChatGPT3.5

1.3 Significance and Beneficiaries of the Study

The significance of this research are as follows,

- **In-depth Feature Analysis:** Conducting an in-depth investigation into the effects of technical indicators, financial statements, and sentiment from news articles establishes a comprehensive method for predicting stock prices, thereby strengthening the reliability of forecasting models. Notably, this study sets itself apart by using an LLM-ChatGPT3.5 for predicting the sentiments on a decade's worth of news articles extracted from The New York Times, specifically related to Apple Inc. To my knowledge, such a scope has not been previously explored by other researchers in the field.
- **Model Optimization and Comparison:** The study contributes to refining predictive accuracy through hyperparameter optimization and comparison between different machine learning models, offering valuable insights into their practical application.
- **Academic and Practical Contribution:** The research fills existing gaps in academic literature and has practical implications by aiding investors and financial analysts in making informed decisions, thus impacting investment strategies and risk management.

The beneficiaries of this work are as follows,

- **Investors and Financial Analysts:** Individual and institutional investors, as well as financial analysts, stand to gain from improved predictive models, which can aid in developing investment strategies and mitigating risks.
- **Academic Researchers:** The study serves as a valuable resource for scholars and researchers focusing on finance and data science, paving the way for further research and development in stock price prediction.
- **Technology Developers and Companies:** Fintech companies and developers can integrate the insights derived from this research to enhance the functionality and accuracy of their

trading platforms and tools, benefiting companies in their financial analysis and strategic planning.

1.4 Research Questions

The research questions are,

RQ1: Justification and Comparative Analysis of Model Selection: Why are the LSTM model chosen over a traditional model like the RF regressor for stock price prediction?

RQ2: Impact of Feature Diversification on Predictive Accuracy: How does the combination of features (technical indicators, financial indicators/ratios, and sentiment from news) influence the accuracy and reliability of stock price predictions across different datasets?

RQ3: Comparative Model Efficacy in Sentiment Analysis: How do different NLP models (Bi-LSTM with word2vec, FinBERT, and ChatGPT3.5) compare in terms of accuracy and effectiveness when applied to sentiment analysis of financial news, and how does their performance translate when applied to unlabelled New York Times dataset?

RQ4: Influence of Sentiment Analysis on Predictive Modeling: To what extent does incorporating sentiment analysis of financial news, obtained through different NLP models, enhance the performance of LSTM models in predicting stock prices?

RQ5: Optimization and Model Robustness: How does hyperparameter optimization impact the performance of LSTM models across various experiments with diversified datasets, and does the optimized model consistently outperform its non-optimized counterpart across all datasets?

1.5 Project Overview

Main Task

The methodology underpinning this study is a blend of rigorous data sourcing, in-depth analysis, and advanced modeling techniques of machine learning and natural language processing. At the foundation of our approach is a multifaceted data collection strategy. This encompasses historical stock prices of Apple Inc. for 10 years, derived technical indicators based on historical data, 10 years of financial statements from Apple Inc.'s annual reports, and 10 years of news articles about Apple from New York Times. With these four different verticals of data related to Apple Inc., we prepare the datasets A, B, C, D1, D2, D3 inline with the objective and research questions, before this dataset preparation, the news article goes as an input to a sub task of sentiment analysis using three sophisticated Natural Language Processing (NLP) models and the output sentiments are used for dataset preparation. Following this feature selection using statistical test for significance to decide the features to be used in model and data split for modeling. Then Structuring the regression models like RF regressor (just to test with only one dataset_A) and LSTM for stock price prediction. Test all the datasets with the basic LSTM model and do hyperparameter optimization for all the datasets A, B, C, D1, D2, D3 and pull out the best models results from each dataset and compare the results of all and answer the research questions and objectives. The below pictorial representation gives more clarity,

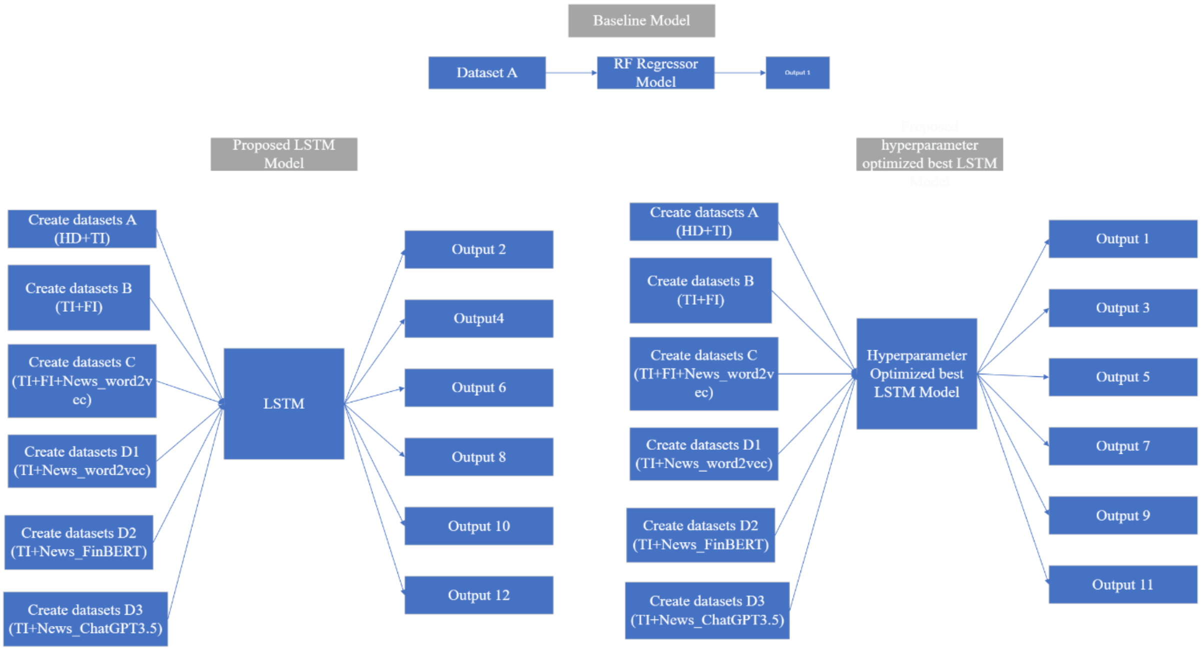

Baseline Model
Dataset A
RF Regressor Model
Output 1
Proposed LSTM Model
Create datasets A (HD+TI)
Create datasets B (TI+FI)
Create datasets C (TI+FI+News_word2vec)
Create datasets D1 (TI+News_word2vec)
Create datasets D2 (TI+News_FinBERT)
Create datasets D3 (TI+News_ChatGPT3.5)
LSTM
Output 2
Output4
Output 6
Output 8
Output 10
Output 12
Proposed hyperparameter optimized best LSTM Model
Create datasets A (HD+TI)
Create datasets B (TI+FI)
Create datasets C (TI+FI+News_word2vec)
Create datasets D1 (TI+News_word2vec)
Create datasets D2 (TI+News_FinBERT)
Create datasets D3 (TI+News_ChatGPT3.5)
Hyperparameter Optimized best LSTM Model
Output 1
Output 3
Output 5
Output 7
Output 9
Output 11

Figure 1: Project Overview – Main Task

Sub Task

The NYTimes data is non-labelled, in order to compare the performance of the three NLP model – Bi-LSTM with word2vec, FinBERT, and ChatGPT3.5 we introduced a financial_phrasebank dataset – which is a famous financial news dataset, and trained and tested the models and compared the results. Then used these models in the NYTimes data to get the sentiments as outputs and these outputs are connected to the dataset preparation of the main task. The below pictorial representation gives more clarity,

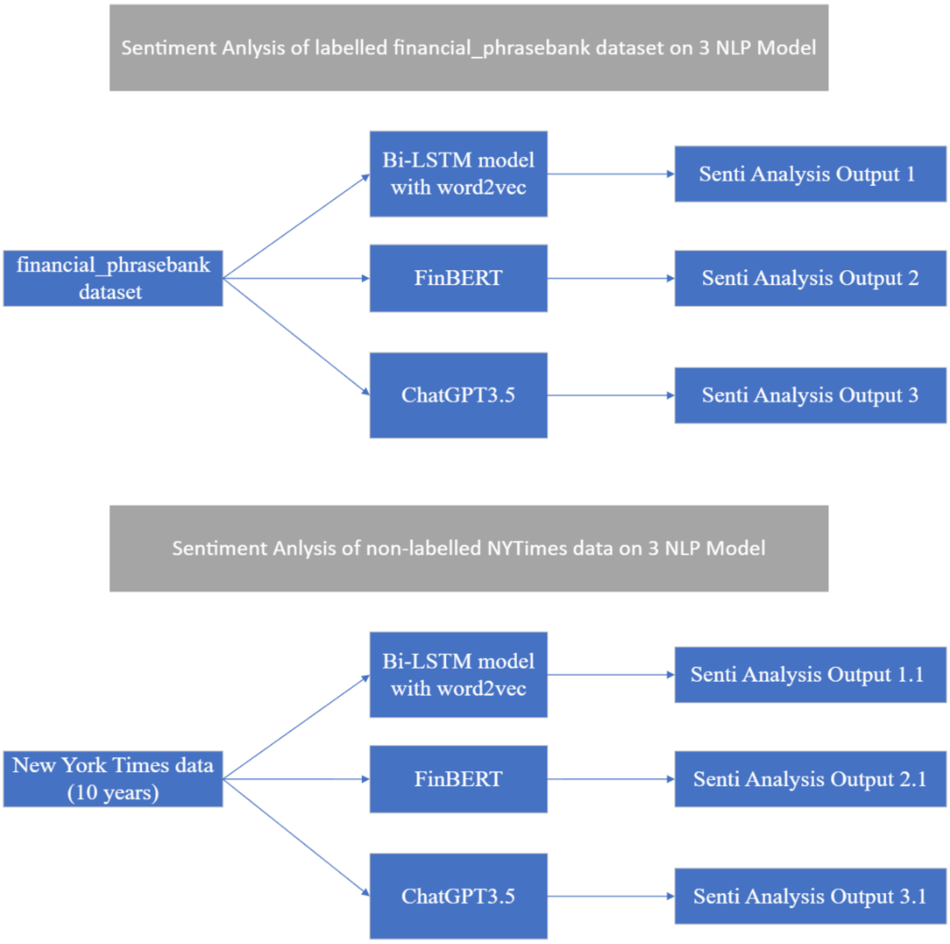

Sentiment Anlysis of labelled financial_phrasebank dataset on 3 NLP Model
financial_phrasebank dataset
Bi-LSTM model with word2vec
FinBERT
ChatGPT3.5
Senti Analysis Output 1
Senti Analysis Output 2
Senti Analysis Output 3
Sentiment Anlysis of non-labelled NYTimes data on 3 NLP Model
New York Times data (10 years)
Bi-LSTM model with word2vec
FinBERT
ChatGPT3.5
Senti Analysis Output 1.1
Senti Analysis Output 2.1
Senti Analysis Output 3.1

Figure 2: Project Overview – Sub Task

Major Change in the plan during the project:

Originally, the project was started with a plan that the Dataset_C with all three different features including technical indicators + financial indicators + sentiments from news will be giving the best prediction, but during the analysis we found that financial indicators are pulling the models performance down, which paved the way for the dataset_D1 with only technical indicators and news sentiments. That is one advancement.

Then, when exploring the techniques of NLP for sentiment analysis initially I have built my own Bi-LSTM model with word2vec word embedding techniques and used this in Dataset C and D1. After the project has extended in pulling the sentiments with best accuracy that is where we came across two more models - financial domain specific FinBERT and the NLP's latest technological improvement of the era LLM – ChatGPT3.5, that is how we created Dataset_D2 and Dataset_D3, respectively.

1.6 Structure of the Report

The ensuing report is structured as follows:

- Chapter 2: Context/Literature Review – Provides an overview of the current state of stock price prediction using machine learning models and different features, discussing relevant theories, practices, and prior works.
- Chapter 3: Methods – Delineates the methodologies employed, detailing data gathering, dataset preparation for the experiments, data cleaning, feature selection, model application, optimization processes and validation metrics.
- Chapter 4: Results – Presents the findings and outputs from applying the RF and LSTM models on the different datasets with a consolidated results at the end of the section.
- Chapter 5: Discussion – Analyses the results in relation to the objectives and within the broader context of existing literature.
- Chapter 6: Evaluation, Reflections, and Conclusions – Offers a comprehensive evaluation of the project, reflecting on the learning and proposing recommendations and future work.

Following the main chapters, the report includes a Glossary, References.

2. Context

This chapter delves into the evolution of stock price prediction, offering insights into the foundational theories, prevailing practices, and prior seminal works that have shaped this intricate field.

When I started this chapter, I had limited knowledge on the stock price prediction using machine learning methods, from the following literature survey, I acquired what people has done on this topic in the past and what is the gap or areas where I can explore more.

2.1 Stock Price Prediction: An Overview

Stock price prediction has historically been a focal point in the financial world, with traders, analysts, and investors constantly seeking ways to foresee market movements. Traditionally, this endeavour leaned heavily on methods like fundamental analysis, which relies on company-specific data (e.g., earnings, assets, liabilities) and technical analysis that prioritizes past market data to forecast future prices. However, predicting stock prices has always been complex, owing to the plethora of factors that can sway them, ranging from macroeconomic indicators and geopolitical events to company-specific news.

With the advent of technology and an increase in available data, the methods for predicting stock prices have evolved. The challenges presented by the multi-faceted nature of financial markets, coupled with the vast volume of data, have driven researchers and practitioners to seek more sophisticated, data-driven models to gain an edge in forecasting.

In this literature review, what we mainly focused on each paper is, what are the features, models, performance metrics they have used for stock price prediction, and of course the results which model has done better? And if there are some different approach or methodologies or techniques used?

The following survey is structured in such a way that first discussed few papers and wrote the inference from those papers, like that it goes on.

2.2 Machine Learning Models and Features used in Stock Price Prediction

Hota, Chakravarty, Paikaray & Bhoyar (2022) in their paper ‘Stock Market Prediction Using Machine Learning Techniques’ they analysed 5 years of American Airlines stock price data with Artificial Neural Networks (ANN) and with some machine learning models like Decision

Tree, Support Vector Regression and Random Forest. They chose only the market open price as input feature to the models. In this analysis Random Forest and ANN model performed better with 0.36 and 0.37 Mean Absolute Percentage Error (MAPE) respectively.

Vijh, Chandola, Tikkiwal & Kumar (2020) in their paper ‘Stock Closing Price Prediction using Machine Learning Techniques’ they analysed 10 years of stock prices data for five companies and predicted stock prices using six new variables derived from the Open, High, Low, and Close values of each company. These new variables included the difference between High and Low (H-L), the difference between Open and Close (O-C), 7-day moving average (MA), 14-day MA, 21-day MA, and 7-day standard deviation (STD Dev). Subsequently, these variables were used as inputs for both Artificial Neural Network (ANN) and Random Forest (RF) models to predict the closing price. The results showcased ANN's slight advantage over RF in stock price prediction, notably for Pfizer Inc. ANN registered an RMSE of 0.42, MAPE of 0.77%, and MBE of -0.0156. In contrast, RF recorded an RMSE of 0.43, MAPE of 0.8%, and MBE of -0.0155. These low MAPE values underscore the strong predictive performance of both the models.

Sen (2018) in his paper ‘Stock Price Prediction Using Machine Learning and Deep Learning Frameworks’ he has experimented eight classification models (Logistic Regression, KNN, Decision Tree, Bagging, Boosting, Random Forest, ANN-classifier, SVM) and eight regression models (Multivariate Regression, Decision Tree, Bagging, Boosting, Random Forest, ANN-regressor, SVM, LSTM) to predict the stock prices of Tata Steel and Hero Moto companies. The author utilized eleven features for the prediction: month, day of the month, day of the week, time, open price percentage change, sensex (NIFTY Index) percentage change, low price difference, high price difference, close price difference, volume difference, and range difference. The analysis employed Sensitivity, Specificity, PPV, NPV, and CA as performance metrics for classification models, while correlation and RMSE were used for regression models. The results revealed that Boosting exhibited the best performance among the classification techniques, while in regression models, LSTM achieved superior performance with an RMSE of 0.2 for Hero Moto and 2.36 for Tata Steel.

Inference: The papers mentioned above analyze historical stock price data of companies, employing newly derived features from fundamental prices alongside some technical indicators. Both traditional machine learning models and neural networks are used for this analysis. Results show that the Random Forest (RF), Artificial Neural Network (ANN), and Long Short-Term Memory (LSTM) models all exhibit commendable performance. Particularly noteworthy is the LSTM's superior performance compared to other models in these studies.

Heo & Yang (2016), in their paper titled "Stock Price Prediction Based on Financial Statements Using SVM," utilized a Support Vector Machine (SVM) model to forecast stock prices drawing insights from financial statements. Their study integrated a company's historical stock price data with its quarterly financial statements, focusing on critical indicators such as Earnings per Share (EPS), Book-Value per Share (BPS), Net Profit Growth Rate (NPGR), stock price after one month, stock price after two months, and Investment Intention (rated on a scale from 1 to 5). The model tested various combinations of these variables. Findings from the research indicated that the use of financial statements for predicting stock prices was more effective in the short term than in the long term. Specifically, the combination of EPS & BPS outperformed other combinations, achieving an accuracy of 57.5%, closely followed by the combination of EPS, BPS, and NPGR at 57%.

Boozer, Jr., Rainwater, and Lowe (2017) in their paper titled “Using financial statement variables to predict stock prices: Lessons from the 2007-2009 financial crisis” engaged in a meticulous examination of the extent to which variables from financial statements, including balance sheet, income statement, and cash flow ratios, could prognosticate the fluctuations in the stock prices of American corporations during the 2007-2009 financial turmoil. Engaging a dataset that spanned a decade (2004-2013) and focusing on fourteen companies listed on the S & P 500, their findings illuminated a complex relationship between the predictive capacity of financial analysis and stock prices, particularly in the pre and post-financial crisis contexts. While company size and sales emerged as potent predictors, liquidity, albeit having a modest impact via net working capital, highlighted an overarching trend of an increase in stock following the crisis. Thus, while the analytical model affirmed the prevailing predictive capacities of certain financial statement variables, it also underscored that shifts in these variables neither pre-emptively indicated nor simultaneously signalled directional shifts in stock prices amid the crisis.

Inference: The above two papers provide varied perspectives on the influence of financial statements in predicting stock prices, presenting an inconclusive stance on their actual impact.

Mohan, Mullapudi, Sammeta, Vijayvergia, & Anastasiu (2019) in their paper named "Stock Price Prediction Using News Sentiment Analysis" the authors incorporated News Sentiment Analysis into the stock price prediction process. Two datasets were utilized, one containing daily stock prices of 500 companies and another sourced from an international daily newspaper website. The author developed several models including ARIMA, FB Prophet, LSTM, LSTM with stock prices and Textual Polarity – where the model has been trained on the current polarity and previous stock prices, LSTM with stock prices and Textual Information – where the whole text has been processed and fed into the neural networks along with the price, LSTM Multivariate model – where the stock price and sentiment polarity were fed for 4 consecutive days and the 5th day's stock price is expected as output. The LSTM model with stock prices and Textual Polarity yielded the best performance, achieving a Mean Absolute Percentage Error (MAPE) of 2.03. Following closely were the LSTM and LSTM with stock prices and Textual Information models with MAPEs of 2.13 and 2.17 respectively. ARIMA and FB Prophet models exhibited higher MAPEs at 7.39 and 7.98 respectively. The LSTM Multivariate model, with a MAPE of 10.43, ranked last among the models.

Maqbool, Aggarwal, Kaur, Mittal, & Ganaie (2023) in their paper named "Stock Prediction by Integrating Sentiment Scores of Financial News and MLP-Regressor: A Machine Learning Approach" the author employed three different algorithms to calculate sentiment scores. The analysis involved ten years of historical stock price data and financial news pertaining to four companies from diverse sectors. The algorithms utilized were Valence Aware Dictionary and Sentiment Reasoner (VADER) from the NLTK library, TextBlob, and Flair, both of which belong to Natural Language Processing (NLP). The sentiment scores obtained from these algorithms, in conjunction with the corresponding stock prices, were used as input for the Multiple Layer Perceptron Regressor (MLP-Regressor). Various combinations of the algorithms were considered during the analysis. This paper predicts the 'trend' on the upcoming day using the previous day's price and sentiment scores and then comparing the prices of the next day with the previous day, 'future trend' by comparing the stock prices of one day with the price after n days, the experiment was performed in predicting the stock prices for 10, 30 and 100 days. Among the various combinations tested, the utilization of FLAIR sentiment scores with MLP-Regressor exhibited the best performance, achieving a MAPE of 1.48. Following closely was the combination of FLAIR and TextBlob with MLP, yielding a MAPE of 1.55.

Inference: From the above papers it is evident that sentiment scores of the Financial News, twitter / social media news about companies has an impact on the stock prices.

2.3 Literature Survey on high-performance features:

A fundamental plan, which entails exploring three features and two models, has been established. The subsequent step involves identifying the most effective indicators/features from the extensive array of over 150+ technical and 100+ financial indicators available, as well as determining which algorithm or NLP technique to employ sentiment analysis of financial news. Consequently, a literature survey becomes an astute strategy to comprehend the extent of work already accomplished in this realm and identify existing gaps. Hence, a thorough literature review will be conducted to discern the most effective technical indicators, financial indicators/ratios, and superior algorithms or NLP techniques for sentiment analysis.

2.3.1 Best Technical Indicators for stock price predictions

In the paper "Predicting stock market index using LSTM," authors Bhandari, Rimal, Pokhrel, Rimal, Dahal, & Khatri (2022) forecasted the closing value of the S&P 500 index using technical indicators such as Moving Average Convergence Divergence (MACD), Average True Range (ATR), and Relative Strength Index (RSI). Additionally, macroeconomic indicators like the Cboe Volatility Index (VIX), Interest Rate (EFFR), Civilian Unemployment Rate (UNRATE), Consumer Sentiment Index (UMCSENT), and US dollar index (USDX) were incorporated. Features were selected based on a correlation matrix, and data was refined through Harr wavelet transformation and normalization. Both single-layer and multi-layer LSTM models were deployed, with hyperparameter optimization applied to each. Evaluation metrics used include RMSE, MAPE, and R2. Results indicated that a straightforward single-layer model with approximately 150 hidden neurons yielded better prediction accuracy.

In the paper titled “Prediction of stock price direction using the LASSO-LSTM model combines technical indicators and financial sentiment analysis,” Yang, Wang, & Li (2022) projected the stock prices of MSFT, AAPL, and BAC by utilizing 37 technical indicators from the TTR package in R and conducting sentiment analysis of financial data through the NLP-based FinBERT tool. Six distinct models, namely RF, SVM, LSTM, LSTMTI, LSTMSI, and LASSO-LSTM, were evaluated. Assessment metrics employed included the Kappa coefficient, ROC curve, sensitivity, specificity, and a confusion matrix. A rolling prediction approach, rather than cross-validation, was utilized for performance evaluation. Results demonstrated that the LASSO-LSTM model consistently outperformed the others, achieving peak accuracies of 71.60% for MSFT, 75.60% for AAPL, and 77.20% for BAC. These findings underscore that the LASSO-LSTM model, when fused with combined indicators, transcends the standard LSTM model in predictive accuracy. Additionally, compared to LSTMSI and LSTMTI, the LASSO-LSTM model integrated with indicators showcased an accuracy enhancement of 15.46% and 10.53%, respectively.

Vargas, Anjos, Bichara, and Evsukoff (2018) delve into the realm of stock market prediction in their paper, “Deep Learning for Stock Market Prediction Using Technical Indicators and Financial News Articles.” In this investigative study, deep learning models utilize financial news headlines alongside particular technical indicators to anticipate daily directional movements in stock prices. The research scrutinizes two unique sets of technical indicators: one including Stochastic %K, Stochastic %D, Momentum, Rate of Change, William's %R, Accumulation/Distribution (A/D) oscillator, and Disparity 5, and the other embracing Exponential Moving Average, Moving Average Convergence-Divergence, Relative Strength Index, On Balance Volume, and Bollinger Bands. The implemented deep learning methods showcase the potential to identify complex patterns in the data, thereby enhancing the accuracy of trading strategies. The outcomes suggest that while Convolutional Neural Networks (CNNs) adeptly extract semantic meanings from text, Recurrent Neural Networks (RNNs) prove superior at capturing context and modeling the temporal nuances intrinsic to the stock market. The research juxtaposes two models: the SI-RCNN, a hybrid model employing CNN for interpreting news and LSTM for analyzing technical indicators, and the I-RNN, which exclusively utilizes LSTM for the technical indicators. The stock price series of the Chevron Corporation was selected for experimentation, and word embedding training was facilitated by the Word2vec model. Notably, the SI-RCNN model attained a prediction accuracy of 56.84%. When this model is integrated within a trading agent, factoring in minimal brokerage fees, it amplifies the initial investment by 13.94% during testing. This performance significantly surpasses the conventional buy-and-hold strategy’s return of 3.22%. Additionally, test accuracies were recorded as 52.52% for I-RNN, 48.92% for I-RNN-2, and 51.08% for SI-RCNN-2.

Inference: Based on the discussions in the above three papers, it is evident that certain technical indicators, specifically MACD, ATR, RSI, OBV, NATR, ADXR, TRIX, ADOSC, AD, Stoch_SD, MAX, and STD, exhibit notable performance in the prediction of stock prices.

2.3.2 Best Financial Indicators for stock price prediction

In a pursuit to unravel the relationship between financial ratios and stock price predictions, Arkan (2016), through the paper "The Importance of Financial Ratios in Predicting Stock Price Trends: A Case Study in Emerging Markets," navigated the nuanced dynamics of financial ratios within emerging markets, notably Kuwait. Anchoring the research on a meticulous examination of 12 financial ratios, the study leveraged data from 15 companies distributed over three pivotal sectors, encompassing a period from 2005 to 2014. A multiple regression model was deployed to explore possible relationships and formulate an equation for estimating stock prices, while the STEPWISE method was applied to trim non-contributory variables, enhancing the model's precision. A pronounced positive correlation was observed between certain ratios and stock price trends. For example, within the industrial sector, Return On Assets (ROA), Return On Equity (ROE), and the net profit ratio emerged as significantly influential. Concurrently, in the service sector, the ROA, ROE, Price-to-Earnings (P/E), and Earnings Per Share (EPS) ratios were underscored, a pattern that was similarly reflected in the investment sector. This research underscored the pivotal role that sector-specific financial ratios play in predicting stock prices, affirming that these ratios can be a potent analytical tool for investors and decision-makers in facilitating astute financial and operational considerations.

Inference: Based on the discussions in the above three papers, it is evident that certain technical indicators, specifically Price-to-Earnings (P/E) Ratio, Price-to-Book (P/B) Ratio, Enterprise Value over EBITDA (EV/EBITDA), Return on Assets (ROA), Return on Equity (ROE), Earnings Per Share (EPS), Price/Earnings-to-Growth (PEG) Ratio, Interest Coverage Ratio (ICR), Operating Margin (%), Net Profit Ratio, Debt-to-Capital Ratio, and Quick Ratio, exhibit notable performance in the prediction of stock prices.

2.3.3 Literature Exploration on Algorithms for Sentiment Analysis

A comprehensive review of the literature will be conducted to explore various Natural Language Processing (NLP) techniques utilized in performing sentiment analysis on financial news datasets.

Sentiment Analysis Techniques in General Domains

In the paper “Sentiment Analysis: An Empirical Comparative Study of Various Machine Learning Approaches,” Bandyopadhyay, Sharma, & Sangal (2017) experimented with combinations of Logistic Regression (LR), Naïve Bayes (NB), and Support Vector Machine (SVM) models alongside feature extraction techniques such as Bag of Words - Count Vectorizer, TF-IDF, unigrams, bigrams, and trigrams. They found that LR, when paired with Count Vectorizer and bigrams, yielded a superior accuracy of 82%. It is notable that Bag of Words (BOW) and Term Frequency-Inverse Document Frequency (TF-IDFs) are frequency-based word embeddings that do not capture the contextual representation of words within the corpus.

In contrast, Murti (2020), in the paper titled “Sentiment Analysis in Movie Reviews Using Word2Vec and Long Short-Term Memory (LSTM),” employed an LSTM model paired with Word2Vec or GLoVe, which are pre-trained feature extraction methods. These methods comprehend the contextual representation of words and assist in establishing semantic relationships between words within the corpus, thereby enhancing the accuracy of sentiment analysis, which in this study reached 86.75%. However, it is noteworthy that techniques like Word2Vec or GLoVe still do not capture the exact meaning of a word within its context.

González-Carvajal and Garrido-Merchán (2021), in their paper “Comparing BERT against Traditional Machine Learning Text Classification,” explored various datasets, such as IMDB movie reviews and financial news, utilizing the Bidirectional Encoder Representations from Transformers (BERT) model and contrasting it with several traditional machine learning models, including Logistic Regression (LR), Linear SVC, Multinomial NB, Voting Classifier, Ridge Classifier, and Passive Aggressive Classifier. BERT emerged superior, achieving an accuracy exceeding 90% across all datasets. Transformer models like BERT and DistilBERT possess the capability to comprehend the context of words, leveraging self-attention mechanisms and the model's architecture.

Transitioning to Domain-Specific Sentiment Analysis

From a review of the aforementioned papers, the significance of choosing appropriate models and feature extraction techniques to achieve optimal accuracy in sentiment analysis becomes evident. However, it's crucial to note that widely recognized models like DistilBERT, Word2Vec, and GLoVe are pre-trained on general text datasets, such as Wikipedia and Google News, and might exhibit suboptimal performance when applied to specialized corpora like those composed of financial statements and news due to the distinct terminology used in such texts. In light of this, we turned to the paper “FinBERT: Financial Sentiment Analysis with Pre-trained Language Models” by Araci (2021). In this study, the author implemented various models and feature extraction methods to perform sentiment analysis on prominent financial datasets, namely, the Financial Phrasebank dataset, TRC-2 financials (data from Reuters), and the FiQA sentiment dataset (developed for the WWW ’18 conference's financial opinion mining and question answering challenge). The array of models explored spanned LSTM with GLoVe, ELMo embedding Bi-LSTM, ULMFit, and Transformers, specifically BERT and the specially proposed model for the financial domain - FinBERT. Given the unique challenges presented by financial sentiment analysis, largely due to the industry-specific language, FinBERT, empowered by a pre-trained feature extraction method developed on a variety of financial documents and news, managed to achieve a remarkable accuracy of 97% on the Financial Phrasebank dataset, while, for comparison, LSTM only achieved 81%. These results underscore FinBERT's adeptness at navigating the intricacies of financial sentiment analysis.

Discussion on Large Language Models in Sentiment Analysis

This is an era of Large Language Models - a deep learning algorithm that can perform a variety of natural language processing (NLP) tasks, like ChatGPT-4 and GPT3 from OpenAI, Google’s LaMDA and PaLM LLM (the basis for Bard). Zhang, Deng, Liu, Pan & Bing (2023), in their paper named “Sentiment Analysis in the Era of Large Language Models: A Reality Check” has provided an incisive exploration into the applicability of LLMs like ChatGPT-4 and GPT-3 in various sentiment analysis tasks, ranging from traditional sentiment classification to intricate, multi-faceted analyses of subjective texts. Their research, encapsulating 13 distinct tasks across 26 datasets, highlights critical insights into both the capabilities and limitations of LLMs in sentiment analysis. Notably, while LLMs have showcased strong zero-shot and few-shot prompt strategy performances in simpler tasks securing an accuracy of over 90% in IMDB sentiment classification and even exceeding 95% in the Yelp-2 dataset. Their application in sentiment analysis, particularly in enhancing depth and domain-specific efficacy, remains a complex challenge and constitutes the current cutting edge of research in the field.

Inference: Emerging from the intellectual exploration encapsulated within this section, it becomes apparent that various neural network architectures, notably RNN, LSTM, and Bi-LSTM, when synergized with word embedding techniques such as word2vec and GLoVe, manifest a superlative capacity for discerning and assimilating the contextual essence embedded within word configurations, thereby markedly overshadowing traditional machine learning models that utilize BOW or TF-IDF in the domain of sentiment analysis. Venturing further into the complexities of contextual interpretation, the advent and application of BERT models present a nuanced approach to understanding the intricacies of word contextuality, facilitating a deeper, more syntactically and semantically rich analysis. It is crucial to underscore the paramountcy of the corpus' nature upon which a model is sculpted; this is vividly exemplified by the heightened predictive accuracy observed in sentiment analysis conducted with FinBERT (a model honed on a corpus of financial news statements) as opposed to its generic counterpart, BERT, which is forged from a general Google News corpus. Transcending these methodologies, the deployment of LLM-ChatGPT3.5, judiciously compared with adeptly designed prompts, also exudes proficiency in sentiment analysis, thereby punctuating another fertile ground warranting further academic exploration and investigation.

2.4 Gaps in the Current Literature

While current literature has examined and compared various models such as statistical, traditional machine learning, and neural networks for stock price prediction, a thorough investigation into the effectiveness of different feature combinations seems to be lacking. Typically, most studies have explored the combinations of technical indicators and financial news at most. However, I aspire to delve deeper into scrutinizing the impact of each feature type by gradually incorporating them into the model - initially testing with Technical Indicators (TIs), followed by a combination of TIs and Financial Indicators (FIs), and finally, integrating TIs, FIs, and Financial News. This iterative method aims to explain how the addition of each feature type enhances stock price prediction, a research approach that has not been widely explored.

In the field of sentiment analysis of financial news, previous studies have not utilized Large Language Models (LLMs) like ChatGPT3.5 in sentiment analysis for predicting stock prices. Therefore, I intend to compare ChatGPT3.5 with other sentiment analysis tools such as FinBERT and Bi-directional Long Short-Term Memory (BI-LSTM) with Word2Vec, facilitating a comprehensive comparison and revealing insights into their respective prediction capabilities.

2.5 Summary of the Literature Review

Navigating through the labyrinth of stock price prediction mechanisms, the inferences drawn from various scholarly papers and studies provide a cogent basis for understanding and strategizing the forthcoming analysis.

As elucidated in the papers reviewed in section 2.2, a commendable analysis was conducted on historical stock price data of companies by employing both traditionally derived and innovative features from fundamental prices alongside various technical indicators. Intriguingly, it was the Random Forest (RF), Artificial Neural Network (ANN), and Long Short-Term Memory (LSTM) models that emerged as significantly efficacious, with the LSTM model, in particular, exhibiting exemplary performance relative to other models. The scholarly explorations probed into various facets of financial data, offering nuanced perspectives on the palpable influence of financial news and social media sentiment on stock prices, while maintaining a modicum of ambivalence regarding the tangible impact of financial statements.

My Plan: Anchored by these insights, my subsequent analysis will harness the robust capabilities of both the RF regressor and the LSTM models, given their proven proficiency in traditional machine learning and advanced neural networks respectively. Given the discernible impact of technical indicators and sentiment analysis on stock price prediction and the ongoing ambiguity surrounding the impact of financial statements, my plan is to judiciously amalgamate all three features within my LSTM model, while also conducting a singular analysis utilizing the RF regressor to establish a substantive baseline for deploying the LSTM model.

The enthralling journey through the scholarly landscapes described in section 2.3 illuminated the prevailing wisdom on the pivotal role of technical and financial indicators in stock price prediction. Distilled from the intellectual discourses, technical indicators such as MACD, ATR, RSI, and others, alongside financial indicators like Price-to-Earnings (P/E) Ratio, Return on Assets (ROA), and more, have all been identified as powerful arbiters of predictive performance in the financial domain.

In the intriguing realm of sentiment analysis, the scholarly artifacts spotlight the laudable capacities of neural network architectures, especially when enriched with word embedding techniques. Further depth is ventured into with the advent of BERT models, revealing the nuanced capacities of these models to weave through the intricacies of contextual word understanding, exemplified with specialized versions such as FinBERT which is sculpted with a financial news corpus and demonstrably outperforms its generalist counterpart.

My Plan: Moving forward, my analysis will assimilate the identified technical and financial indicators, ensuring a thorough incorporation of these potent variables into the predictive model. Meanwhile, on the sentiment analysis frontier, my approach will be tri-pronged, exploring the efficacies of three distinct algorithms - a general domain model (Bi-LSTM with word2vec), a finance-focused model (FinBERT), and the avant-garde LLM. This multifaceted approach, born from the nature of our data source - the New York Times, and the subsequent uncertainty regarding which model will exhibit superior contextual understanding and predictive accuracy in this sphere, promises a thorough exploration and comprehension of their respective capacities, thus ensuring a comprehensive and robust analysis.

3. Methods

3.1 Introduction / Overview

In this section, we will explain the methodology what we followed in executing the project.

The general methodology for the stock price prediction using LSTM model and sentiment analysis of financial news data using NLP models are as under,

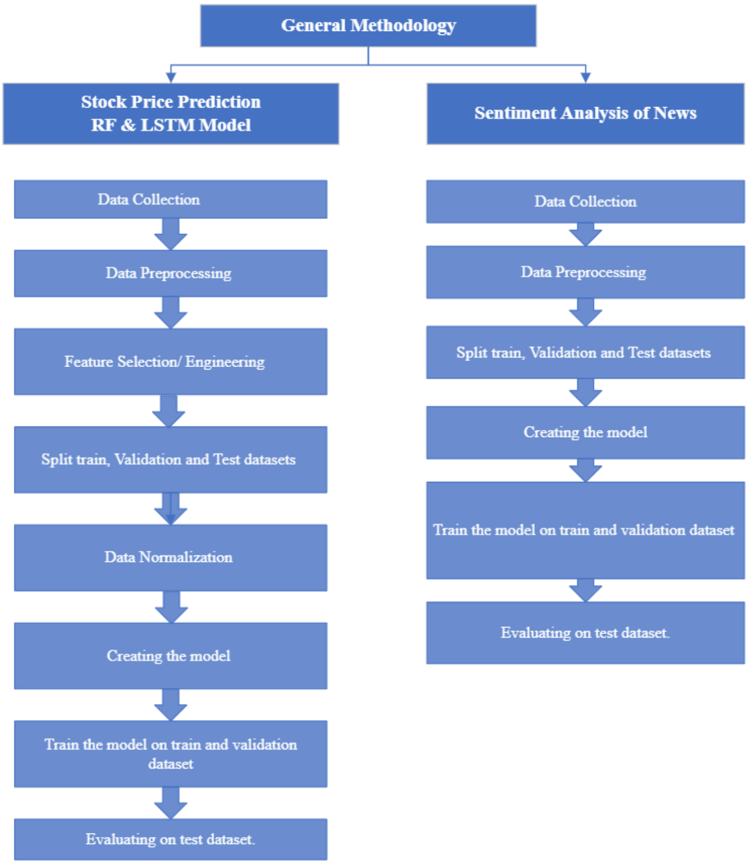

General Methodology
Stock Price Prediction RF & LSTM Model
Sentiment Analysis of News
Data Collection
Data Preprocessing
Feature Selection/ Engineering
Split train, Validation and Test datasets
Data Normalization
Creating the model
Train the model on train and validation dataset
Evaluating on test dataset.
Data Collection
Data Preprocessing
Split train, Validation and Test datasets
Creating the model
Train the model on train and validation dataset
Evaluating on test dataset.

Figure 3: General Methodology

Here is the workflow of the experiments of this project, The codes were also structured in the same structure.

Main Task

Created Dataset_A - Historical stock price + Technical Indicators

- Experimented it with RF regressor → Output_1
- Experimented it with LSTM Model → Output_2
- Experimented it with Hyperparameter Optimized LSTM Model → Output_3

Created Dataset_B - Historical stock price + Technical Indicators + Financial Indicators/Ratios

- Experimented it with LSTM Model → Output_4
- Experimented it with Hyperparameter Optimized LSTM Model → Output_5

Created Dataset_C - Historical stock price + TI + FI + Financial News NYT (word2vec)

- Experimented it with LSTM Model → Output_6
- Experimented it with Hyperparameter Optimized LSTM Model → Output_7

Created Dataset_D1 - Historical stock price + TI + Financial News NYT (word2vec)

- Experimented it with LSTM Model → Output_8
- Experimented it with Hyperparameter Optimized LSTM Model → Output_9

Created Dataset_D2 - Historical stock price + TI + Financial News NYT (FinBERT)

- Experimented it with LSTM Model → Output_10
- Experimented it with Hyperparameter Optimized LSTM Model → Output_11

Created Dataset_D3 - Historical stock price + TI + Financial News NYT (LLM-ChatGPT3.5)

- Experimented it with LSTM Model → Output_12
- Experimented it with Hyperparameter Optimized LSTM Model → Output_13

Sub Task

Here comes the sub task of sentiment analysis of financial news from New York Times.

Downloaded financial_phrasebank dataset - labelled with sentiments

- Senti Analysis 1 - Bi-LSTM model with word2vec on financial_phrasebank dataset
- Senti Analysis 2 - FinBERT on financial_phrasebank dataset
- Senti Analysis 3 - LLM on financial_phrasebank dataset

Extracted 10 years of Apple News from New York Times – unlabelled

- Senti Analysis 1.1 - Bi-LSTM model with word2vec on NewYork Times dataset
- Senti Analysis 2.1 - FinBERT on NewYork Times dataset
- Senti Analysis 3.1 - LLM on NewYork Times dataset

In the below figure both the main task and sub task work flow is furnished in detail.

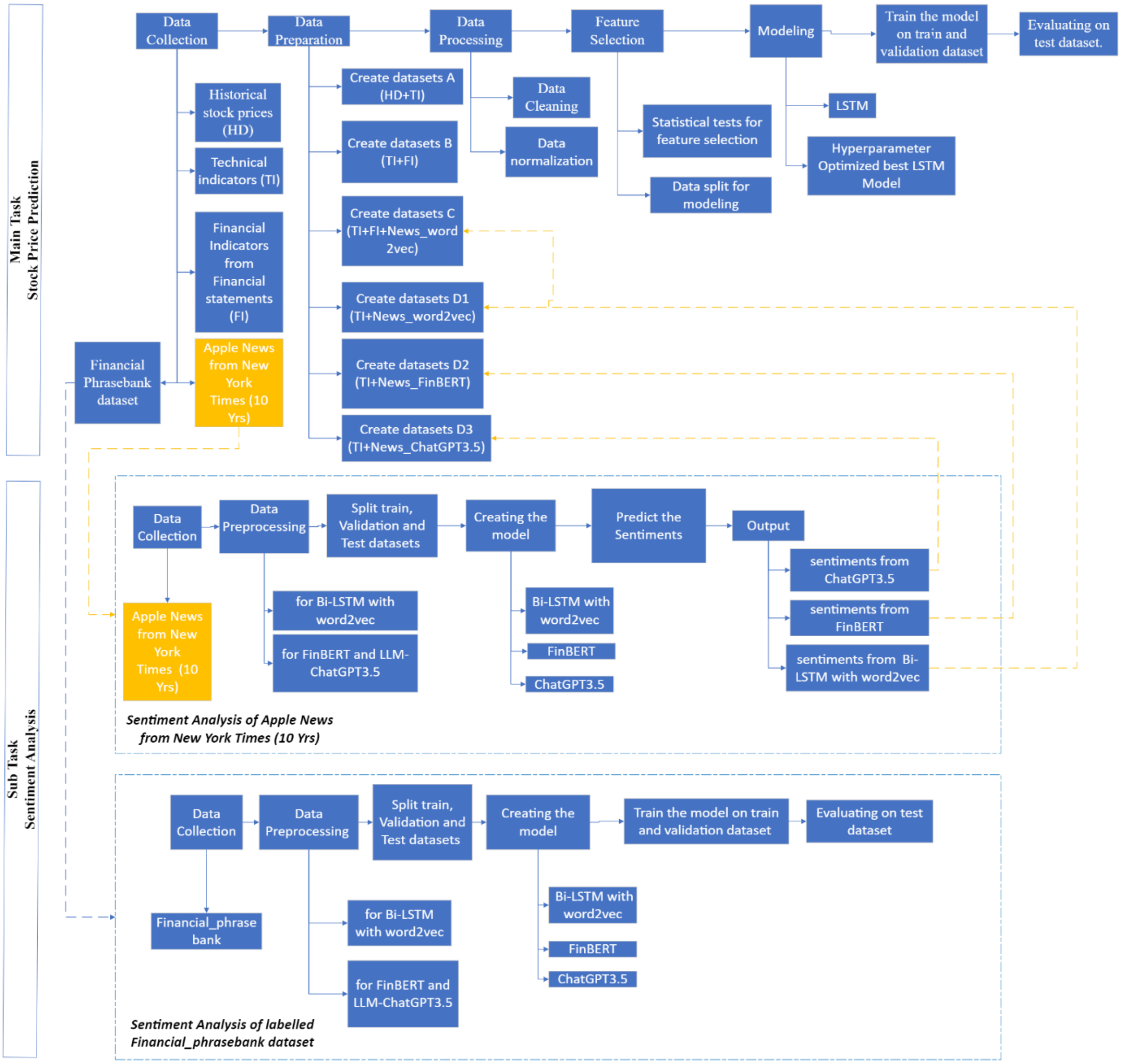

Main Task
Stock Price Prediction
Data Collection
Data Preparation
Data Processing
Feature Selection
Modeling
Train the model on train and validation dataset
Evaluating on test dataset.
Historical stock prices (HD)
Technical indicators (TI)
Financial Indicators from Financial statements (FI)
Apple News from New York Times (10 Yrs)
Financial Phrasebank dataset
Create datasets A (HD+TI)
Create datasets B (TI+FI)
Create datasets C (TI+FI+News_word 2vec)
Create datasets D1 (TI+News_word2vec)
Create datasets D2 (TI+News_FinBERT)
Create datasets D3 (TI+News_ChatGPT3.5)
Data Cleaning
Data normalization
Statistical tests for feature selection
Data split for modeling
LSTM
Hyperparameter Optimized best LSTM Model
Sub Task
Sentiment Analysis
Data Collection
Data Preprocessing
Split train, Validation and Test datasets
Creating the model
Predict the Sentiments
Output
Apple News from New York Times (10 Yrs)
for Bi-LSTM with word2vec
for FinBERT and LLM-ChatGPT3.5
Bi-LSTM with word2vec
FinBERT
ChatGPT3.5
sentiments from ChatGPT3.5
sentiments from FinBERT
sentiments from Bi-LSTM with word2vec
Sentiment Analysis of Apple News from New York Times (10 Yrs)
Data Collection
Data Preprocessing
Split train, Validation and Test datasets
Creating the model
Train the model on train and validation dataset
Evaluating on test dataset
Financial_phrase bank
for Bi-LSTM with word2vec
for FinBERT and LLM-ChatGPT3.5
Bi-LSTM with word2vec
FinBERT
ChatGPT3.5
Sentiment Analysis of labelled Financial_phrasebank dataset

Figure 4: Overview of Methodological Approach

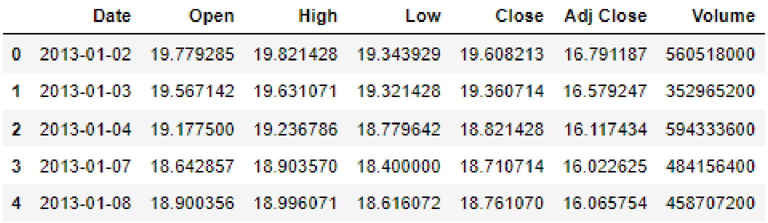
3.2 Data collection and pre-processing

3.2.1 Data Collection, Sources Identification, and nature of data

Fundamental Historical Stock Price Data of Apple Inc.

The foundation of our empirical analysis lies in the Historical Stock Price Data of Apple Inc., extracted for the last 10+ years (i.e., from 01/01/2013 to 01/08/2023), serving as a pivotal data point in our research. This data is garnered through the utilization of the 'yfinance' library, a potent Python library that interfaces with Yahoo Finance, efficiently allowing data retrieval through the identification of the NASDAQ ticker symbol, AAPL.

Nature of the dataset: This dataset contains 2662 instances with 7 variables namely one date variable and 6 numeric variables namely - Date, Open, High, Low, Close, Adj Close, and Volume. Here is the snippet of the dataset as under,



	Date	Open	High	Low	Close	Adj Close	Volume
0	2013-01-02	19.779285	19.821428	19.343929	19.608213	16.791187	560518000
1	2013-01-03	19.567142	19.631071	19.321428	19.360714	16.579247	352965200
2	2013-01-04	19.177500	19.236786	18.779642	18.821428	16.117434	594333600
3	2013-01-07	18.642857	18.903570	18.400000	18.710714	16.022625	484156400
4	2013-01-08	18.900356	18.996071	18.616072	18.761070	16.065754	458707200

Figure 5: Historical stock price data

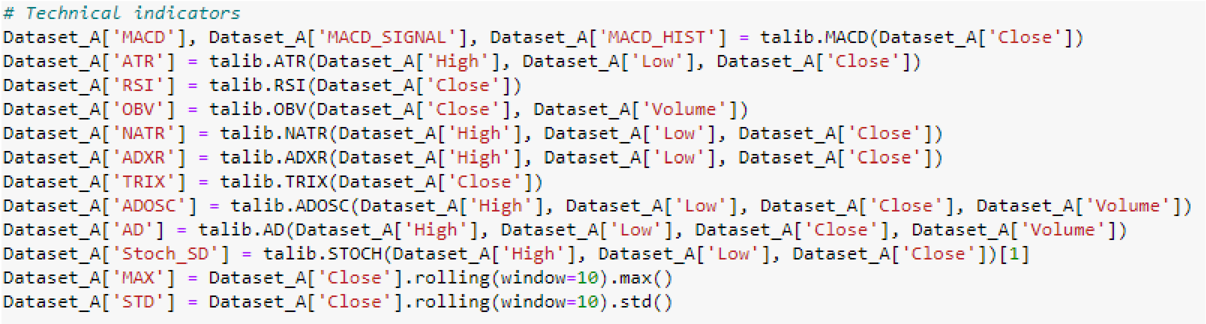
The choice of utilizing Yahoo Finance as the data source pivots on its esteemed reputation and widespread use among financial analysts and researchers. It is acclaimed for providing accurate and reliable historical data, thus ensuring the integrity and robustness of our dataset.

Note: I understood that for linear time series models like ARIMA or SARIMA, transforming non-stationary target variables (like closing prices) to stationary data is crucial because these models assume the underlying statistical properties (mean, variance, and autocorrelation) of the data remain constant over time. Non-stationary data, where these properties shift, can lead to unreliable predictions and misinterpretations of modelled relationships, thereby compromising the validity of the model's forecasts and inferential capabilities.

Whereas, Random Forest Regressor, being a non-linear model, does not obligate data to be stationary, as it does not hinge its predictions on linear relationships or constant statistical properties, instead, learning patterns through a decision-tree based approach.

<u>Technical Indicators data:</u>

Serving as critical metrics in our analytical arsenal, are systematically derived from fundamental prices such as open, minimum, maximum, close, and volume. The technical indicators can either be executed through formulas for each indicator or, alternatively, leveraged through a proficient library named TA-Lib. This library, equipped with more than 150 technical indicators, facilitates effortless extraction of these indicators by merely inputting the fundamental prices. Below is a code snippet embracing all the technical indicators that have been meticulously selected based on the literature survey 2.3.1 – Best Technical Indicators for stock price prediction.



```
# Technical indicators
Dataset_A['MACD'], Dataset_A['MACD_SIGNAL'], Dataset_A['MACD_HIST'] = talib.MACD(Dataset_A['Close'])
Dataset_A['ATR'] = talib.ATR(Dataset_A['High'], Dataset_A['Low'], Dataset_A['Close'])
Dataset_A['RSI'] = talib.RSI(Dataset_A['Close'])
Dataset_A['OBV'] = talib.OBV(Dataset_A['Close'], Dataset_A['Volume'])
Dataset_A['NATR'] = talib.NATR(Dataset_A['High'], Dataset_A['Low'], Dataset_A['Close'])
Dataset_A['ADXR'] = talib.ADXR(Dataset_A['High'], Dataset_A['Low'], Dataset_A['Close'])
Dataset_A['TRIX'] = talib.TRIX(Dataset_A['Close'])
Dataset_A['ADOSC'] = talib.ADOSC(Dataset_A['High'], Dataset_A['Low'], Dataset_A['Close'], Dataset_A['Volume'])
Dataset_A['AD'] = talib.AD(Dataset_A['High'], Dataset_A['Low'], Dataset_A['Close'], Dataset_A['Volume'])
Dataset_A['Stoch_SD'] = talib.STOCH(Dataset_A['High'], Dataset_A['Low'], Dataset_A['Close'])[1]
Dataset_A['MAX'] = Dataset_A['Close'].rolling(window=10).max()
Dataset_A['STD'] = Dataset_A['Close'].rolling(window=10).std()
```

Figure 6: Technical Indicators

Websites like Investopedia and MetaStock, along with John J & Murphy's (1999) book 'Technical Analysis of the Financial Markets', offered clear explanations about the technical indicators and I have explained about the 12 technical indicators in two sentences as under,

- MACD (Moving Average Convergence Divergence): MACD = (12-day EMA - 26-day EMA), a momentum oscillator that signals trend direction by identifying the relationship between two moving averages, helping to pinpoint potential buy and sell signals.
- ATR (Average True Range): ATR = (1/n) * (Sum of TR for n periods), reflecting market volatility by averaging the true range (TR), calculated as the maximum of [(High - Low), abs(High - Close(previous)), abs(Low - Close(previous))], over a specific period (n).
- RSI (Relative Strength Index): RSI = 100 - (100 / (1 + RS)), where RS (Relative Strength) = (Average Gain over n periods) / (Average Loss over n periods), providing insights into the magnitude of recent price changes to evaluate overbought or oversold conditions.
- OBV (On Balance Volume): OBV = Previous OBV + Current Trading Volume (if closing price is higher) or OBV = Previous OBV - Current Trading Volume (if closing price is lower), used to predict price movements through cumulative volume analysis.
- NATR (Normalized Average True Range): NATR = (ATR / Close) * 100, offering a perspective on volatility related to the closing price, facilitating comparisons between different securities or time periods.

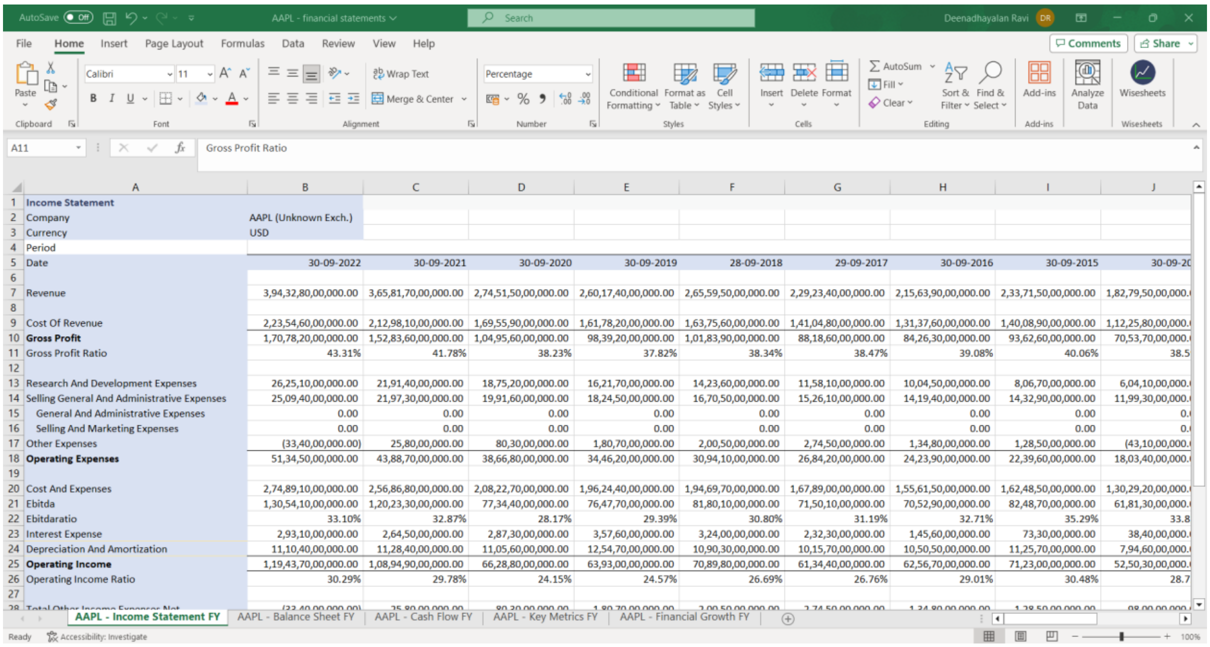
- ADXR (Average Directional Movement Index Rating): ADXR = (ADX(current period) + ADX(n periods back)) / 2, which averages the ADX values over a set period to determine the strength of the trend over time.
- TRIX (Triple Exponential Moving Average): TRIX = EMA(EMA(EMA(Price, n), n), n), which filters price noise by focusing on the inherent trends of the triple EMA, signifying reversals and directional shifts.
- ADOSC (Chaikin A/D Oscillator): ADOSC = [(3-day EMA of ADL) - (10-day EMA of ADL)], where ADL (Accumulation/Distribution Line) analyzes the distribution and accumulation phases by comparing close, high, and low prices.
- AD (Accumulation/Distribution Index): AD = [(Close - Low) - (High - Close)] / (High - Low) * Volume + Previous AD, interpreting the flow of money by correlating volume to price changes.
- Stoch_SD (Stochastic Oscillator %D): Stoch_SD = (100 * (Close - Low(n))) / (High(n) - Low(n)), where n represents the look-back period, assists in identifying overbought and oversold zones by gauging the price position relative to its range over a defined timeframe.
- MAX (Maximum Value): MAX is a simple calculation identifying the highest value of a particular data set (such as price) over a specified number of periods, providing an insight into peak values.
- STD (Standard Deviation): STD = sqrt[(Σ(xi - μ)^2) / n], a measure of the amount of variation or dispersion of a set of values, serving as a key volatility indicator in the context of stock prices, where xi represents each value in the dataset, μ is the mean, and n is the number of observations.

Nature of the dataset: This dataset contains 2662 instances with 15 variables namely one date variable and 14 numeric variables i.e. technical indicators arrived as mentioned above.

Financial Indicators and Ratios data:

These are extracted from the financial statements that every company releases annually and quarterly. For this analysis, I sourced the annual statements and derived pertinent financial indicators from them. Typically, a financial statement encompasses the Income Statement, Balance Sheet, and Cash Flow Sheet.

At first, our team utilized Yahoo Finance for data, but it only offered the last four years of financial statements. We then discovered Macrotrends, a site that compiled a decade of financial indicators, yet, regrettably, their web protection mechanisms prevented data extraction via web-scraping. Macrotrends sources its data from the U.S. Securities and Exchange Commission (SEC), a federal agency that oversees the enforcement of federal securities laws and regulates related organizations, including the securities industry and stock exchanges. SEC offers abundant data through companies' 10-K filings (annual reports), thus providing a thorough look into their financial statuses. We began web-scraping from SEC, an open-source platform, but fortuitously found Wise_Excel, another website where we could consolidate and download a decade's worth of financial statements, as shown in the Excel snippet below,



A11 | Gross Profit Ratio

	A	B	C	D	E	F	G	H	I	J
1	Income Statement									
2	Company	AAPL (Unknown Exch.)								
3	Currency	USD								
4	Period									
5	Date	30-09-2022	30-09-2021	30-09-2020	30-09-2019	28-09-2018	29-09-2017	30-09-2016	30-09-2015	30-09-20
6										
7	Revenue	3,94,32,80,00,000.00	3,65,81,70,00,000.00	2,74,51,50,00,000.00	2,60,17,40,00,000.00	2,65,59,50,00,000.00	2,29,23,40,00,000.00	2,15,63,90,00,000.00	2,33,71,50,00,000.00	1,82,79,50,00,000.
8										
9	Cost Of Revenue	2,23,54,60,00,000.00	2,12,98,10,00,000.00	1,69,55,90,00,000.00	1,61,78,20,00,000.00	1,63,75,60,00,000.00	1,41,04,80,00,000.00	1,31,37,60,00,000.00	1,40,08,90,00,000.00	1,12,25,80,00,000.
10	**Gross Profit**	1,70,78,20,00,000.00	1,52,83,60,00,000.00	1,04,95,60,00,000.00	98,39,20,00,000.00	1,01,83,90,00,000.00	88,18,60,00,000.00	84,26,30,00,000.00	93,62,60,00,000.00	70,53,70,00,000.
11	Gross Profit Ratio	43.31%	41.78%	38.23%	37.82%	38.34%	38.47%	39.08%	40.06%	38.5
12										
13	Research And Development Expenses	26,25,10,00,000.00	21,91,40,00,000.00	18,75,20,00,000.00	16,21,70,00,000.00	14,23,60,00,000.00	11,58,10,00,000.00	10,04,50,00,000.00	8,06,70,00,000.00	6,04,10,00,000.
14	Selling General And Administrative Expenses	25,09,40,00,000.00	21,97,30,00,000.00	19,91,60,00,000.00	18,24,50,00,000.00	16,70,50,00,000.00	15,26,10,00,000.00	14,19,40,00,000.00	14,32,90,00,000.00	11,99,30,00,000.
15	General And Administrative Expenses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.
16	Selling And Marketing Expenses	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.
17	Other Expenses	(33,40,00,000.00)	25,80,00,000.00	80,30,00,000.00	1,80,70,00,000.00	2,00,50,00,000.00	2,74,50,00,000.00	1,34,80,00,000.00	1,28,50,00,000.00	(43,10,00,000.
18	**Operating Expenses**	51,34,50,00,000.00	43,88,70,00,000.00	38,66,80,00,000.00	34,46,20,00,000.00	30,94,10,00,000.00	26,84,20,00,000.00	24,23,90,00,000.00	22,39,60,00,000.00	18,03,40,00,000.
19										
20	Cost And Expenses	2,74,89,10,00,000.00	2,56,86,80,00,000.00	2,08,22,70,00,000.00	1,96,24,40,00,000.00	1,94,69,70,00,000.00	1,67,89,00,00,000.00	1,55,61,50,00,000.00	1,62,48,50,00,000.00	1,30,29,20,00,000.
21	Ebitda	1,30,54,10,00,000.00	1,20,23,30,00,000.00	77,34,40,00,000.00	76,47,70,00,000.00	81,80,10,00,000.00	71,50,10,00,000.00	70,52,90,00,000.00	82,48,70,00,000.00	61,81,30,00,000.
22	Ebitdaratio	33.10%	32.87%	28.17%	29.39%	30.80%	31.19%	32.71%	35.29%	33.8
23	Interest Expense	2,93,10,00,000.00	2,64,50,00,000.00	2,87,30,00,000.00	3,57,60,00,000.00	3,24,00,00,000.00	2,32,30,00,000.00	1,45,60,00,000.00	73,30,00,000.00	38,40,00,000.
24	Depreciation And Amortization	11,10,40,00,000.00	11,28,40,00,000.00	11,05,60,00,000.00	12,54,70,00,000.00	10,90,30,00,000.00	10,15,70,00,000.00	10,50,50,00,000.00	11,25,70,00,000.00	7,94,60,00,000.
25	**Operating Income**	1,19,43,70,00,000.00	1,08,94,90,00,000.00	66,28,80,00,000.00	63,93,00,00,000.00	70,89,80,00,000.00	61,34,40,00,000.00	62,56,70,00,000.00	71,23,00,00,000.00	52,50,30,00,000.
26	Operating Income Ratio	30.29%	29.78%	24.15%	24.57%	26.69%	26.76%	29.01%	30.48%	28.7
27										
28	Total Other Income Expenses Net	(33,40,00,000.00)	25,80,00,000.00	80,30,00,000.00	1,80,70,00,000.00	2,00,50,00,000.00	2,74,50,00,000.00	1,34,80,00,000.00	1,28,50,00,000.00	98,00,00,000.

Figure 7: Financial Statements

In accordance with our exploration in Section 2.3.2, 'Optimal Financial Indicators for Stock Price Prediction,' we elected 12 financial indicators. Directly extracted from the financial statements are the Price-to-Earnings (P/E) Ratio, Price-to-Book (P/B) Ratio, Enterprise Value over EBITDA (EV/EBITDA), Return on Assets (ROA), Return on Equity (ROE), and Earnings Per Share (EPS). In contrast, the Price/Earnings-to-Growth (PEG) Ratio, Interest Coverage Ratio (ICR), Operating Margin (%), Net Profit Ratio, Debt-to-Capital Ratio, and Quick Ratio necessitate derivation through formulas, involving two or three components from the financial statements. Our comprehension of these financial indicators was enriched by resources like Investopedia and a book name “Financial Statement Analysis: A Practitioner’s Guide (Wiley Finance)” by Fridson & Alvarez (2011) providing insightful delineations of each financial indicator, which are briefly explained as follows,

- Price-to-Earnings (P/E) Ratio: P/E Ratio equals the market value per share divided by earnings per share. It signifies how much investors are willing to pay for each dollar of earnings.
  P/E = Market Value Per Share / Earnings Per Share
- Price-to-Book (P/B) Ratio: The P/B Ratio is calculated by dividing the current price per share by the book value per share. This ratio provides insight into how much shareholders are paying for the net assets of the company.
  P/B = Market Price Per Share / Book Value Per Share
- Enterprise Value over EBITDA (EV/EBITDA): EV/EBITDA is computed by dividing the enterprise value (EV) of a company by its earnings before interest, taxes, depreciation, and amortization (EBITDA). It evaluates a company's value, including debt and excluding non-operating expenses.
  EV/EBITDA=Enterprise Value / EBITDA
- Return on Assets (ROA): ROA is the net income divided by total assets, presenting how effectively a company generates profit from its assets.
  ROA = Net Income / Total Assets
- Return on Equity (ROE): ROE is calculated as net income divided by shareholder’s equity, illustrating the efficiency in generating profits from shareholders’ investments.
  ROE = Net Income / Shareholder′s Equity
- Earnings Per Share (EPS): EPS equals net income minus dividends on preferred stock, all divided by the average outstanding shares, indicating the company’s profitability per shareholder.
  EPS = (Net Income – Dividends on Preferred Stock) / Average Outstanding Shares
- Price/Earnings-to-Growth (PEG) Ratio: PEG Ratio is the P/E Ratio divided by the annual EPS growth, used to determine a stock's value while considering the company's earnings growth.
  PEG = P/E Ratio / Annual EPS Growth

- Interest Coverage Ratio (ICR): ICR is determined by dividing EBIT by interest expense, measuring a company's ability to cover its interest payments with its profits.

  ICR = EBIT / Interest Expense
- Operating Margin (%): Operating Margin is obtained by dividing operating income by net sales and is expressed as a percentage, representing the efficiency of a company in controlling costs.

  Operating Margin (%) = (Operating Income / Net Sales) × 100
- Net Profit Ratio: Net Profit Ratio equals net profit after taxes divided by net sales, providing a glimpse into the profitability of a company after all expenses.

  Net Profit Ratio = Net Profit After Taxes / Net Sales
- Debt-to-Capital Ratio: This ratio is calculated by dividing a company’s total debt by its total capital, providing insight into a company’s financial leverage.

  Debt−to−Capital Ratio = Total Debt / Total Capital
- Quick Ratio: Quick Ratio, or acid-test ratio, is determined by subtracting inventories from current assets and then dividing by current liabilities, illustrating a company's ability to meet its short-term obligations.

  Quick Ratio = (Current Assets – Inventories) / Current Liabilities

Nature of the dataset: This dataset contains 10 instances with 13 variables namely one date variable and 12 numeric variables i.e. financial indicators arrived as mentioned above.

Apple News Data:

In the pursuit of comprehensive sentiment analysis regarding Apple's financial news, an exhaustive endeavor was undertaken to extract pertinent data from various financial news channels and websites, employing methods of web scraping and API utilization. A week was devoted to this task, exploring platforms like Nasdaq and FinViz, which respectively offered limited data of one and two years, and assessing News APIs such as Google News, Yahoo News, Bloomberg, Bing News, and the Financial Post, only to encounter barriers in the form of financial charges or constraints on data duration. The eventual solution was found in the New York Times' Article Search API, where we can search news with a key word, in our case apple related news, and it has its own set of limitations, such as ten news articles per request and a daily restriction of 500 requests, further constrained by a limitation on the number of requests per minute. So, codes were devised to meticulously collect 10 news for 10 days in one request and like that looped to collect one year's news in one csv and like that we did for 10 years. From the JSON response we have secured the following items for our analysis 'Pub Date', 'Headline', 'Snippet', 'Lead Paragraph', 'Abstract', 'Source', and 'Web URL'. Ten years of data was thereby stored into a CSV for subsequent sentiment analysis, with the code encapsulated in a separate Jupyter notebook, named 'NYT News Extraction'.

Nature of the dataset: This dataset contains 3981 instances with 3 variables namely one date variable and 6 other variables namely - Headline, Abstract. Since sometimes the headline are not enough informative to decide on the sentiment, we have included abstract also. And about the rows, there are multiple news on the same day so for 10 years we arrived 3981 instances.

Note: In order to get the sentiments of the NYTimes news about Apple company, this Apple news dataset (3981 x 3) will go through a sub task of sentiment analysis using three sophisticated NLP techniques mentioned in the section 3.5. The output sentiments from the NLP models are with 3981 instances only, after that in order to merge this with the other datasets we have undergone some customised post processing (section 3.2.4) to group it on date and chosen a dominant sentiment for the days with multiple sentiments. Finally, from each of the NLP models with some subtask performed to select the dominant sentiment for multiple news days, we stored the output in pickle to use in the main task dataset preparation. The out is in the format of 1474 instances with 2 variables namely date and dominant_sentiment – dominant sentiment of the day.

Financial_phrasebank data:

The financial_phrasebank dataset is a specifically made for sentiment analysis in the financial domain. This solely focuses on the financial news statements. The dataset is available in the huggingface website. This is made by Malo, Sinha, Korhonen, Wallenius & Takala (2014) named "Good debt or bad debt: Detecting semantic orientations in economic texts". The dataset is annotated by 16 people with adequate background knowledge in financial segment. The dataset consists of 4846 sentences totally which is agreed by 50% and above annotators, 4217 sentences were agreed by 66% and above annotators, 3453 sentences were agreed by 75% and above annotators and finally 2264 sentences were agreed by 100% annotators, and we have chosen the sentence agreed by 100% of the annotators. So, now our dataset consists of 2264 instance with 2 variables namely sentence and label.

3.2.2 Dataset preparation (A, B, C, D1, D2, D3)

With the above four different input data we will create six datasets which will be experimented in the models to answer the research questions. The six datasets are as under,

Preparation of Dataset_A: In this dataset the fundamental historical prices dataset (2662 x 7) will be merged with technical indicators dataset (2662 x 15), so now the having date as the common point we have 2662 rows and 21 variables named Dataset_A.

Preparation of Dataset_B: In this dataset the fundamental historical stock prices are merged with technical indicators and additionally financial indicators / Ratios i.e., Dataset_A (2662 x 21) + Financial Indicator (10 x 13) and this is done by having date as the common point. So, this dataset is now having 2662 rows and 33 variables. Since we are considering annual financial statements, we have only one instance for a year at September end. In filling the NaNs of the financial indicators, I have tried ‘0’s and then a method called ‘ffill’.

Preparation of Dataset_C: In this dataset the fundamental stock price data is merged with all other three features i.e., technical indicators, financial indicators, and sentiments from sentiment analysis of NYTimes dataset by Bi-LSTM with word2vec model i.e., Dataset_B (2662 x 33) + sentiments from Bi-LSTM-word2vec (1474 x 2) and this is merged by having date as the common point. So, this dataset is now having 2662 instances with 34 variables.

Preparation of Dataset_D1: In this dataset the fundamental stock price data is merged with sentiments from sentiment analysis performed on NYTimes dataset by Bi-LSTM with word2vec model i.e., Dataset_A (2662 x 21) + sentiments from Bi-LSTM-word2vec (1474 x 2) and this is merged by having date as the common point. So, this dataset is now having 2662 instances with 34 variables. I got NaNs on the Dominanat sentiment column which are from the non-news days, I have used forward filled method to deal this NaNs. My reason to use forward fill here is if there is a positive/negative news in the market the stock prices reacts until it gets another news.

Preparation of Dataset_D2: In this dataset the fundamental stock price data is merged with sentiments from sentiment analysis performed on NYTimes dataset by FinBERT model i.e., Dataset_A (2662 x 21) + sentiments from FinBERT (1474 x 2) and this is merged by having date as the common point. So, this dataset is now having 2662 instances with 34 variables. I got NaNs on the Dominanat sentiment column which are from the non-news days, I have used forward filled method to deal this NaNs.

Preparation of Dataset_D3: In this dataset the fundamental stock price data is merged with sentiments from sentiment analysis performed on NYTimes dataset by LLM – ChatGPT3.5 model i.e., Dataset_A (2662 x 21) + sentiments from LLM - ChatGPT3.5 model (1474 x 2) and this is merged by having date as the common point. So, this dataset is now having 2662 instances with 34 variables. I got NaNs on the Dominanat sentiment column which are from the non-news days, I have used forward filled method to deal this NaNs.

Now coming to the point why have we created these many datasets? How is it related to the research questions?

In order to answer the RQ2 - How does the incremental addition of varied features (technical indicators, financial indicators/ratios, and sentiment from financial news) influence the accuracy and reliability of stock price predictions across different datasets? I have created dataset A, B and C where I have added one feature by one feature. By running these datasets into the LSTM models we can compare the results and understand the stock price predictive capacity of each feature.

Next, in order to answer the first phase of RQ 3 – which of the NLP model is best in predicting the sentiments of the news? Since the NYTimes dataset is freshly extracted from the website and its unlabelled, we pulled a famous financial news dataset i.e., financial_phraseback dataset – which is labelled by 16 annotator. Trained and tested all these three NLP models into this dataset and compared the results to get the best NLP model to predict the sentiments of the news.

The second phase of RQ-3, how does their performance translate when applied to unlabeled data from different sources (like the NewYork Times dataset)? This is not comparable so what we thought of comparing the sentiments of NLP models indirectly, means took the sentiments of all three NLP models and created dataset D1, D2, and D3 and predicted the stock price and compare the results and find which NLP model has better prediction of stock price. An indirect comparison.

Next, in order to answer RQ4 – Which of the NLP model has a better prediction in stock prices? I have created dataset D1, D2, D3 with sentiments of NYTimes news data from three NLP models. By running these datasets D1, D2, D3 into the LSTM model we can compare the results to know which NLP technique has performed good in predicting the stock prices.

Reason behind the formation of D1, D2 and D3:

If you have the best NLP model in predicting the sentiments of the news data (first phase of RQ3), then why we have taken sentiments from all the NLP models and created D1, D2 and D3 dataset? The answer in the RQ3 is best NLP model in financial_phrasebank dataset, and we have performed sentiment analysis on the NYTimes dataset and got the results but can't be validated and also sentiment predictions are sensitive to the nature of the corpus, we don't the nature of the statements in the NYTimes dataset, why I am saying this is, if it's a general news then word2vec might perform better because word2vec was trained on google news, if NYTimes statements are financial then FinBERT might do better. And LLM can handle both in a better way. So that is how the RQ4 came, what's the performance all the NLP models in predicting the stock price and there by the respective datasets were created D1, D2, and D3.

3.2.3 Data Cleaning and Pre-processing Procedures for Sentiment Analysis

Here first we see about the data preprocessing for the sub task – sentiment analysis part and then to the main task – stock price prediction.

3.2.3.1 for Bi-LSTM model with word2vec:

For both the datasets of NLP model, financial_phrasebank dataset and the NYTimes dataset we followed the below cleaning and pre-processing steps,

Step1: Pull the text and the label as x and y, then split the data into 80% for train, 10% for validation and 10% for test in case of financial_phrase bank dataset, in NYTimes its just text we don't split because we don't have labels to validate.

Step 2: Preprocessing of text using regular expression and NLTK packages consists of the following steps,

- Separation of contractions like "ain't": "am not", "can't": "cannot", etc.

- removal of URLs, HTML tags, non-alphabetic characters.

- Convert to lower case

- tokenize the text

- remove stop words

- lemmatizing the tokens – converting all the words to their base form like, converting walking, walked, walked to walk.

- Have not done stemming – which removes the last few characters of a word to bring it down to its base form, eg: jumping to jump. But it also removes ing from caring and make it as car where the meaning of the word changes. So, it always better to avoid stemming and go for lemmatizing.

Step 3: tokenizing, indexing and sequencing the pre-processed text – in this step each and every word of the corpus becomes a token, and the tokens will be matched with unique number. The vocabulary size kept is 5000 for the financial_phrasebank dataset because there are 4918 unique token in that corpus and for NYTimes model kept 8000 because it has 7658 unique token in it.

Step 4: padded sequencing - fix a pad_len according to the average length of the sentence and create a padded sequence, so that all the input statements to the neural networks will be in the same length, which is a requirement from the model. Fixed 50 for financial_phrasebank dataset and 100 for the NYTimes dataset.

Step 5: word embedding - download the pre-trained word2vec vectorizer for word embedding ‘word2vec-google-news-300' this has 300 embedding dimensions in it. Create a dictoniary for the word embedding and then create the embedding matrix.

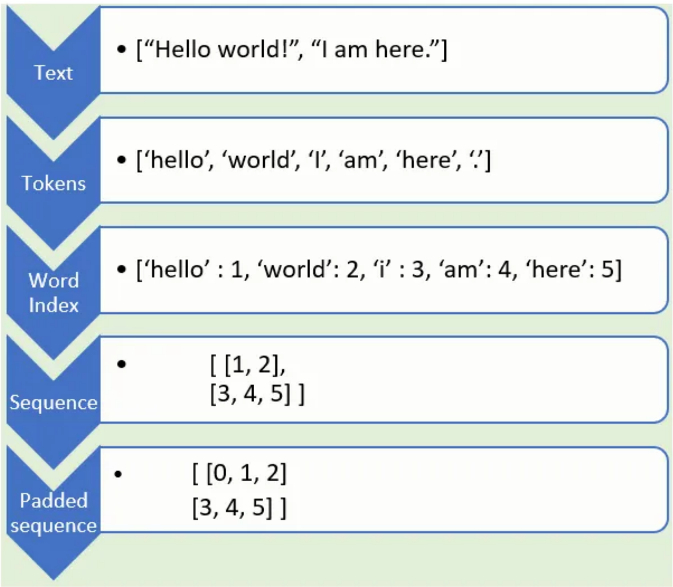

Text
• [“Hello world!”, “I am here.”]
Tokens
• [‘hello’, ‘world’, ‘I’, ‘am’, ‘here’, ‘.’]
Word Index
• [‘hello’ : 1, ‘world’: 2, ‘i’ : 3, ‘am’: 4, ‘here’: 5]
Sequence
• [[1, 2],
[3, 4, 5]]
Padded sequence
• [[0, 1, 2]
[3, 4, 5]]

Figure 8: NLP – text pre-processing

Note: The data has been split into train, validation and test set for the financial_phrasebank dataset, for the NYTimes dataset we will be predicting the sentiments for all the data, so haven't split into train and test, everything is test.

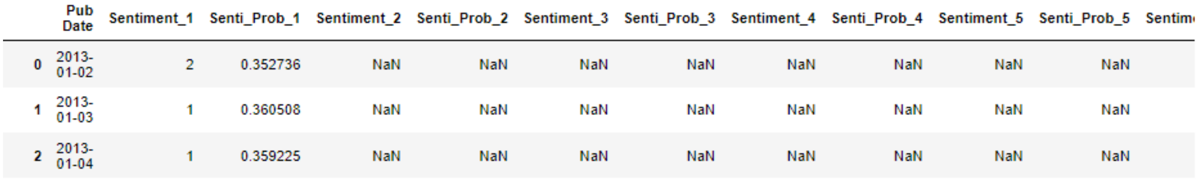
Now the data is ready to give as an input to the model.

3.2.3.2 for FinBERT and LLM-ChatGPT3.5 model

The FinBERT and LLM_ChatGPT-3.5 models are pre-trained models, so we need to give the input text/sentence as such the model is pre-trained to take care of the cleaning and pre-processing. So just check for NaN and send it to the model.

3.2.4 Post-processing of NLP model outputs to merge with the other features:

The results we get from the NLP models have multiple news for one day and few days without news, in order to merge this sentiment data with the main dataset containing TI and FI. We have grouped the results by date and spread the multiple news of a day across the columns, like as under,



	Pub Date	Sentiment_1	Senti_Prob_1	Sentiment_2	Senti_Prob_2	Sentiment_3	Senti_Prob_3	Sentiment_4	Senti_Prob_4	Sentiment_5	Senti_Prob_5	Sentim
0	2013-01-02	2	0.352736	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
1	2013-01-03	1	0.360508	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	2013-01-04	1	0.359225	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

Then formulated logics to get one domiant sentiment for a day. The logics are as under,

- Count the no. of negative, neutral and positive sentiments and choose the sentiment with higher counts, eg: if the count_neg = 2, Count_neu = 1 and Count_pos = 0, then in this case the dominant sentiment is 0.

- If the count of negative and neutral or positive and neutral are same then, I have chosen the dominat sentiment to be neg and pos respectively, eg: if the count_neg = 0, count_neu =2, and count_pos = 2, in this case the dominant sentiment is positive.

- If the counts of negative and positive are same, then I have gone one step further and compared the intensity of the news by comparing the sentiment scores of the sentiments and go with the higher sentiment score. Eg: if the count_neg = 1, count_neu = 0, count_pos =1, in this case compare the sentiment scores senti_score_neg = 0.9, senti_score_pos = 0.6, now the dominant sentiment is negative in this case, because the sentiment score of negative is greater than the sentiment score of the positive sentiment, which means the negative news on that day has higher intensity than the positive news of the day, so going with negative news.

The final output of the sentiment analysis on three NLP techniques are stored in separate pickles for later use in our dataset preparation. The data has 1474 instance with 2 columns namely date and dominant_sentiment.

3.3 Feature Selection and Data splitting

From the literature survey with the reference from the previous works we have chosen 12 technical indicators and 12 financial indicators for analysis, before taking all the indicators to the modelling stage for prediction, let us conduct a statistical test for significance which are utilized to determine whether a particular hypothesis about a relationship between variables is valid or not. The below flowchart for choosing a statistical test is referred from https://www.scribbr.com/statistics/statistical-tests/

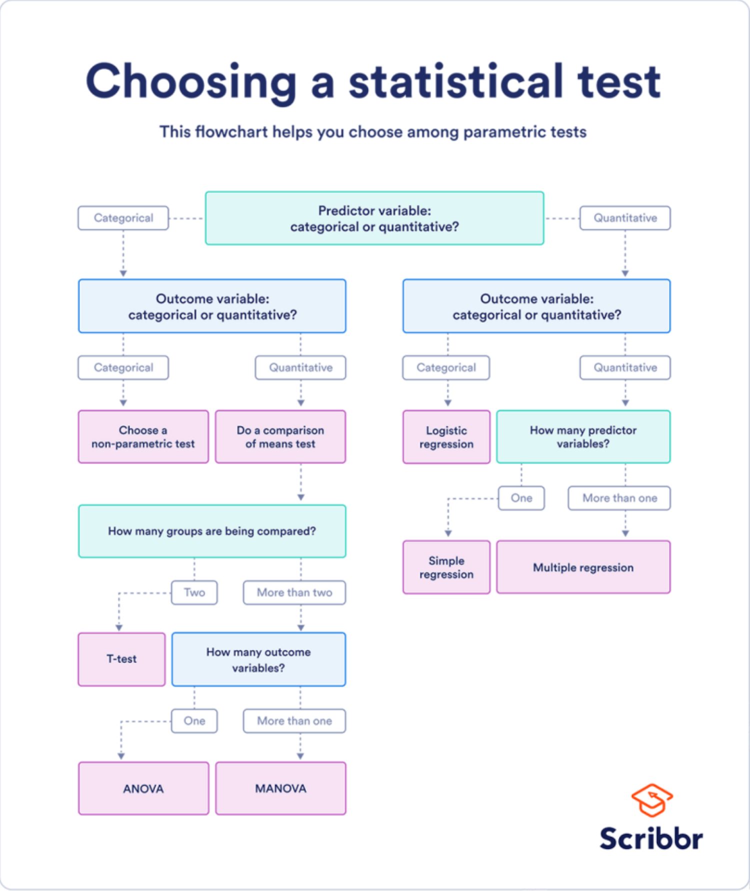

Choosing a statistical test
This flowchart helps you choose among parametric tests
Categorical
Predictor variable: categorical or quantitative?
Quantitative
Outcome variable: categorical or quantitative?
Outcome variable: categorical or quantitative?
Categorical
Quantitative
Categorical
Quantitative
Choose a non-parametric test
Do a comparison of means test
Logistic regression
How many predictor variables?
One
More than one
How many groups are being compared?
Simple regression
Multiple regression
Two
More than two
T-test
How many outcome variables?
One
More than one
ANOVA
MANOVA
Scribbr

Figure 9: General Flowchart for choosing a statistical test

The flowchart for choosing the statistical test for the subject case is as under,

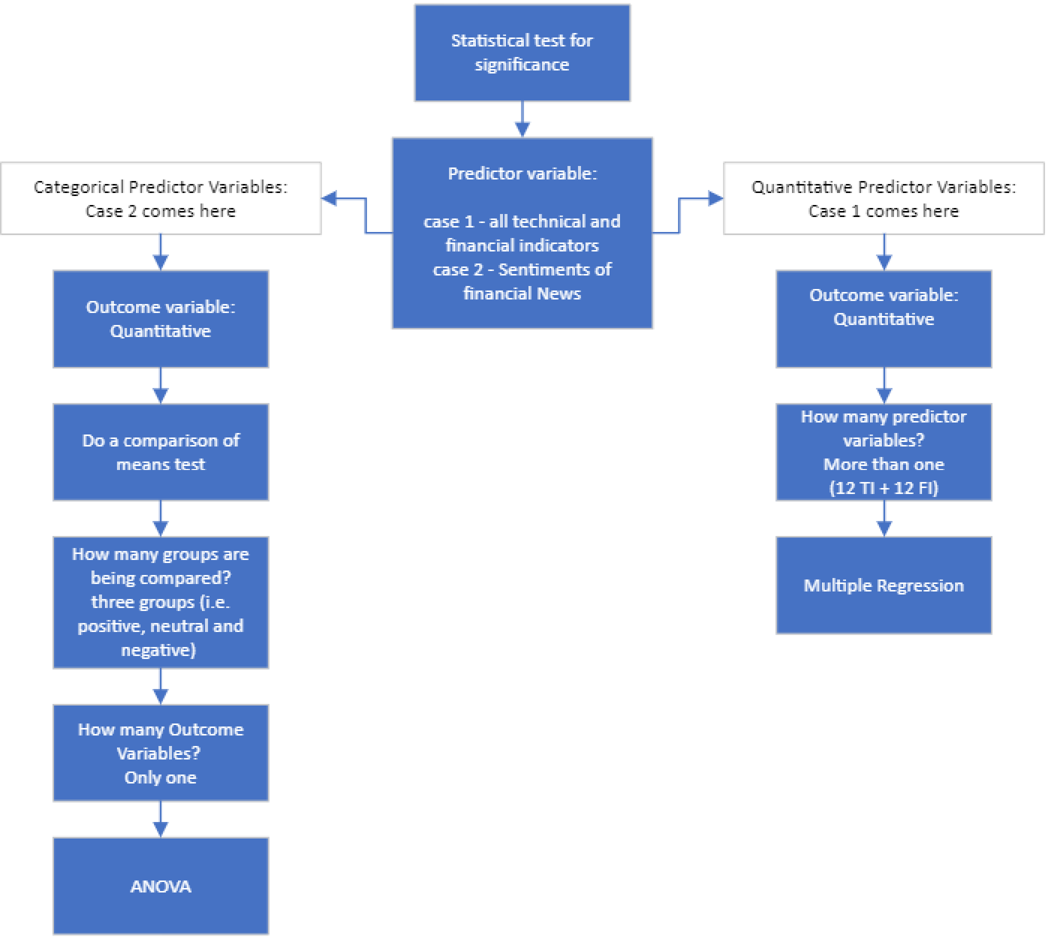

Statistical test for significance
Predictor variable:
case 1 - all technical and financial indicators
case 2 - Sentiments of financial News
Categorical Predictor Variables: Case 2 comes here
Quantitative Predictor Variables: Case 1 comes here
Outcome variable: Quantitative
Do a comparison of means test
How many groups are being compared? three groups (i.e. positive, neutral and negative)
How many Outcome Variables? Only one
ANOVA
Outcome variable: Quantitative
How many predictor variables? More than one (12 TI + 12 FI)
Multiple Regression

Figure 10: Flowchart for choosing a statistical test.

Acceptance and Rejection Levels in Statistical Significance Testing

We conducted Multiple Regression test and ANOVA test in the appropriate places and first let me fix the acceptance and rejection levels in the statistical significance testing.

In hypothesis testing, the acceptance and rejection levels are determined by the significance level (α), which is the probability of rejecting the null hypothesis when it is true. I standard choice of alpha is 0.05 and we are also using the same.

- If $P < \alpha$: The result is declared statistically significant, and the null hypothesis is rejected.
- If $P \geq \alpha$: The result is not considered statistically significant, and the null hypothesis is not rejected.

Here P is the p-value obtained in the test and it represents the probability of observing a test statistic as extreme as the one computed, assuming the null hypothesis is true.

Interpreting the Multiple Regression and ANOVA Results

Case 1. (TI & FI): Multiple Regression Analysis:

In the context of multiple regression analysis, each coefficient’s p-value tests the null hypothesis that the coefficient is equal to zero (no effect). A low p-value (< 0.05) indicates that you can reject the null hypothesis. In other words, a predictor that has a low p-value is likely to be a meaningful addition to your model because changes in the predictor's value are related to changes in the response variable.

Considerations:

- R-squared: Indicates the proportion of the variance in the dependent variable that the independent variables explain.
- Coefficients: Represent the mean change in the dependent variable for one unit of change in the predictor variable while holding other predictors in the model constant.
- P-value for each coefficient: Tests the null hypothesis that the coefficient is equal to zero.

Case 2. Analysis of Variance (ANOVA) Test:

ANOVA is used to analyze the differences among group means in a sample. The ANOVA test has important assumptions that must be satisfied in order for the associated p-value to be valid:

- The samples are independent.
- Each sample is from a normally distributed population.
- The population standard deviations of the groups are all equal.

Interpretation:

- F-statistic: A larger F-statistic suggests that the model explains a lot more variability in the outcome variable than a model that does not include any predictors.
- P-value for the F-statistic: Tests the null hypothesis that all of the model coefficients are equal to zero.

Discussion on the results:

In Dataset_A which is full of technical indicators we conducted Multiple Regression test and found that $p < 0.05$ for the following variables MACD, MACD_SIGNAL, MACD_HIST, ATR, RSI, OBV, NATR, TRIX, Stoch_SD, MAX, and STD, indicating they are statistically significant predictors of 'Close' at the 5% significance level. Whereas for ADXR, ADOSC, and AD variables the $p > 0.05$ and states that these variables are not statistically significant to the predictors.

In the Dataset_B which is full of technical indicator and financial indicators we conducted the Multiple Regression test and found that $p < 0.05$ for the following variables MACD, MACD_SIGNAL, MACD_HIST, ATR, RSI, OBV, NATR, TRIX, Stoch_SD, MAX, and STD, indicating they are statistically significant predictors of 'Close' at the 5% significance level. Whereas for all the financial indicator variables the $p > 0.05$ and states that these variables are not statistically significant to the predictors.

In the Dataset_C which is having sentiments of the financial news we conducted ANOVA test and found that the P-value is 0.000805, which is below the common alpha level of 0.05, this shows that $P < 0.05$ so the independent variable is statistically significant in predicting the dependent variable. The F-Statistic value is 7.145 and is used to determine the statistical significance and if it is substantially more significant than 1 and corresponding with a small P-value, it signals that there are noteworthy differences in group means. Finally, the "Dominant_Sentiment" variable does have a statistically significant effect on the dependent variable, "Close”.

Note: Although p-values inform whether a variable has a statistically significant relationship with the output, it does not quantify how well our model is able to predict the output for new data points, so considering the correlation and the practical knowledge (relying on the literature survey again) we are not omitting the financial indicators.

Correlation matrix:

The correlation matrix and Correlation with close price for the dataset C is presented here,

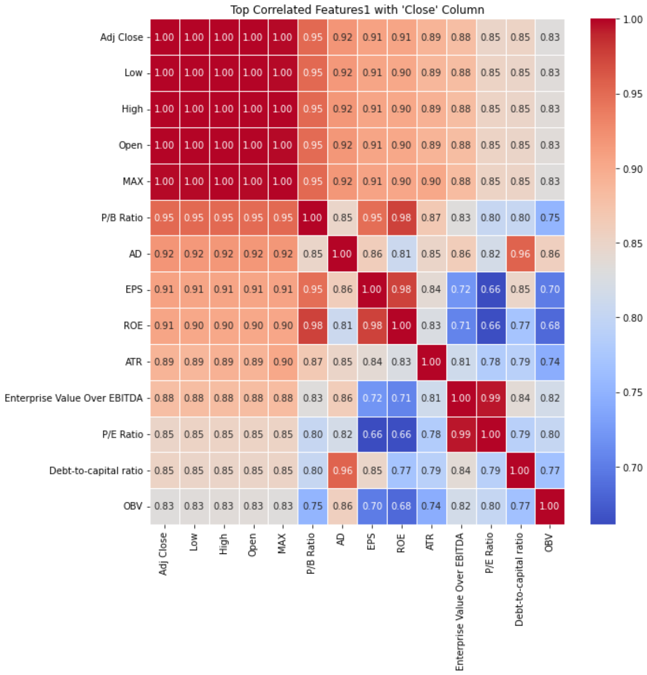


Figure 11: Correlation matrix of Dataset_C

Correlation with Close Price:

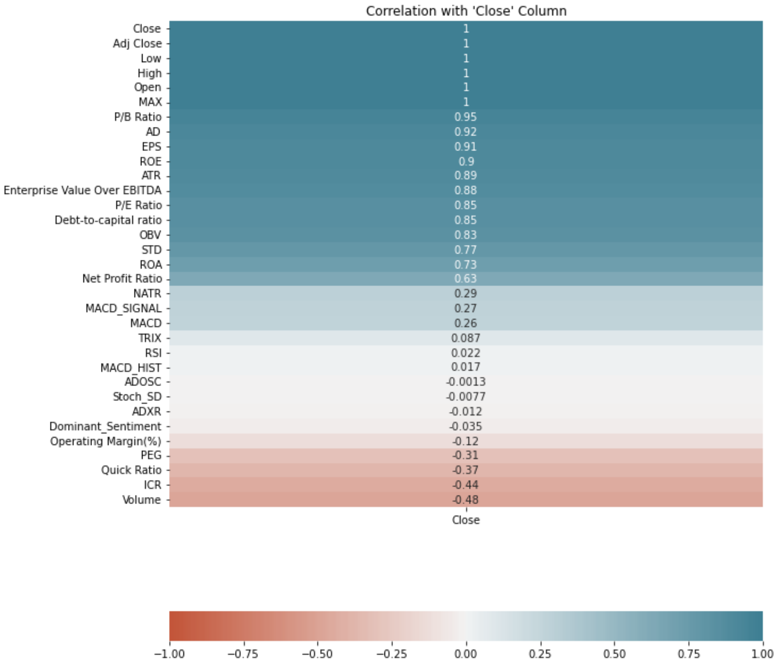


Figure 12: Correlation with close price for Dataset_C

Guided by statistical significance testing, correlation matrix analysis, and prevailing domain knowledge from the literature, feature selection was performed to mitigate collinearity issues and identify an optimal set of predictive variables. The selected technical indicators encompass **MAX, AD, ATR, OBV, and STD**. Meanwhile, the financial indicators comprise **P/B Ratio, EPS, ROE, Enterprise Value Over EBITDA, P/E Ratio, Debt-to-capital ratio, ROA, Net Profit Ratio, and ICR.** Additionally, the **dominant_sentiment** derived through NLP constitutes a pivotal variable.

With the practical knowledge we don't give the Low, High and Adj Close and Volume as input to the model, because practically these inputs are not available for the model to predict the close price of the day, in future. Even though we know the Open price during the start of the day, we don't give it to the model to predict the close price of the day, because the open price is almost +/- 1 to 2% almost all the time, so the model gets overfitted with this variable.

Data splitting for modelling:

After the selection of feature save it is x and the target variable in y, since this is a time series data make the shuffle – False and split the last 10% of the data for testing and the first 90% data for training and validation.

3.4 Stock Price Prediction Models

3.4.1 Random Forest (RF) Regressor

The Random Forest (RF) Regressor is an ensemble learning method widely utilized for regression and classification tasks. This algorithm has garnered commendation for its robustness and ability to handle complex datasets, making it a viable candidate for stock price prediction. By constructing multiple decision trees during the training phase and outputting the average prediction of individual trees for regression tasks, the RF Regressor can capture the non-linear dependencies in the stock market data effectively and thereby predict future prices with notable accuracy.

Working Principle and Architecture, Random Forest operates by creating multiple decision trees and merging them together to get a more accurate and stable prediction. Each tree in the forest is built by using a random sample of the training data. During the construction of a tree, each split is determined using a random subset of the features, adding an extra layer of randomness to the model, which helps to make it more robust and prevent overfitting. In a regression context, the final prediction is made by averaging the predictions from all trees in the forest. Here is a pictorial representation of how the random forest regressor works reference:https://medium.com/@bhatshrinath41/a-comprehensive-guide-to-random-forest-regression-43da559342bf,

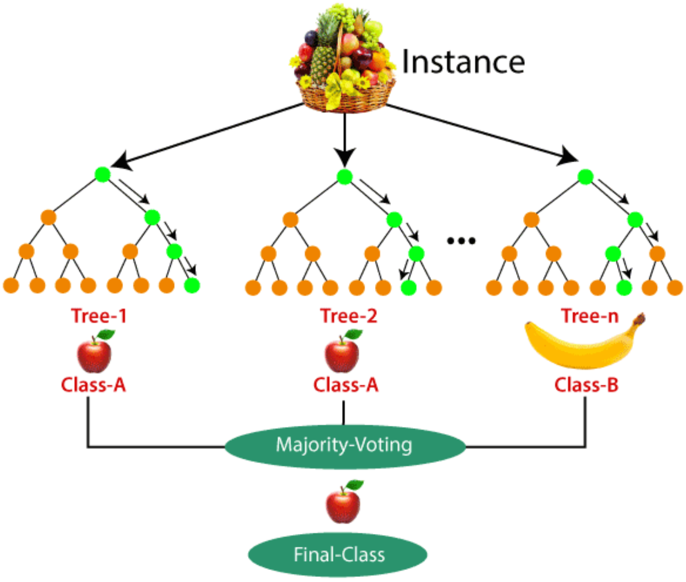


Figure 13: Random Forest Regressor – working flowchart

Parameters Tuning, Hyperparameter tuning in a Random Forest Regressor involves adjusting various parameters to enhance the model's predictive capacity and the hyper parameter are,

- **n_estimators**: Number of trees in the forest. Increasing the number typically improves accuracy but also computational cost.
- **max_depth**: The maximum depth of each tree. Constraining the tree depth helps to prevent overfitting.
- **min_samples_split**: The smallest number of samples required to split a node. Larger values prevent creating nodes that only fit a few samples.
- **min_samples_leaf**: The minimum number of samples required to be in a leaf node. Larger numbers prevent creating leaves that only fit a few samples, reducing overfitting.

Implementation of RF regressor for stock price prediction

The implementation initiates with the demarcation of dependent and independent variables, followed by partitioning the data into training and test datasets, adhering to a 90-10 split. Normalize the data by fit and transform to train data and only transform to the test data, which avoids the data leakage from test to train data. After this, an objective function is defined to delineate the hyperparameter tuning space and return the Mean Absolute Error (MAE) from cross-validation. The model’s hyperparameters are meticulously optimized using a technique, potentially employing an optimization library - **Optuna**, culminating in the determination of optimal parameters. The explored hyperparameter ranges are:

- n_estimators: 20 to 200,
- max_depth: 5 to 50,
- min_samples_split: 2 to 15,
- min_samples_leaf: 1 to 10

Executing the study yields the best hyperparameters corresponding to minimized loss. Subsequently, the model is retrained using these optimal parameters and validated with unseen test data, facilitating the extraction of performance metrics 1 and 2, as specified in section 3.6.

3.4.2 Long Short-Term Memory (LSTM) Model

Long Short-Term Memory (LSTM) networks, a specialized kind of Recurrent Neural Networks (RNNs), have become a cornerstone in predicting sequences and time-series data due to their capacity to remember information for prolonged periods, which is crucial for stock price prediction. Given the temporal dependencies in stock prices, where future prices are influenced by the preceding ones, LSTMs are apt for this task. They can model intricate patterns and sequences in the stock price data, enabling them to predict future prices based on the historical ones, while effectively managing challenges like vanishing or exploding gradients, which are prevalent in traditional RNNs.

Working Principle and Architecture of LSTM, LSTM networks are engineered to remember patterns over time, making them adept for tasks involving sequences, such as time-series forecasting. The architecture consists of a chain of neurons, where each cell encompasses three gates: the forget gate, input gate, and output gate. The forget gate decides what information to discard from the cell state, the input gate updates the cell state with new information, and the output gate determines the next hidden state. By governing the flow of information to be remembered or forgotten at each time step through these gates, LSTMs can maintain and leverage long-term dependencies in the data, thus facilitating in modeling sequences effectively, like temporal patterns in stock prices.

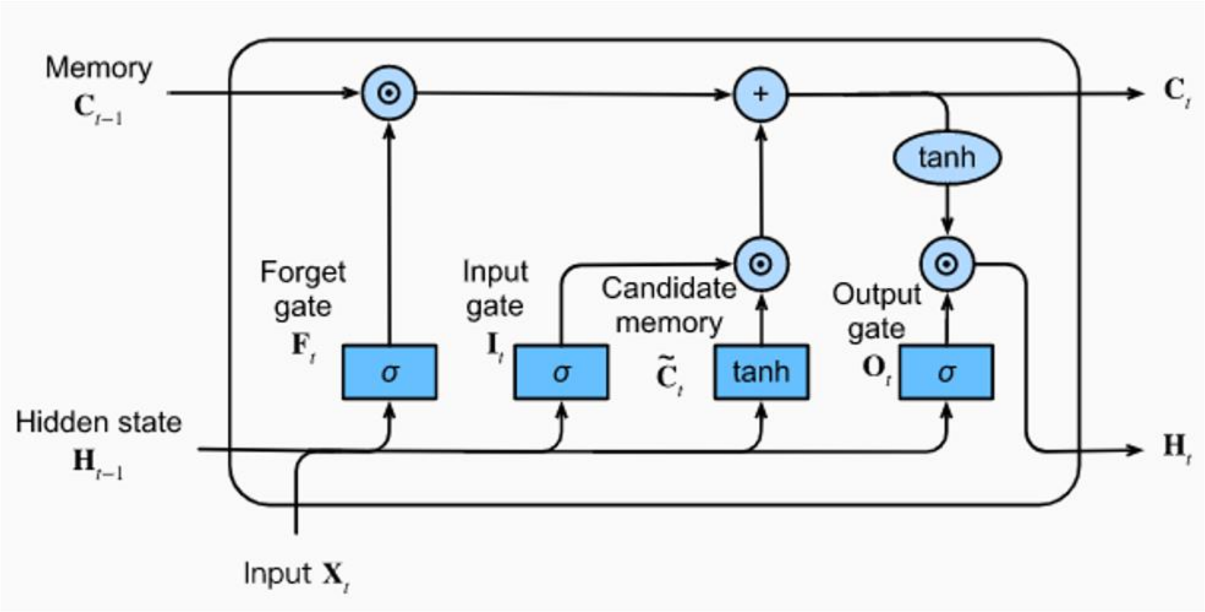


Figure 14: LSTM Architecture

Hyperparameter Tuning in LSTM, Tuning an LSTM model involves adjusting various hyperparameters to enhance its predictive performance. Key hyperparameters include the number of neurons in the LSTM layer, the number of layers, batch size, and epochs. Moreover, optimization algorithms (like Adam or SGD), learning rate, and dropout rate for regularization also significantly impact the model’s capability to generalize. Tuning can be approached through grid search, random search, or more sophisticated optimization methods such as Bayesian optimization. An apt selection of hyperparameters can greatly influence the LSTM's ability to learn from historical stock prices and subsequently its effectiveness in predicting future prices, thereby providing insightful forecasts and facilitating more informed trading decisions.

Implementation of LSTM Model for Stock Price Prediction

In this project, we used Expanding Window method for time-series cross-validation, using the **TimeSeriesSplit** function with **n_splits=5**, ensuring that the temporal structure of the data is preserved during model validation.

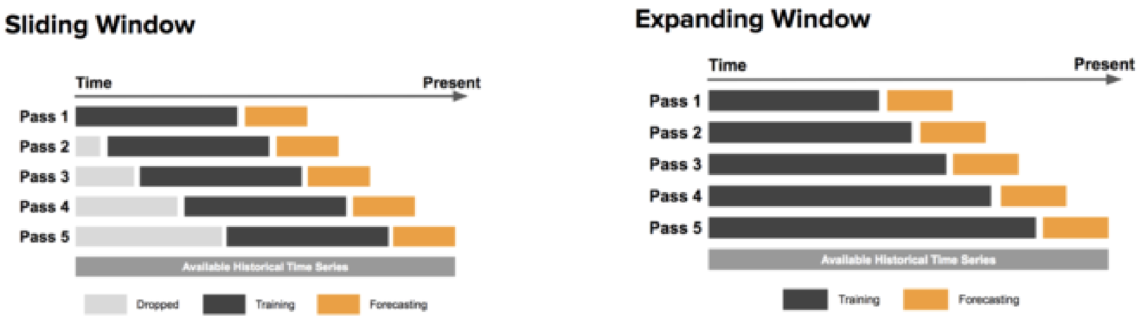


Figure 15: Sliding and Expanding window cross validation

Expanding Window Cross-Validation (CV) is a technique optimal for time-series data, where, unlike the fixed-size window in Sliding Window CV, the training window expands over time while maintaining the chronological order. Essentially, this method involves training the model on an initially small dataset, gradually incorporating more data in subsequent iterations, thereby consistently utilizing all available historical data for training and ensuring that predictions are generated for unseen, future data points.

The architecture and training process of the LSTM model is as follows:

- **Data Scaling**: Both the input features and target variable are scaled using **MinMaxScaler** to fit and transform the training data, while only transforming the test data, ensuring that the neural network receives appropriately scaled inputs.
- **Model Architecture**: The LSTM model is defined sequentially with the following layers:
    - First LSTM layer with 50 neurons, returning sequences, and accepting an input shape corresponding to the number of features.
    - A Dropout layer for regularization, with a dropout rate of 0.1, reducing the likelihood of overfitting by ignoring randomly selected neurons during training, and hence decreasing the model's sensitivity to specific weights.
    - A second LSTM layer with 50 neurons, not returning sequences.
    - A Dense layer with 25 neurons to further process the sequence information.
    - An output Dense layer with a 1 neuron, as we are predicting a single continuous output (stock price).

- **Model Compilation**: The model is compiled with the **Adam** optimizer, specifying a learning rate of 0.0001 and clipping the gradients to a maximum value of 1.0 to prevent exploding gradients. The loss function is Mean Squared Error, which is standard for regression problems, aiming to minimize the square of the prediction errors.
- **Training with Early Stopping**: The model is trained for a maximum of 20 epochs with a batch size of 1 using **fit**, with early stopping (monitoring validation loss with **patience=3**) to prevent overtraining by halting training when the validation loss does not decrease for 3 consecutive epochs. This way, the model doesn't over-learn the training data, preserving its ability to generalize to new data.

During each fold of the time-series cross-validation, the model is evaluated and metrics (MAE, MSE, and MAPE) are computed for performance assessment. Then, predictions are produced and analysed against actual values. A line plot illustrating the training and validation loss across epochs provides insights into the model's learning progression and aids in diagnosing issues like overfitting or underfitting.

Finally, the model is employed to make predictions on an unseen test set, providing a practical gauge of its forecasting capability when faced with novel data. Metrics computed on this test set furnish a conclusive evaluation of the model's predictive prowess, culminating in a comprehensive assessment of the LSTM network in the context of stock price prediction. This holistic methodology ensures rigorous validation and testing, safeguarding against overfitting and underfitting, while providing reliable predictive performance estimates.

3.5.3 Hyper Parameter Optimized LSTM Model

In this segment, the architecture has been tuned to optimize the model's predictive capabilities and to manage the challenge of temporal dependencies and sudden market shifts effectively.

In an attempt to boost the performance of the LSTM model, hyperparameter tuning has been executed as under,

- **Number of LSTM Layers:** Range – (1, 2)

  The model can have either 1 or 2 LSTM layers, allowing for variation in model depth and complexity.

- **Number of Neurons:** Range for each layer – 32 to 256 with step: 32

The model experiments with various sizes of LSTM layers, from the lower complexity (32) to a higher one (256), in steps of 32, to explore different capacity levels.

- **Dropout Rate:** Range for each layer - 0.0 to 0.5 with Step: 0.1

  Dropout rates are considered at every layer to introduce regularization, aiming to prevent overfitting by iteratively exploring values from no dropout (0.0) to a moderate dropout level (0.5), in steps of 0.1.

- **Learning Rate for Adam Optimizer:** 0.1, 0.01, 0.001, 0.0001

  The model investigates diverse learning rates, from fast learning (0.1) to very slow learning (0.0001), to gauge how speedily or gradually the model adapts its weights during training.

Each point has its specified parameter range or choices, providing a clear view of the potential configurations the model might assume during the hyperparameter optimization process. This wide range of parameter space is catered through **Keras Tuner**, specifically the **RandomSearch strategy**, is employed for exploring the parameter space and choosing sets of parameters that minimize the validation Mean Squared Error (MSE). In this experiment maximum trail of the random search is limited to 50 trails. It automates the experimentation of different architectures and hyperparameters, ensuring that the model does not settle for a sub-optimal solution. Furthermore, to avoid overfitting and to save computational resources, **early stopping** is applied during training, which halts the learning process if the model ceases to improve on the validation data.

As a result of this hyperparameter tuning, the optimal model and its corresponding best hyperparameters have been determined. Now, we will evaluate the best model using the unseen test data, which consists of the last 10% of the historical data, and derive performance metrics 1 and 2, as detailed in Section 3.6 for comparison with other models.

The best hyper parameters are consolidated and presented as under,

Best Hyperparameter						
Experiment no.	**Experiment_3**	**Experiment_5**	**Experiment_7**	**Experiment_9**	**Experiment_11**	**Experiment_13**
Experiment Description \\ parameters	**Dataset_A with Hyperparameter Optimized best LSTM Model**	**Dataset_B with Hyperparameter Optimized best LSTM Model**	**Dataset_C with Hyperparameter Optimized best LSTM Model**	**Dataset_D1 with Hyperparameter Optimized best LSTM Model**	**Dataset_D2 with Hyperparameter Optimized best LSTM Model**	**Dataset_D3 with Hyperparameter Optimized best LSTM Model**
Layer 1 - Neurons	256	96	128	192	96	192
Layer 1 - dropout	0.4	0.3	0	0.4	0	0.4
Layer 2 - Neurons	-	160	-	-	256	-
Layer 2 - dropout	-	0.3	-	-	0	-
Layer 3 - Neurons	96	96	192	64	128	96
Layer 3 - dropout	0	0.1	0.1	0	0	0.1
Dense layer	1	1	1	1	1	1
Learning Rate	0.001	0.001	0.1	0.01	0.001	0.01

Table 1: Best Model's hyperparameters - consolidated results

3.5 Sentiment Analysis of Financial News

In this section we will focus on the working principle, architecture, and implementation of the model for our analysis for the following three models which we used for sentiment analysis of financial news.

- Bi-LSTM model with word2vec
- Pretrained FinBERT model
- Large Language Model – ChatGPT3.5

3.5.1 Bi-LSTM with Word2Vec

Bi-Directional Long Short-Term Memory (Bi-LSTM), an evolution of traditional LSTM networks, explores the significant capability of understanding sequences by manoeuvring through input data in both forward and backward directions. This allows the model to have a more comprehensive context for each data point, amplifying its ability to recognize patterns that are pertinent to sequences observed in the financial text, where the order and relationships between words often hold essential predictive power.

Word2Vec, a robust model developed to transform words into high-dimensional vectors, provides a spatial domain where words with similar meanings occupy proximate locations. This proximity in the vector space inherently captures semantic and syntactic relationships between words, facilitating the model to understand contextual similarities and differences amongst them, which is crucial when deciphering meaning from financial texts laden with jargon and specific terminologies.

Interplay in Sentiment Analysis, the amalgamation of Bi-LSTM and Word2Vec in sentiment analysis intertwines the sequential memory and bi-directional context understanding of the former with the semantic depth offered by the latter. Such integration allows the model to adeptly navigate through the labyrinth of financial language, identifying not just explicit sentiments but also understanding the nuanced emotional undercurrents concealed within the sophisticated textual data. This sophisticated approach endeavors to pinpoint the subtleties of positive, negative, and neutral sentiments within financial news, which often encapsulate investor perceptions and market moods, thereby playing a pivotal role in predicting potential stock price movements.

Implementation of Bi-LSTM for the Sentiment Analysis of news data

Data split, preprocessing and embedding preparation, should be implemented as mentioned in the section 3.2.3

Model Architecture and Compilation, we construct an embedding matrix tailored to our specific vocabulary, ensuring the model utilizes relevant embeddings during training. Then, we assemble our Bi-LSTM model by commencing with an **Embedding** layer which uses our created embedding matrix and is set as non-trainable, ensuring the word embeddings remain static during training. The subsequent **Bidirectional LSTM** layer, consisting of 100 neurons, processes the embedded sequences, capturing dependencies and contextual information from both directions in the sequences. Concluding with a **Dense** layer containing three neurons and utilizing softmax activation, the model outputs probability distributions across the three sentiment classes. Compiled with the 'adam' optimizer and 'sparse_categorical_crossentropy' loss function, it is aptly designed for the classification task at hand.

Training and Prediction, model training, spanning 15 epochs with a batch size of 16, utilizes a validation dataset for on-the-go performance evaluation, and once trained, it predicts class probabilities for the test set. By determining the class with the maximum probability using argmax, we translate the model’s predictions into tangible class labels, ready for evaluation against actual test labels. This thorough implementation, integrating Word2Vec embeddings with Bi-LSTM, embodies a holistic approach, ensuring robust semantic understanding and sequence modelling, aiming for potent sentiment classification in financial news analysis.

3.5.2 FinBERT

The methodology implemented utilizes a pretrained FinBERT model, developed by Huang, Allen H., Hui Wang, and Yi Yang (2022), and is accessible via Hugging Face – a company of French-American origin, celebrated for crafting tools that enable the construction of applications underpinned by machine learning, particularly through its distinguished 'transformers' library, dedicated to applications in natural language processing.

FinBERT stands out as a BERT model, which has been pretrained using text related to financial communication, intending to boost both the research and practice tied to financial NLP. This model has been trained on three distinguished financial communication datasets, amassing a total of 4.9B tokens, which includes Corporate Reports (10-K & 10-Q) with 2.5B tokens, Earnings Call Transcripts with 1.3B tokens, and Analyst Reports contributing 1.1B tokens.

The variant of the model we utilized, finbert-tone, has been finely tuned on 10,000 manually annotated sentences (categorized as positive, negative, or neutral) sourced from analyst reports. This specific model showcases excellent performance in analyzing financial tone. It is recommended for those who seek to utilize FinBERT specifically for financial tone analysis endeavors.

By referring the paper, we understood that the architecture depicted in the subject BERT model specialized for sequence classification tasks, instantiated via the **BertForSequenceClassification** class. BERT (Bidirectional Encoder Representations from Transformers) is a transformer-based architecture known for its powerful contextualized embeddings, achieving state-of-the-art results across various NLP tasks. The architecture begins with **BertEmbeddings**, which creates word embeddings that also consider positional and token-type information. The core model, **BertEncoder**, consists of several identical layers (**BertLayer**), each employing a multi-head self-attention mechanism (**BertAttention**) and a position-wise feed-forward network. The attention mechanism involves generating query, key, and value representations, followed by scaled dot-product attention and output linear transformation. Following attention, an intermediate feed-forward network (**BertIntermediate**) with GELU activation transforms the representation, which is subsequently refined by another linear layer in **BertOutput**. This encoder processes input embeddings into high-level features, which can be used for various downstream tasks. Each layer and subcomponent employs dropout and layer normalization for stable and robust training. This deeply stacked, self-attention mechanism allows BERT to capture complex syntactic and semantic information, making it apt for a multitude of NLP applications, including the sequence classification depicted here.

The input dataset is supplied to the aforementioned model, which preprocesses and interacts with the data to determine the sentiment of the statements.

3.5.3 Large Language Model (LLM) – ChatGPT-3.5

In the context of sentiment analysis within the domain of financial news, Zhang, Deng, Liu, Pan & Bing (2023), through their work titled "Sentiment Analysis in the Era of Large Language Models: A Reality Check", illuminate the capabilities of Large Language Models (LLMs) in comprehending and deriving sentiment from textual data. Utilizing the API guidelines of ChatGPT-3.5, and the above paper we have formulated the implementation of ChatGPT3.5 for our subject task.

Preliminary Set-Up and Agent Function Deployment, the foundational step involved the importation of necessary libraries and configuration of API keys, crucial for interfacing with ChatGPT-3.5. The creation of an 'agent' function acted as a crucial intermediary, ensuring a streamlined and error-managed communication with the API throughout the request-response cycle.

Crafting and Strategizing Prompts for LLM, core to the methodology is the neatly crafted prompt, enriched by **few shot prompting strategy** to navigate the LLM in delivering precise sentiment analyses. The prompt carefully incorporates diverse financial scenarios and correlated sentiments as examples, bestowing the model with a thorough contextual framework and thereby refining its ability to decode sentiments in financial headlines.

Iterative Development and Feedback Integration, the development of a proficient prompt journeyed through numerous iterations of predictions and meticulous validations. Initiated with a clear-cut instructive: " You are a financial analyst. Your job is to analyze news headlines and predict whether the sentiment is Positive, Negative, or Neutral. ", the model was set on a path devoid of domain-bias, preserving the authenticity of early predictions. Ensuing predictions were scrutinized, with erroneous outputs providing invaluable insights into the model's initial understanding of financial sentiment.

Enhancing Contextual Understanding Through Iteration, misinterpretations were analyzed and utilized to inform the development of clarifying examples, thus gradually calibrating the model's perception and enhancing its analytical accuracy in financial contexts. A spectrum of seven example statements with labels, emerged from prior mispredictions, was embedded into the prompt, each presenting diverse financial contexts and associated sentiments to richly inform the model's understanding.

Data Preparation and Analytical Execution, conclusive steps entailed data preprocessing to tailor financial news data for ChatGPT-3.5 analysis. Data, once prepped, was dispatched via API requests, navigated by the now meticulously refined prompt, to extract sentiments. Subsequent predictions were collated and archived, ready to undergo further validation processes.

Note: We have refined the prompt using the financial_phrasebank dataset and tested the same dataset to compare the accuracy of this model with other models. Furthermore, the same prompt was applied to the NYTimes dataset to predict the sentiment of the corpus.

3.6 Validation and Performance Metrics

In this section we will focus on the performance metrics used to validate the models. In the validation perspective the complete work can be split into three types as following,

- First, all the NLP models on the labelled dataset (financial_phrasebank dataset).

- Second, all the NLP models on the non-labelled dataset (NYTimes dataset).

- Third, the main task – all the of stock price prediction models on the Datasets (A, B, C, D1, D2, D3)

3.6.1 NLP Models on labelled dataset

The evaluation of Natural Language Processing (NLP) models, particularly when scrutinized against a labelled dataset, necessitates a meticulous approach to ensure that the derived sentiments accurately mirror the underlying emotional tone encapsulated within the financial text. The financial_phrasebank dataset, owing to its labelled nature, provides a propitious platform for validating the sentiment prediction capabilities of our NLP models by comparing the predicted sentiments against actual labels. This validation metrics is referred from the book named “Introduction to Machine Learning with Python: A Guide for Data Scientists” by Mueller & Guido (2016)

Confusion Matrix gives a more granular view into the model's predictive capabilities by highlighting True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). This matrix is paramount in understanding not just where the model is correct, but also in discerning the nature and context of the errors it makes. These insights are instrumental in refining the model, adjusting thresholds, and potentially re-engineering features to enhance its predictive acumen.

Classification Metrics

- Accuracy: The quintessence of model evaluation, accuracy, elucidates the proportion of predictions that the model has discerned correctly. It embodies the ratio of correctly predicted sentiments (both positive and negative) to the total number of observations. While intuitively straightforward, accuracy can sometimes mask disparities in the predictive performance across different sentiment classes.

  Accuracy= (TP + TN) / (TP + TN + FP + FN)

- Precision: Precision, sometimes referred to as positive predictive value, illuminates the model’s aptitude to avoid false positives, essentially quantifying the accuracy of the model when declaring a sentiment as positive.

  Precision= TP / (TP + FP)

- Recall (or Sensitivity): A metric focusing on the model’s ability to identify and correctly predict all relevant instances, recall provides insight into how well the model is able to detect positive sentiments amidst the labyrinthine financial narratives.

  Recall= TP / (TP + FN)

- F1-score: Acting as the harmonic mean of precision and recall, the F1-score serves as a balanced metric, especially vital when the class distributions are imbalanced, by considering both false positives and false negatives.

  F1 Score= 2 × ((Precision × Recall) / (Precision + Recall))

Analysing these metrics cohesively grants a comprehensive view of the model's capabilities and potential areas of improvement. It elucidates the model’s proficiency in deciphering the subtle nuances enveloped within financial texts and its adeptness in reliably predicting sentiments. This meticulous validation through varied metrics ensures that the models are not just accurate but also robust, capable of handling the myriad of expressions and contextual intricacies embedded within financial discourses.

3.6.2 NLP Models on non-labelled dataset

Investigating the Natural Language Processing (NLP) models' proficiency on the NYTimes dataset, an inherently non-labelled set, poses a unique set of challenges, especially concerning validation. The absence of predefined labels essentially eradicates the feasibility of employing traditional validation metrics, demanding an innovative, albeit indirect, visual validation strategies. **Through this visual validation nothing is conclusive, it is for better understanding the model’s performance.**

Sentiment Class Distribution, In the absence of a labelled benchmark, evaluating the distribution of predicted sentiment classes across the dataset allows for an observational analysis. This inspection serves not as a concrete validation but as a method to ensure the model's outputs align reasonably with anticipated sentiment distributions, based on historical and contextual knowledge of the dataset’s content.

Hypothesis-Driven Validation, the formulated hypothesis – that the sentiment derived from financial news is directly proportional to stock close prices – becomes a linchpin around which an explorative validation approach is constructed. This hypothesis insinuates that positive news sentiment should theoretically correlate with a rise in close prices, while negative sentiment should see an inverse relationship.

Cumulative Sentiment and Close Price Correlation: Exploiting the 10-year expanse of data, the cumulative sentiment obtained from the NLP models is graphed alongside the close price trend. Here, the expectation is that the model adept at accurately deciphering sentiment from financial news will exhibit a trend line that correlates, to a degree, with the close price graph. Here is an example,

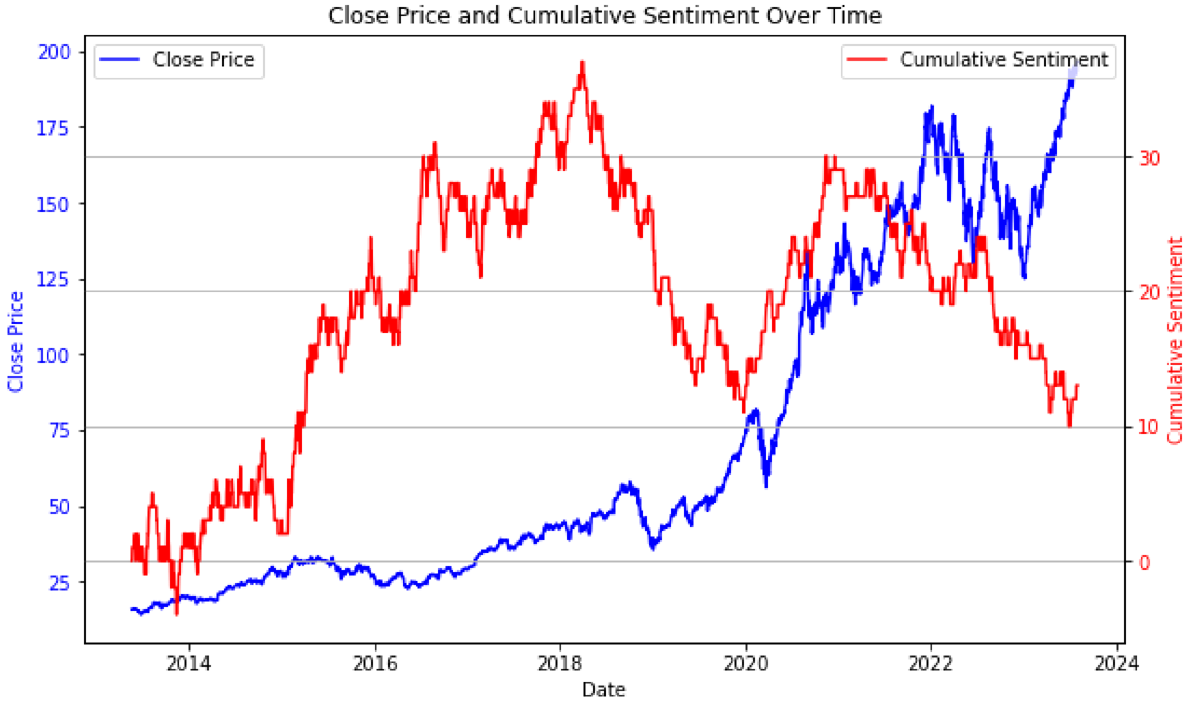


Figure 16: Close Price vs Cumulative sentiment over time - example

Analyzing Correlation and Divergence: the scrutiny of areas where the sentiment trend aligns with the close price, as well as where it diverges, provides a nuanced understanding of the model's performance. Notable peaks and troughs in the price trend should, ideally, be mirrored by analogous movements in the sentiment trend, affirming the model's capability to comprehend and translate financial news into relevant sentiment classes.

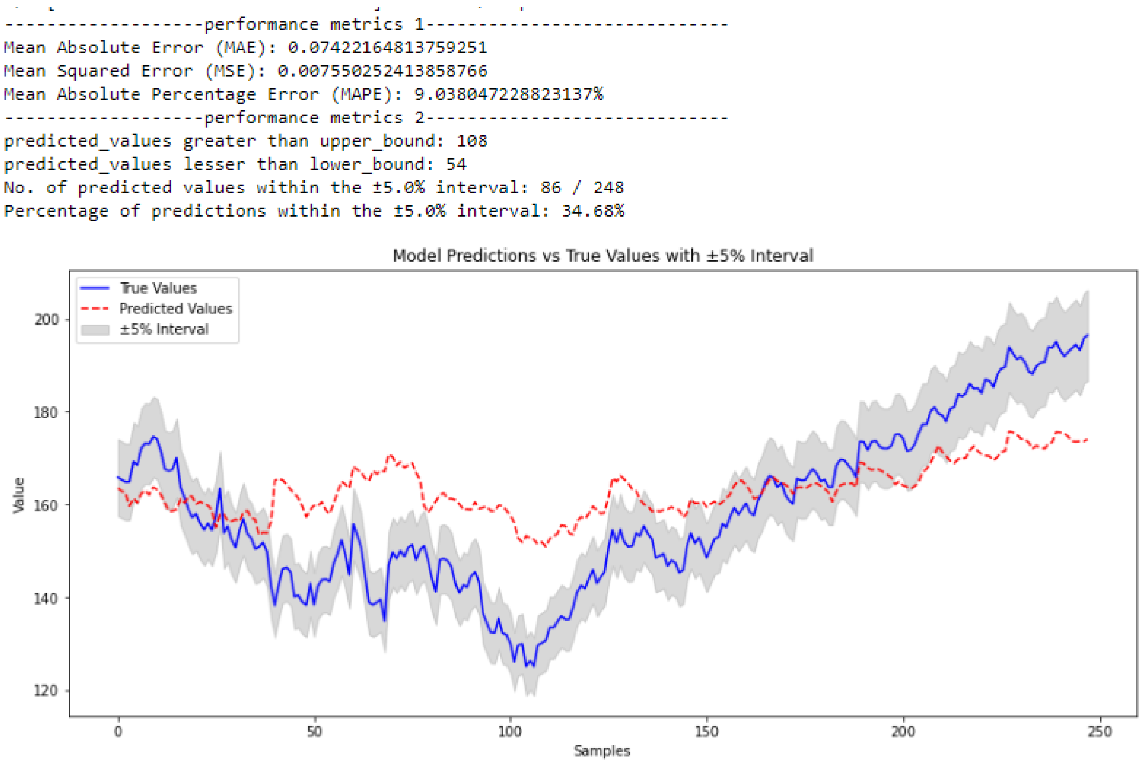
Visual Insights vs the empirical validation: Upon inspecting the above two visual methods of validation, a preliminary observation may surface, suggesting a particular model is better sentiment analysis. To substantiate this intuitive inference, the derived sentiments can be strategically applied to an LSTM model specializing in stock price prediction. This exercise pivots on investigating whether the visual coherence between sentiment and closing price trajectories is mirrored in a tangible improvement in the predictive outputs of the stock price prediction model. Integrating sentiment data as a feature into the LSTM model (i.e. Dataset D1, D2, and D3) allows for a subsequent, empirical evaluation of its predictive performance, thereby offering a metric-based verification of the initial visual assessment and ensuring that the model's perceived efficacy is reflected in a quantifiable enhancement of predictive accuracy in the realm of stock price forecasting. This amalgamation of visual and empirical validation techniques aims to provide a comprehensive and rigorous assessment of the NLP model's capability in the real-world application of financial forecasting.

3.6.3 Stock Price Prediction Models

In this research, two prominent models were adopted for stock price prediction: the Random Forest (RF) Regressor and Long Short-Term Memory (LSTM) networks. Fundamentally functioning as regression models, their performance was initially measured using traditional metrics such as Mean Absolute Error (MAE), Mean Squared Error (MSE), and Mean Absolute Percentage Error (MAPE), collectively referred to as **Performance Metrics 1.** This validation metrics is referred from the book named “Introduction to Machine Learning with Python: A Guide for Data Scientists” by Mueller & Guido (2016).

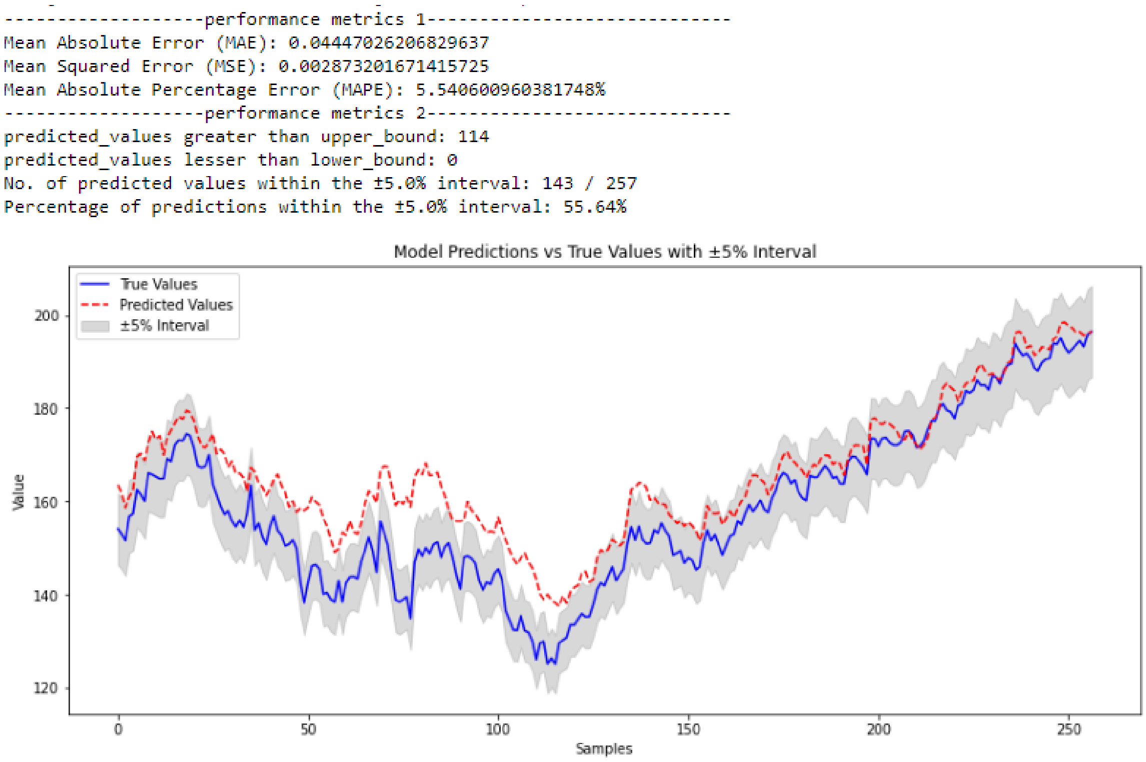
- Mean Absolute Error (MAE): This metric calculates the average absolute differences between the observed actual outcomes and the predictions made by the model.
  $MAE = (1/n) * \sum i=1n |yi - \hat{y}i|$
  where yi are the actual values and ŷi are the predicted values.
- Mean Squared Error (MSE): MSE measures the average of the squares of the errors, that is, the average squared difference between the estimated values and the actual value.
  $MSE = (1/n) * \sum i=1n (yi - \hat{y}i)2$
  where yi are the actual values and ŷi are the predicted values.
- Mean Absolute Percentage Error (MAPE): MAPE quantifies the average absolute percent difference between observed and predicted values, excluding the cases where the actual observation is zero.
  $MAPE = (n/100\%) * \sum i=1n | yi - \hat{y}i | / |yi|$
  where yi are the actual values and ŷi are the predicted values.

However, subsequent analysis, particularly on datasets A, B, and C through experiments 2, 4, and 6, highlighted a limitation: Performance Metrics 1 was somewhat inadequate in capturing and elucidating the trend in the stock price predictions, thereby presenting a challenge when attempting to coherently compare the model's effectiveness in the context of stock price prediction. For example, I have presented results of two experiments as under,



```
------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.07422164813759251
Mean Squared Error (MSE): 0.007550252413858766
Mean Absolute Percentage Error (MAPE): 9.038047228823137%
------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 108
predicted_values lesser than lower_bound: 54
No. of predicted values within the ±5.0% interval: 86 / 248
Percentage of predictions within the ±5.0% interval: 34.68%
```

Figure 17:Performance Metrics 2 - example 1



```
------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.04447026206829637
Mean Squared Error (MSE): 0.002873201671415725
Mean Absolute Percentage Error (MAPE): 5.540600960381748%
------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 114
predicted_values lesser than lower_bound: 0
No. of predicted values within the ±5.0% interval: 143 / 257
Percentage of predictions within the ±5.0% interval: 55.64%
```

Figure 18: Performance Metrics 2 - example 2

Comparing the MAPE of both the experiments the example 1 is 9.0% for example 2 it is 5.5%, but visually it is clearly evident that when compared to example 1, the example 2 has trained better and learnt the trends and noise of the close price to some extent and this is not reflecting clearly in the MAPE values, there is only slight change. In the stock price prediction task, it is very important to have a performance metrics which closely observes the trend and how far the prediction is mirroring the actual data line, because when the prediction goes beyond a bandwidth the stop loss may get triggered, so measuring the prediction and how it lies within or out of the bandwidth is very important in stock price prediction kind of problems.

In order to overcome this, I have devised **performance metrics 2** – plotting the confidence interval band of +/- 5% to the close price line and predicting how many no. of prediction points are with the limit, as the no. of points within the limit increases, the model is better. In our examples the example 1 is having percentage of prediction within interval of 34.6% and example 2 is with additional 20% that is 55.6%, whereas the difference showed by MAPE is 3.5%. From this performance metrics 2 talks more about the model predictivity.

4. Results

In this section we will focus on the outputs of the all the models on the all the datasets explained earlier,

4.1 Sentiment Analysis Model's Performance

In this section, the outputs of the three sophisticated NLP model's sentiment analysis on the labelled financial_phrasebank dataset and NYTimes dataset. The output contains,

For financial_phrasebank dataset:

- Classification Report

- Distribution of sentiments labels vs prediction

- Confusion Matrix

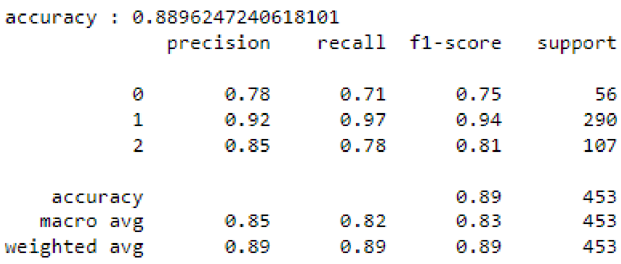
For NYTimes dataset:

- Distribution of predicted sentiments

- Cumulative sentiment vs close price graph

4.1.1 Senti Analysis 1 - Bi-LSTM (word2vec) on financial_phrasebank dataset

The sentiment analysis conducted in this section utilizes a Bi-LSTM (Bidirectional Long Short-Term Memory) model integrated with Word2Vec embeddings. The chosen dataset for this analysis is the financial_phrasebank, which primarily focuses on financial sentiments.



```
accuracy : 0.8896247240618101
              precision    recall  f1-score   support

           0       0.78      0.71      0.75        56
           1       0.92      0.97      0.94       290
           2       0.85      0.78      0.81       107

    accuracy                           0.89       453
   macro avg       0.85      0.82      0.83       453
weighted avg       0.89      0.89      0.89       453
```

Figure 19: Classification Report – Bi-LSTM with word2vec

The dataset categorizes sentiments into three distinct classes, 0 class - Negative, 1 class - Neutral, 2 class - Positive.

Overall Accuracy: The model showcases a prediction accuracy of 88.9%.

Precision, Recall, F1-score, and NPV:

- For Class 0: Precision stands at 0.78, recall at 0.71, and F1-score at 0.75.
- Class 1 impresses with a precision of 0.92, recall of 0.97, and an F1-score of 0.94.
- Class 2 exhibits a precision of 0.85, recall of 0.78, and an F1-score of 0.81.

The weighted average across precision, recall, and F1-score is consistent at 0.89.

Actual vs. Predicted Sentiment Distributions:

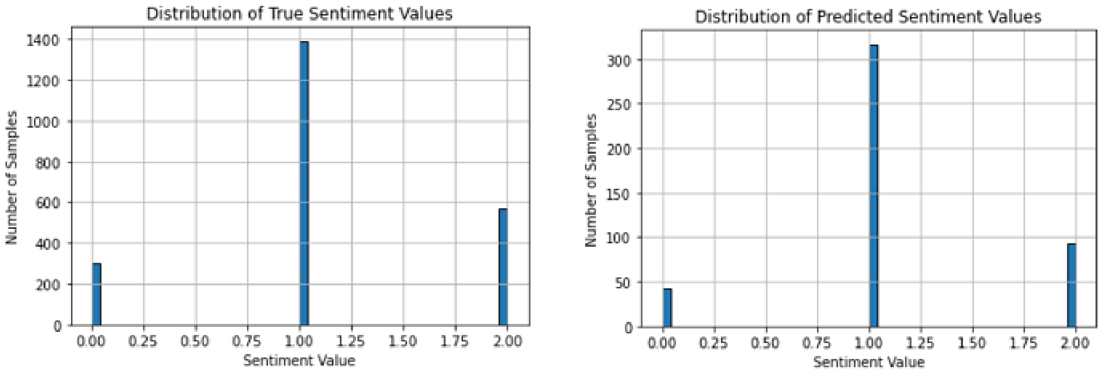


Figure 20: Class Distribution - True sentiments (left) and Predicted Sentiments (right) – Bi-LSTM word2vec

The above class distribution of true vs predicted sentiments visually clarifies that the classes of the predictions (right) are in the same proportion to the classes of true sentiment values (left). As evident in the above class distribution, there is a class imbalance inclined towards the neutral class, and our target is to predict the negatives and positives mainly. Let's deep dive into the confusion matrix for understanding this better.

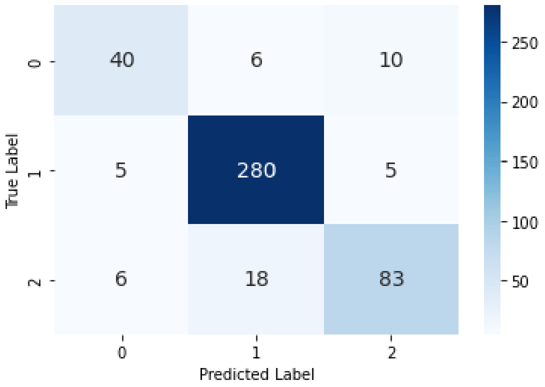


Figure 21: Confusion Matrix – Bi-LSTM with word2vec

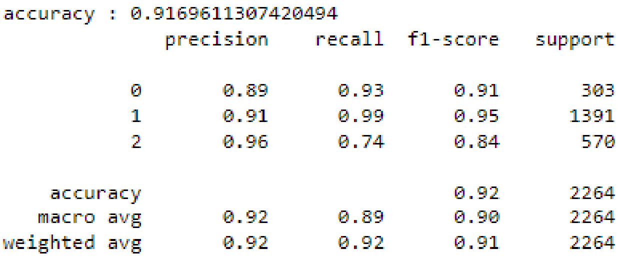
Confusion Matrix Insights:

Upon examining the confusion matrix, we observe the following trends:

- **Class 0 (Negative)**: Out of the total predictions, 40 instances are correctly classified as negative. However, there is a misclassification with 6 instances predicted as Neutral and 10 as Positive. Almost 20% of the negatives are predicted as positive, confusing with 2 opposite classes will lead us to a high risk of misinterpretation of the anlysis.
- **Class 1 (Neutral)**: The model demonstrates strong predictive power for neutral sentiments, correctly classifying 280 instances. But it's worth noting that 5 instances were misclassified as Negative and another 5 as Positive. Given the predominant representation of this class, the model seems inclined to predict Neutral more often, which can be attributed to the class imbalance.
- **Class 2 (Positive)**: 83 instances are accurately tagged as positive sentiments. However, there are 6 instances mistaken as Negative and 18 as Neutral.This shows that the models is slightly confusing the positive ones as neutral.

4.1.2 Senti Analysis 2 - FinBERT on financial_phrasebank dataset

The sentiment analysis conducted in this section utilizes FinBERT. The chosen dataset for this analysis is the financial_phrasebank, which primarily focuses on financial sentiments.



```
accuracy : 0.9169611307420494
              precision    recall  f1-score   support

           0       0.89      0.93      0.91       303
           1       0.91      0.99      0.95      1391
           2       0.96      0.74      0.84       570

    accuracy                           0.92      2264
   macro avg       0.92      0.89      0.90      2264
weighted avg       0.92      0.92      0.91      2264
```

Figure 22: Classification Report - FinBERT

The dataset categorizes sentiments into three distinct classes, 0 class - Negative, 1 class - Neutral, 2 class - Positive.

Overall Accuracy: The model showcases a prediction accuracy of 91.6%.

Precision, Recall, F1-score, and NPV:

- For Class 0: Precision stands at 0.89, recall at 0.93, and F1-score at 0.91
- Class 1 impresses with a precision of 0.91, recall of 0.99, and an F1-score of 0.95

- Class 2 exhibits a precision of 0.96, recall of 0.74, and an F1-score of 0.84.

The weighted average are precision – 0.92, recall – 0.92, and F1-score – 0.91.

Actual vs. Predicted Sentiment Distributions:

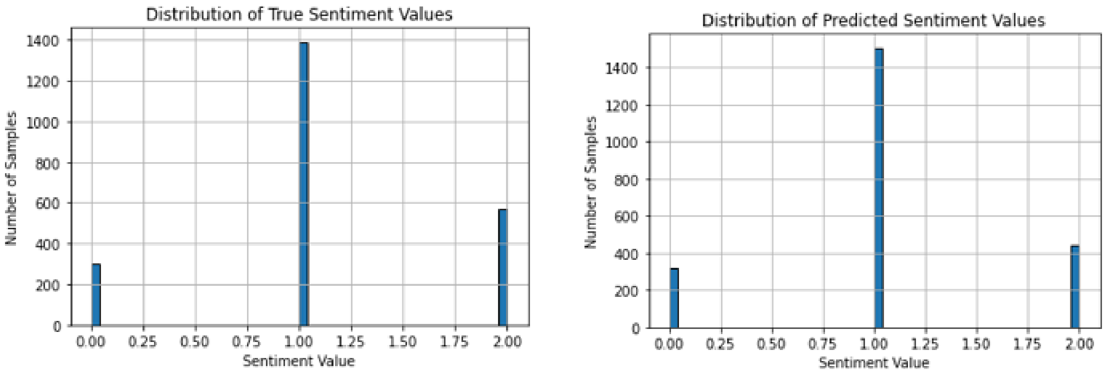
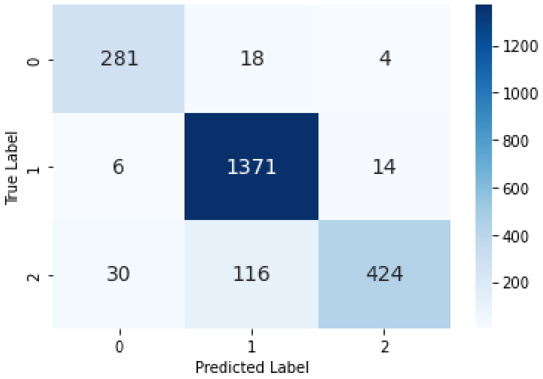


Figure 23: : Class Distribution - True sentiments (left) and Predicted Sentiments (right) - FinBERT

The above class distribution of true vs predicted sentiments visually clarifies that negative predictions are proportional, whereas the positive prediction are slightly lower than true and that was compensated in the neutral class being over predicted. As evident in the above class distribution, there is a class imbalance inclined towards the neutral class, and our target is to predict the negatives and positives mainly. Let's deep dive into the confusion matrix for understanding this better.



	0	1	2
0	281	18	4
1	6	1371	14
2	30	116	424

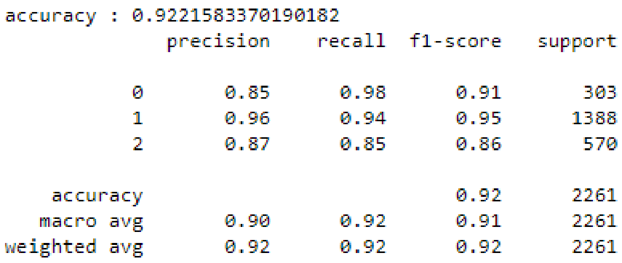
Figure 24: Confusion Matrix - FinBERT

Confusion Matrix Insights: Upon examining the confusion matrix, we can discern the following trends:

- **Class 0 (Negative):** Out of the total predictions, 281 instances are correctly identified as negative. Whereas 18 instances predicted as Neutral and 4 as Positive. This says that the model has learnt better to predict negative class.
- **Class 1 (Neutral):** The model exhibits a remarkable capability for discerning neutral sentiments, accurately classifying 1371 instances. However, 6 instances were marked as Negative and 14 as Positive. Due to the class skewness, prediction of this class is well trained to the model.
- **Class 2 (Positive):** 424 instances are precisely labelled as positive sentiments. Whereas, 30 instances misclassified as Negative and 116 as Neutral. This indicates that the model is lagging in predicting the positives and misinterpreting it with neutrals and negatives.

4.1.3 Senti Analysis 3 - LLM on financial_phrasebank dataset

The sentiment analysis conducted in this section utilizes LLM-ChatGPT3.5. The chosen dataset for this analysis is the financial_phrasebank, which primarily focuses on financial sentiments.



```
accuracy : 0.9221583370190182
              precision    recall  f1-score   support

           0       0.85      0.98      0.91       303
           1       0.96      0.94      0.95      1388
           2       0.87      0.85      0.86       570

    accuracy                           0.92      2261
   macro avg       0.90      0.92      0.91      2261
weighted avg       0.92      0.92      0.92      2261
```

Figure 25: Classification Report – ChatGPT3.5

The dataset categorizes sentiments into three distinct classes, 0 class - Negative, 1 class - Neutral, 2 class - Positive.

Overall Accuracy: The model showcases a prediction accuracy of 92.2%.

Precision, Recall, F1-score, and NPV:

- For Class 0: Precision stands at 0.85, recall at 0.98, and F1-score at 0.91
- Class 1 impresses with a precision of 0.96, recall of 0.94, and an F1-score of 0.95
- Class 2 exhibits a precision of 0.87, recall of 0.85, and an F1-score of 0.86

The weighted average across precision, recall, and F1-score is consistent at 0.92

Actual vs. Predicted Sentiment Distributions:

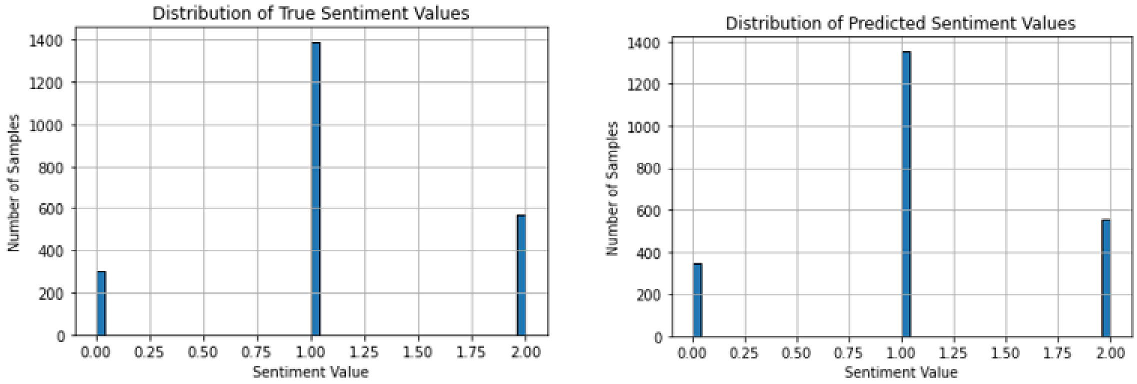


Figure 26: Class Distribution - True sentiments (left) and Predicted Sentiments (right) – ChatGPT3.5

The above class distribution of true vs predicted sentiments visually clarifies all the classes of the prediction are almost proportional with the true values. As evident in the above class distribution, there is a class imbalance inclined towards the neutral class, and our target is to predict the negatives and positives mainly. Let's deep dive into the confusion matrix for understanding this better.

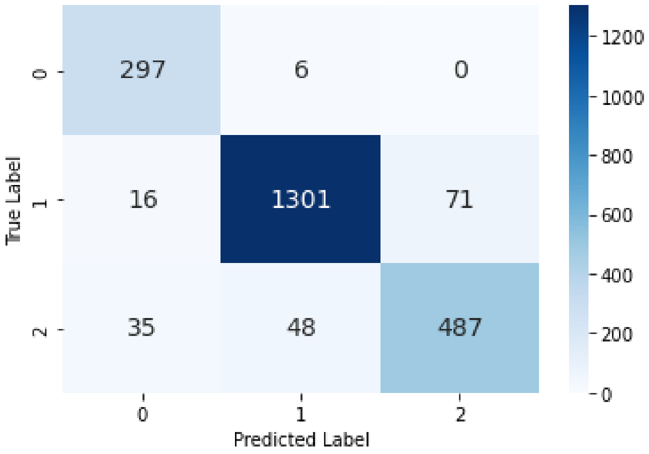


Figure 27: Confusion Matrix - ChatGPT3.5

Confusion Matrix Insights:

Upon digging into the confusion matrix, we can identify the following patterns:

- **Class 0 (Negative)**: From the aggregate predictions, 297 instances are correctly pinpointed as negative. Only 6 instances are projected as Neutral and 0 as Positive. This shows that the model has predicted almost all the negatives in a better way.

- **Class 1 (Neutral)**: The model continues to showcase robust discernment for the neutral sentiments, marking 1301 instances accurately. Only 16 instances were marked Negative and 71 as Positive. The model has a slight confusion between neutral and positive.
- **Class 2 (Positive)**: 487 instances are spot-on labelled as positive sentiments. However, there is a variance with 48 instances wrongly classified as Neutral and 35 as Negative. The model lags in predicting positives correctly, it is misinterpretation on neutral class is considerable but its is miss interpreting nearly 7 to 8% as negative, which is a threat. Need to give some more positive and negative examples to the prompt.

This 71 true neutral wrongly predicted as positive and, 48 true positive wrongly predicted as neutral has nullified and couldn't recognize visually in the class distribution.

4.1.4 Senti Analysis 1.1, 2.1, and 3.1 all models on New York Times dataset

The sentiment analysis conducted in this section utilizes a Bi-LSTM (Bidirectional Long Short-Term Memory) model integrated with Word2Vec embeddings. The chosen dataset for this analysis is the **non-labelled NYTimes dataset**. Since there are no true labels to compare the predictions, we do not have classification report and confusion matrix. So, we visually analyse the class distribution of the predictions and a chart where the cumulative sentiments were matched with the close price. **These two methods of validation are not empirical and conclusive**, that's why we have taken the sentiments of all three model into the LSTM models for predicting the stock prices where we can compare the results of stock price prediction and say which of the NLP model has given higher prediction accuracy. Just we can see the overview and the correlation of sentiments with close price over time.

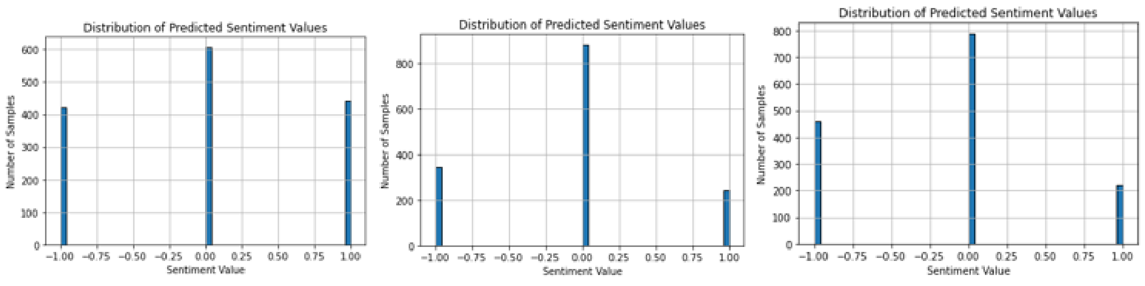


Figure 28: Class distribution - Bi-LSTM with word2vec (left), FinBERT(center), LLM-ChatGPT3.5(right)

In Bi-LSTM model with word2vec the sentiment predictions of the news is not skewed towards the neutral class, and the positive and negative are in the range of 400 instances /days and neutral at 600 instances/days. On the other hand, FinBERT model predicted almost 400 as negative, 850 as neutral and 225 as positive classes and whereas ChatGPT3.5 model is with 450 as negative, 800 as neutral and 200 as positive classes. If you note one thing the proportions of the FinBERT and ChatGPT models are almost same, and different when compared to the Bi-LSTM model. This could be because of the nature of the corpus and the models used in it. word2vec is good in predicting general news statements and FinBERT is specially tuned for the for the financial statements, and since ChatGPT prompts are tuned in line with the financial_phrasebank dataset which is a financial corpus. The FinBERT and ChatGPT might behave in one way and whereas word2vec in a different way. But with this we cannot conclude which prediction is better.

Cumulative Sentiment vs Close Price:

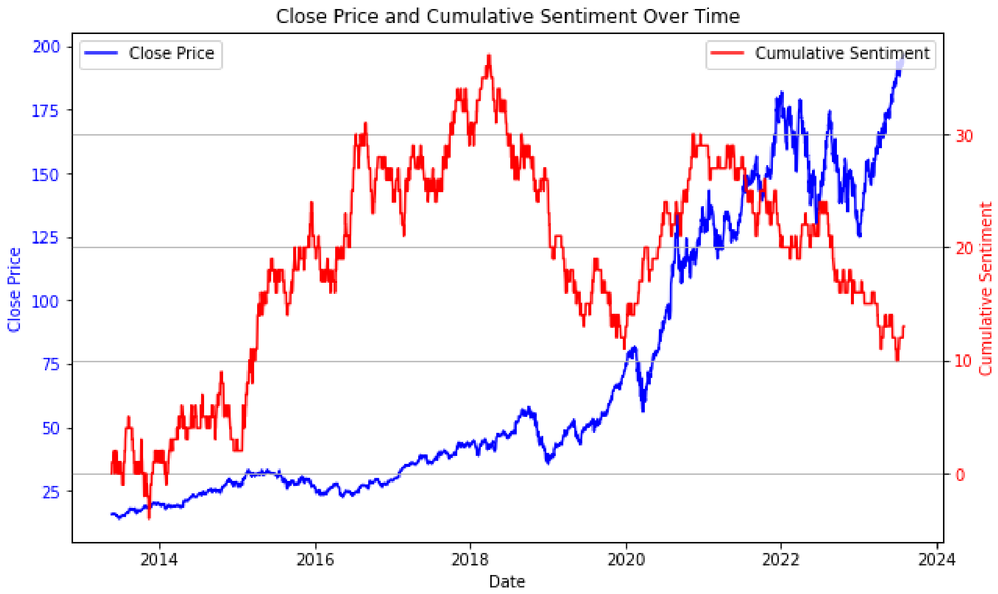


Figure 29: Close Price vs Cumulative sentiment over time - Bi-LSTM with word2vec

In this the graph, the cumulative sentiments seems to be getting the overall trend of the close price. According to this graph, in the first 5 years the cum sentiments had gone to the peak whereas the stock prices were very gradually grown to almost 25% of its pricing today. In the last five years the close price has drastically gone high with fluctuations, here the sentiments were falling and stagnating.

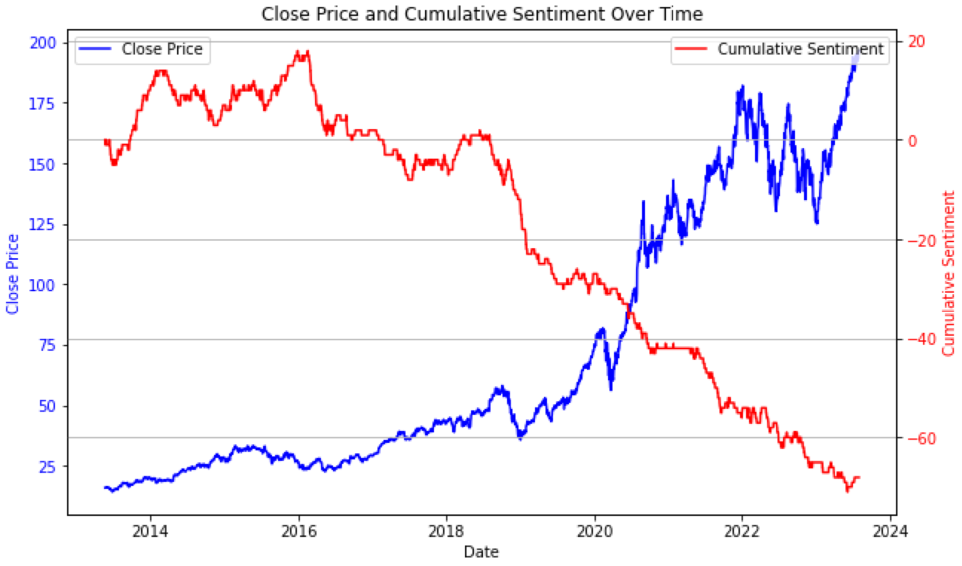


Figure 30: Close Price vs Cumulative sentiment over time – FinBERT

In FinBERT model, for the first five years the sentiments and the close price has maintained the same trend of gradual raise. But in the second half the close price has drastically gone high and the sentiments has fallen down, which means that continuous negative news according to the model.

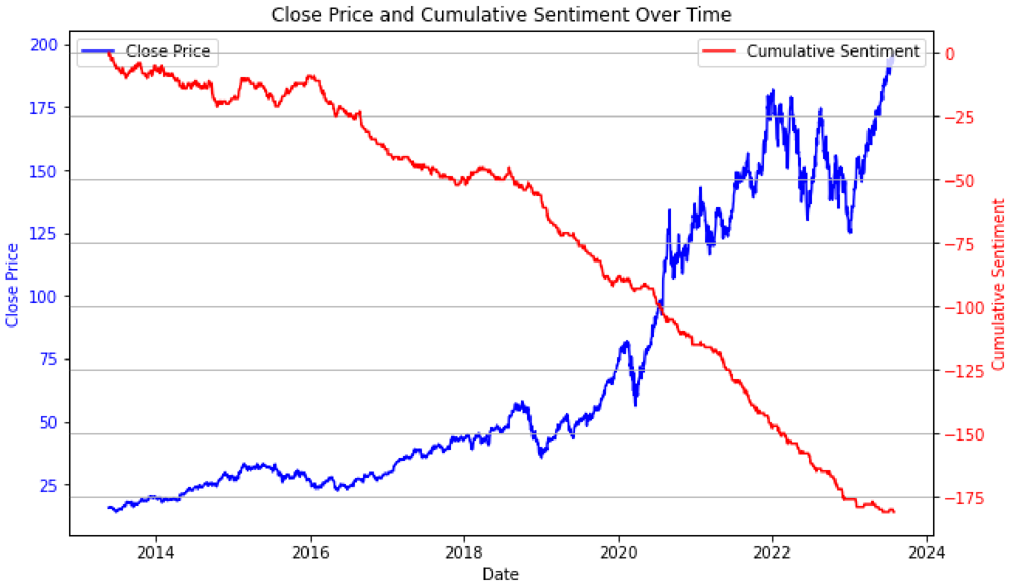


Figure 31: Close Price vs Cumulative sentiment over time - ChatGPT3.5

In the ChatGPT model, the sentiment started falling from the start, for the first few years it was gradual fall then it’s a drastic fall.

With this visual analysis, and with a fact that NYTimes is a general statement corpus, it seems that Bi-LSTM with word2vec might have predicted well. **But nothing is conclusive**.

Let us use the sentiments of these three models separately into the main task LSTM model and access through the results of the stock price prediction.

4.2 Stock Price Prediction Model's Performance

In this section, the outputs of the baseline model (RF Regressor) on Dataset_A and then the proposed model (LSTM) with all the 6 different datasets (A, B, C, D1, D2, and D3) are plotted here. The output contains,

- Expanding window cross validation performance graph

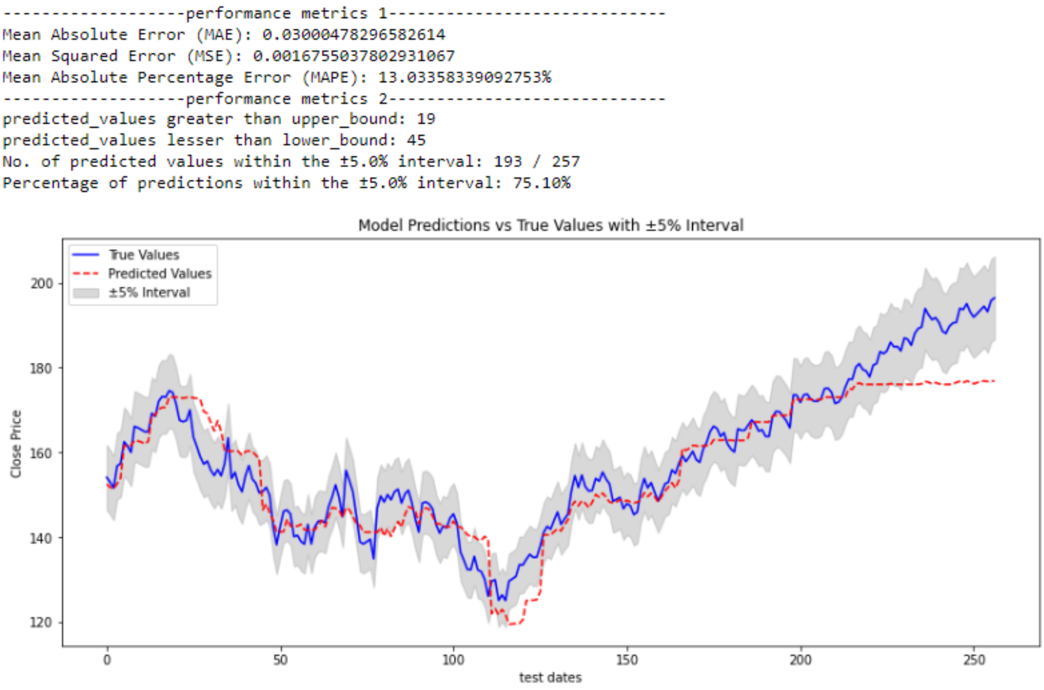
- Model loss graph – training vs validation losses over epoch,

- Performance metrics 1 & 2,

- Model predictions vs true values with +/- 5% interval

4.2.1 Baseline Model Performance: Output 1

The outcomes of the Random Forest Regressor model on the Dataset_A (Historical Price + Technical Indicators) are as follows,



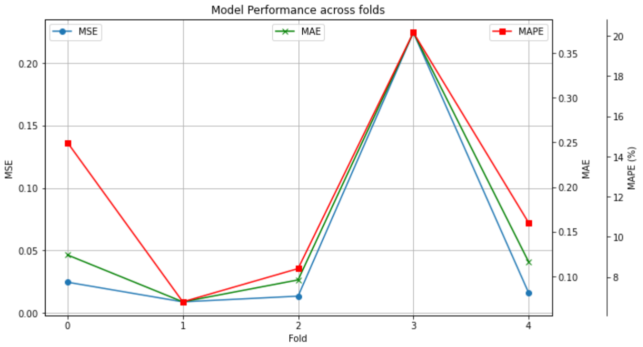
```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.03000478296582614
Mean Squared Error (MSE): 0.0016755037802931067
Mean Absolute Percentage Error (MAPE): 13.03358339092753%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 19
predicted_values lesser than lower_bound: 45
No. of predicted values within the ±5.0% interval: 193 / 257
Percentage of predictions within the ±5.0% interval: 75.10%
```

Figure 32: Experiment 1 – Model Prediction vs True Value

In the experiment 1, the model attained a MAPE of 13% and 75% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, it is noteworthy that the prediction line exhibits flattening after day 200 or during the final 50 days.

4.2.2 LSTM with Dataset_A: Output 2 & 3

The outcomes of the LSTM model and Hyper parameter optimized best model on the Dataset_A (Historical Price + Technical Indicators) are as follows,



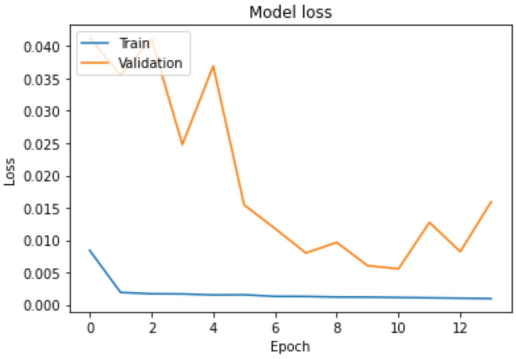
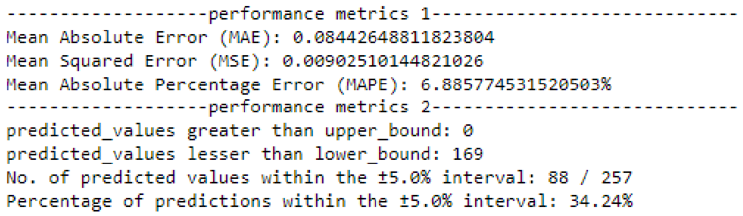


Figure 33: Exp-2 Expanding Window CV performance

Figure 34: Exp-2 Train loss vs Val loss

The above chart on the left explains the performance of the model across the folds of the Expanding window cross validation. The chart on the right shows the train vs validation loss, in which the validation loss has converged from 10th epoch and the training has been stopped to avoid overfitting.



```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.08442648811823804
Mean Squared Error (MSE): 0.00902510144821026
Mean Absolute Percentage Error (MAPE): 6.885774531520503%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 0
predicted_values lesser than lower_bound: 169
No. of predicted values within the ±5.0% interval: 88 / 257
Percentage of predictions within the ±5.0% interval: 34.24%
```

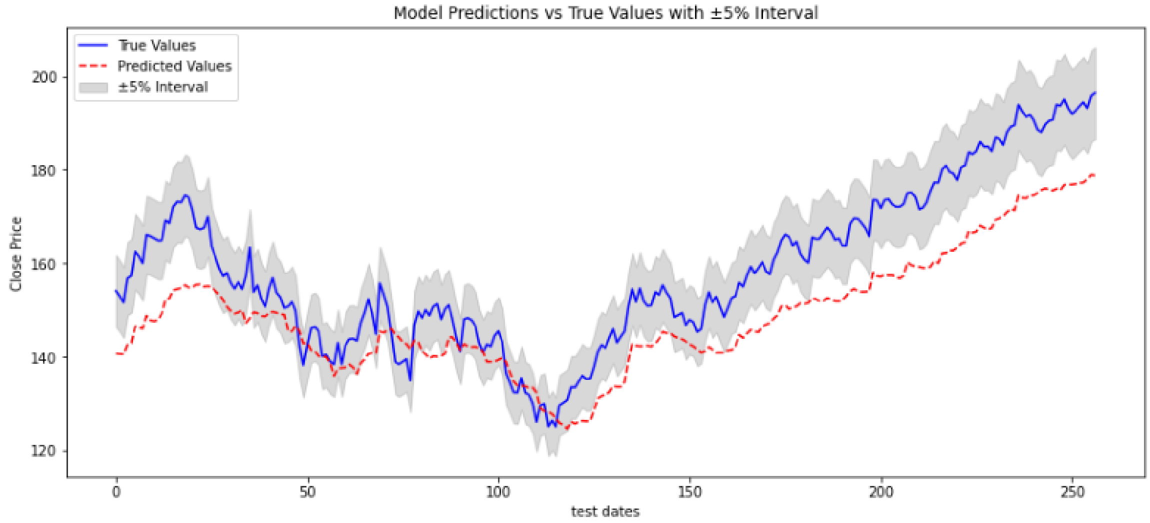
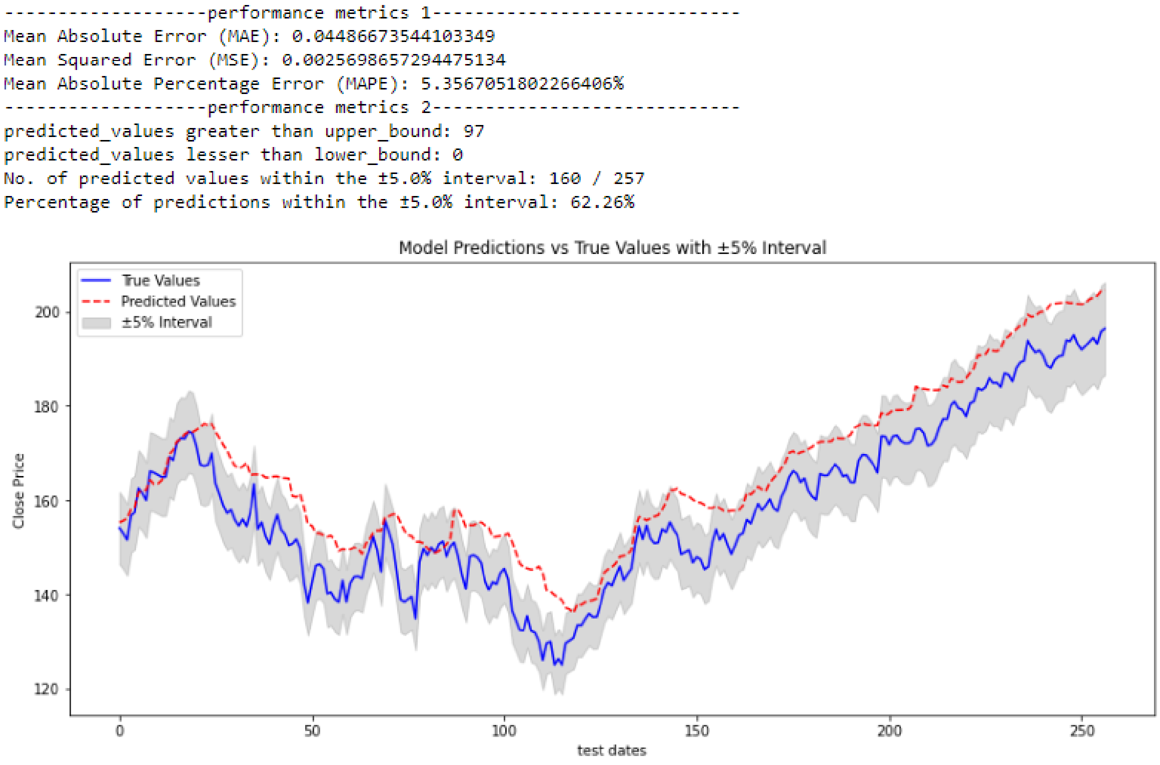


Figure 35: Experiment 2 – Model Prediction vs True Value

In the experiment 2, the model attained a MAPE of 6.8% and 34% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the prediction line on the last steep is diverging from the actual values.



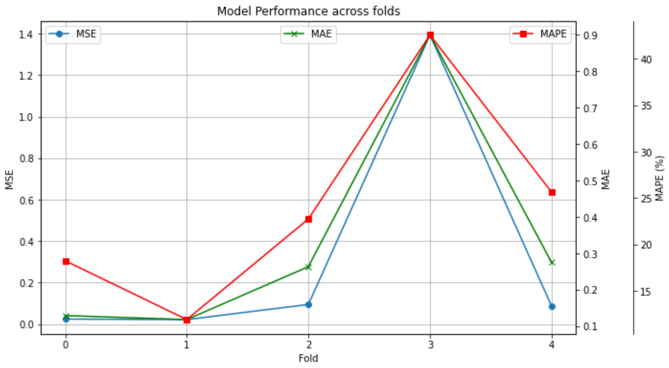
```
-------------------performance metrics 1-----------------------------
Mean Absolute Error (MAE): 0.04486673544103349
Mean Squared Error (MSE): 0.0025698657294475134
Mean Absolute Percentage Error (MAPE): 5.3567051802266406%
-------------------performance metrics 2-----------------------------
predicted_values greater than upper_bound: 97
predicted_values lesser than lower_bound: 0
No. of predicted values within the ±5.0% interval: 160 / 257
Percentage of predictions within the ±5.0% interval: 62.26%
```

Figure 36: Experiment 3 – Model Prediction vs True Value

In the experiment 3, the model attained a MAPE of 5.3% and 62% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the prediction line consistently surpasses the true value, signalling a bias towards the overestimation.

4.2.3 LSTM with Dataset_B: Output 4 & 5

The outcomes of the LSTM model and Hyper parameter optimized best model on the Dataset_B (Historical Price + Technical Indicators + Financial Indicators) are as follows,



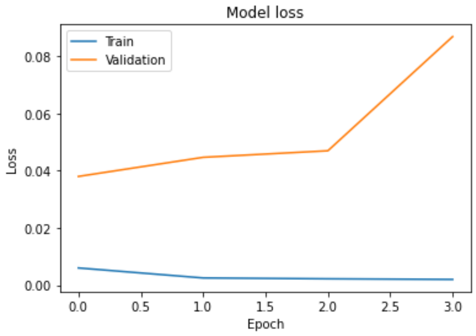
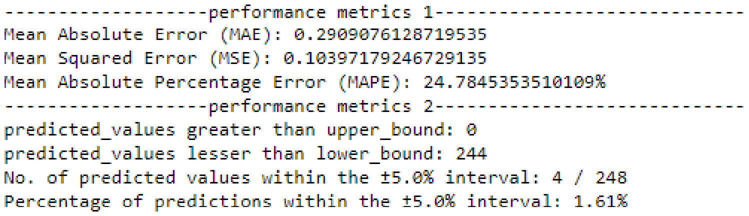


Figure 37: Exp-4 Expanding Window CV performance

Figure 38: Exp-4 Train loss vs Val loss

The above chart on the left explains the performance of the model across the folds of the Expanding window cross validation. The chart on the right shows the train vs validation loss, in which the validation loss has started rising from the 1st epoch and it kept on increasing.



```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.2909076128719535
Mean Squared Error (MSE): 0.10397179246729135
Mean Absolute Percentage Error (MAPE): 24.7845353510109%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 0
predicted_values lesser than lower_bound: 244
No. of predicted values within the ±5.0% interval: 4 / 248
Percentage of predictions within the ±5.0% interval: 1.61%
```

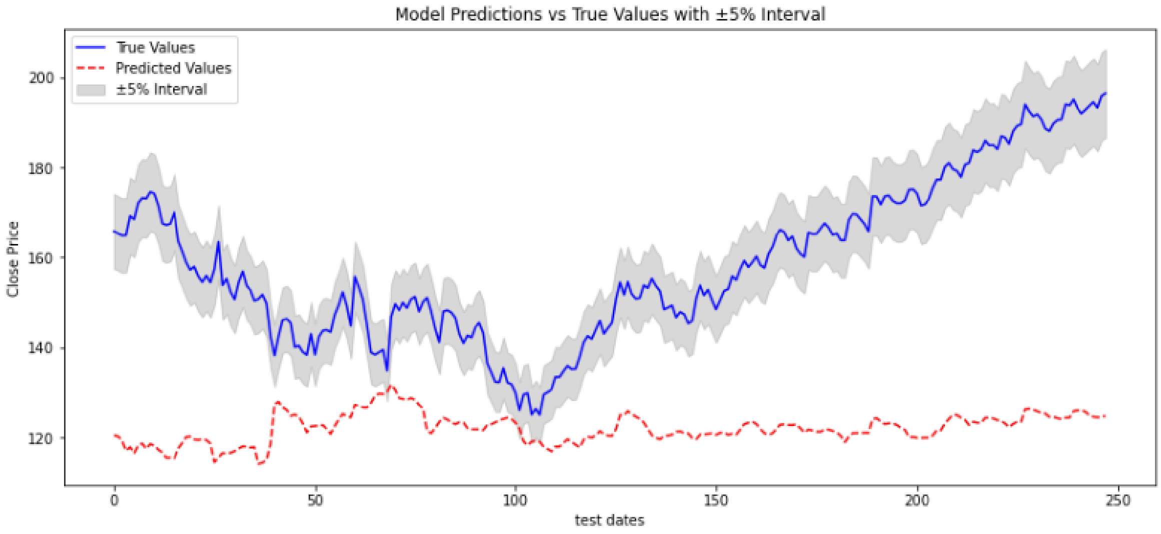
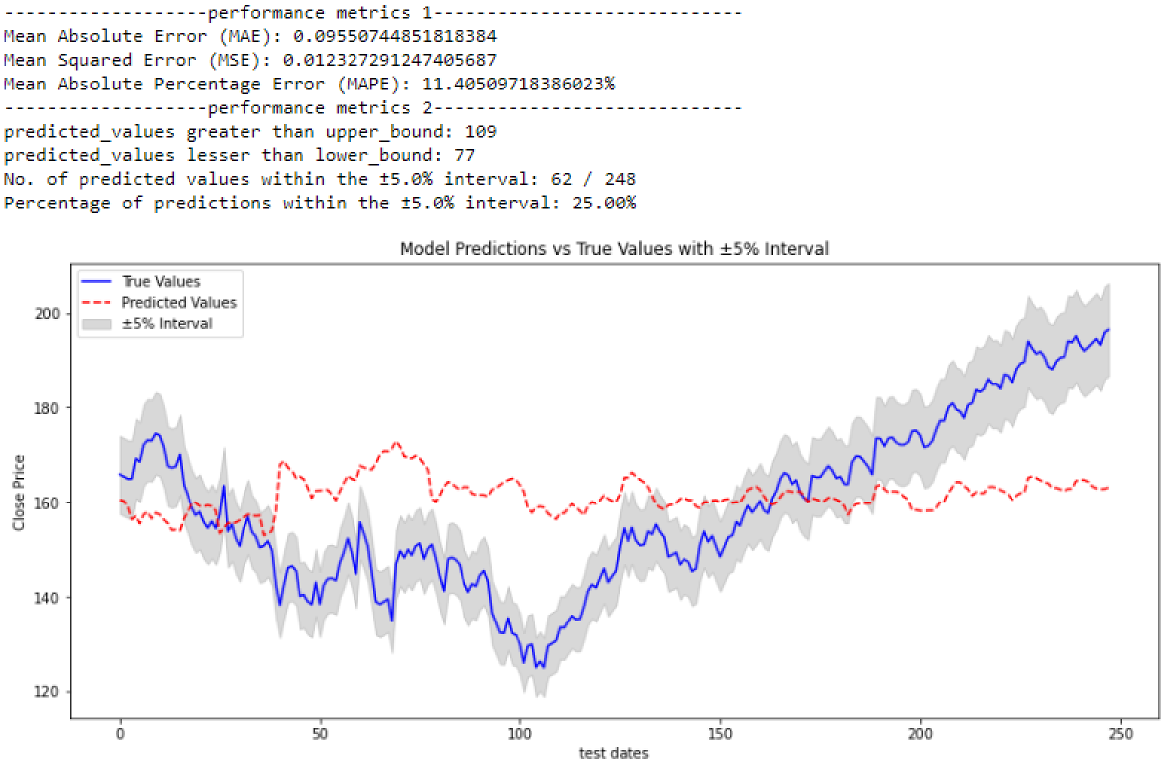


Figure 39: Experiment 4 – Model Prediction vs True Value

In the experiment 4, the model attained a MAPE of 24.7% and only 1.6% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the prediction line has almost flattened through the test set and does not even get the trend of the actuals.



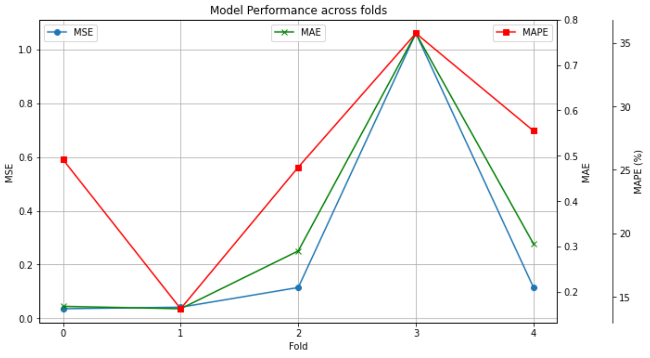
```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.09550744851818384
Mean Squared Error (MSE): 0.012327291247405687
Mean Absolute Percentage Error (MAPE): 11.40509718386023%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 109
predicted_values lesser than lower_bound: 77
No. of predicted values within the ±5.0% interval: 62 / 248
Percentage of predictions within the ±5.0% interval: 25.00%
```

Figure 40: Experiment 5 – Model Prediction vs True Value

In the experiment 5, the model attained a MAPE of 11.4% and 25% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the prediction line has almost flattened through the test set and does not even get the trend of the actuals, but the hyper parameter optimization has shifted the base to 160 whereas in the experiment 4 it was around 120.

4.2.4 LSTM with Dataset_C: Output 6 & 7

The outcomes of the LSTM model and Hyper parameter optimized best model on the Dataset_C (Historical Price + Technical Indicators + Financial Indicators + Financial News) are as follows,



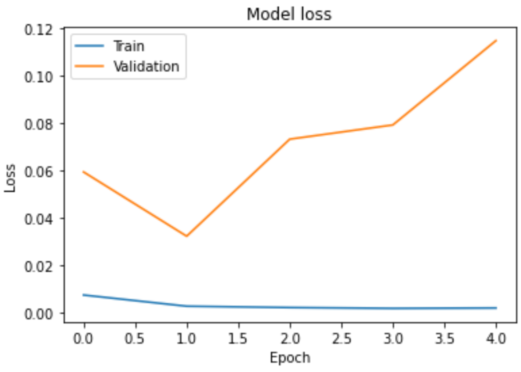
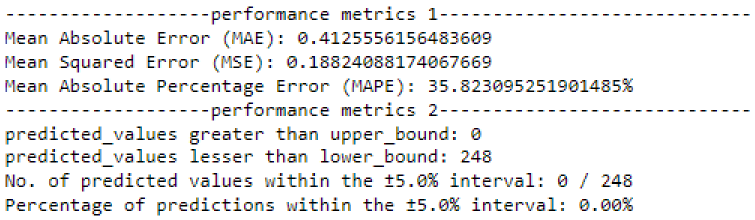


Figure 41: Exp-6 Expanding Window CV performance

Figure 42: Exp-6 Train loss vs Val loss

Similar to the earlier experiment 4 & 5, the above chart on the left explains the performance of the model across the folds of the Expanding window cross validation. The chart on the right shows the train vs validation loss, in which the validation loss has started rising from the 1st epoch and it kept on increasing.



```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.4125556156483609
Mean Squared Error (MSE): 0.18824088174067669
Mean Absolute Percentage Error (MAPE): 35.823095251901485%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 0
predicted_values lesser than lower_bound: 248
No. of predicted values within the ±5.0% interval: 0 / 248
Percentage of predictions within the ±5.0% interval: 0.00%
```

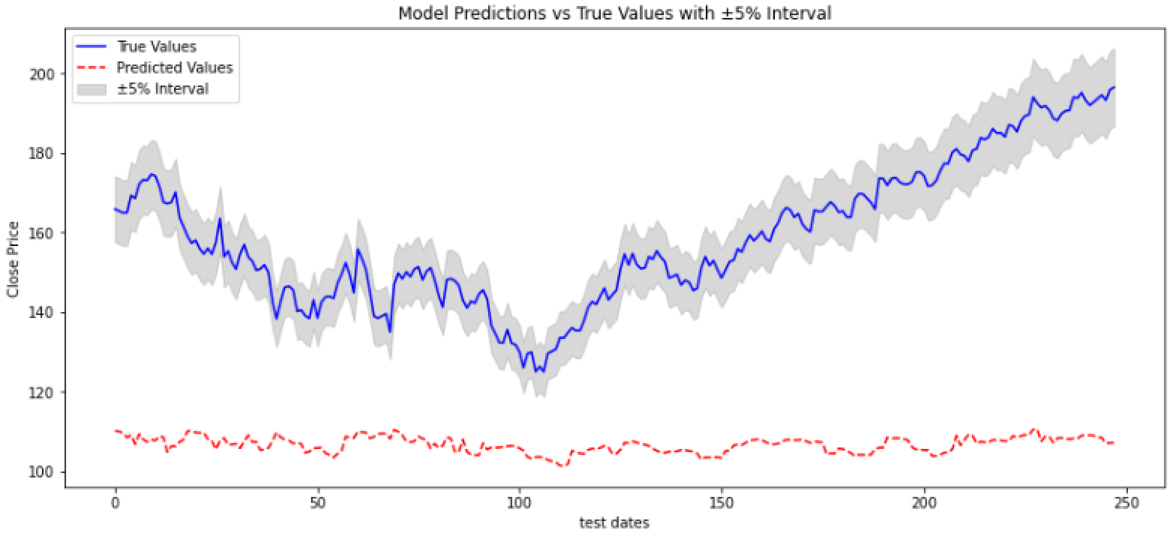
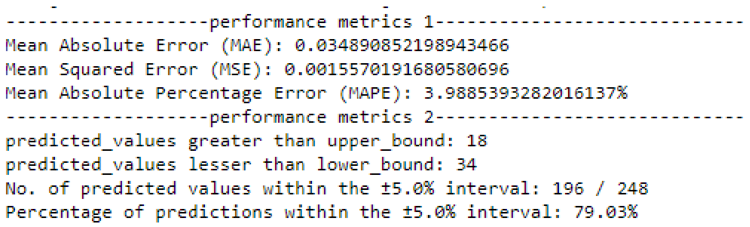


Figure 43: Experiment 6 – Model Prediction vs True Value

In the experiment 6, the model attained a MAPE of 35.8% and none of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the prediction line has almost flattened through the test set and does not even get the trend of the actuals and having a base of 110 dollars.



```
-------------------performance metrics 1---------------------------
Mean Absolute Error (MAE): 0.034890852198943466
Mean Squared Error (MSE): 0.0015570191680580696
Mean Absolute Percentage Error (MAPE): 3.9885393282016137%
-------------------performance metrics 2---------------------------
predicted_values greater than upper_bound: 18
predicted_values lesser than lower_bound: 34
No. of predicted values within the ±5.0% interval: 196 / 248
Percentage of predictions within the ±5.0% interval: 79.03%
```

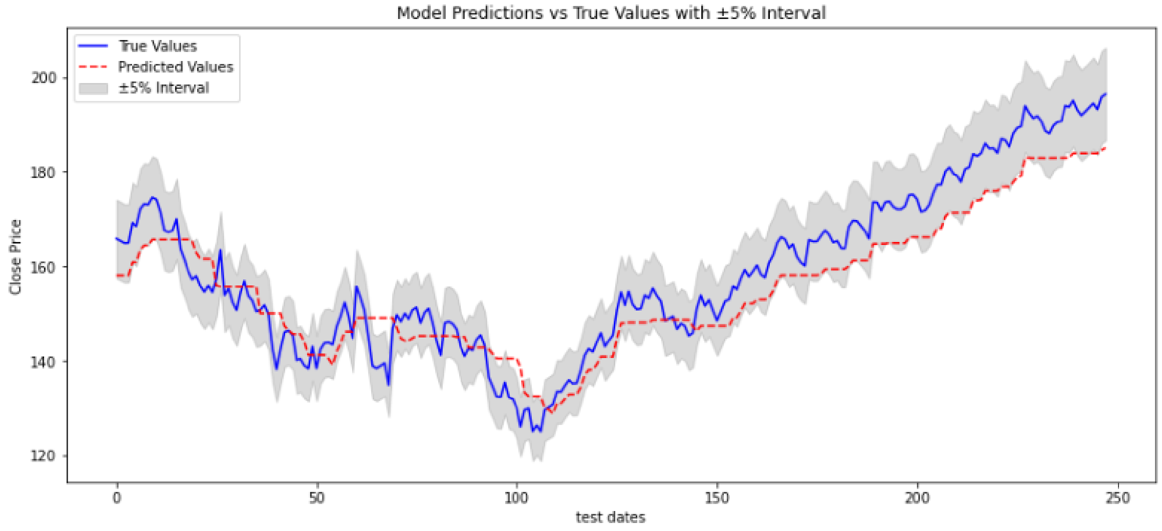
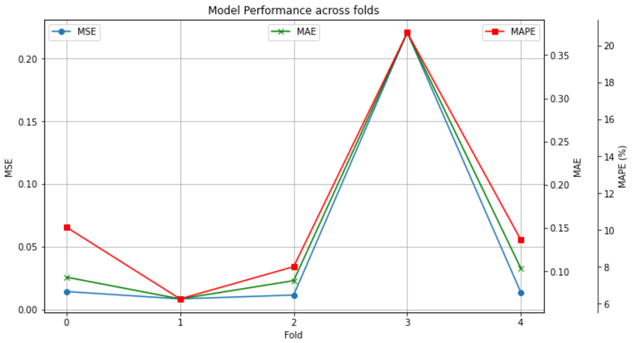


Figure 44: Experiment 7 – Model Prediction vs True Value

In the experiment 7, the model attained a drastic rise from the non-hyper parameter tunned model (i.e. experiment 6) to a MAPE of 3.9% and 79% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the prediction line has got the trends of the actuals, but not getting the highs and lows of the true line.

4.2.5 LSTM with Dataset_D1: Output 8 & 9

The outcomes of the LSTM model and Hyper parameter optimized best model on the Dataset_D1 (Historical Price + Technical Indicators + Financial News – Bi-LSTM with word2vec) are as follows,



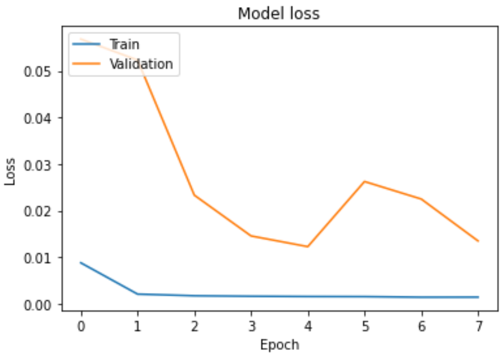
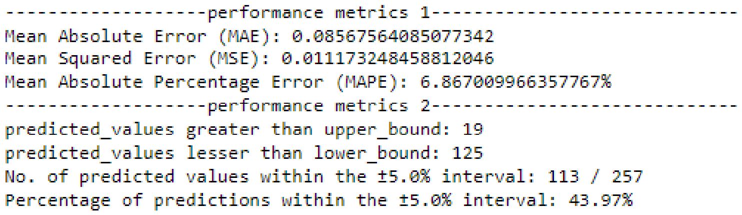


Figure 45: Exp-8 Expanding Window CV performance

Figure 46: Exp-8 Train loss vs Val loss

In this experiment also, the above chart on the left explains the performance of the model across the folds of the Expanding window cross validation and having the same pattern. The chart on the right shows the train vs validation loss, in which the validation loss has started converging from 4th epoch.


```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.08567564085077342
Mean Squared Error (MSE): 0.011173248458812046
Mean Absolute Percentage Error (MAPE): 6.867009966357767%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 19
predicted_values lesser than lower_bound: 125
No. of predicted values within the ±5.0% interval: 113 / 257
Percentage of predictions within the ±5.0% interval: 43.97%
```

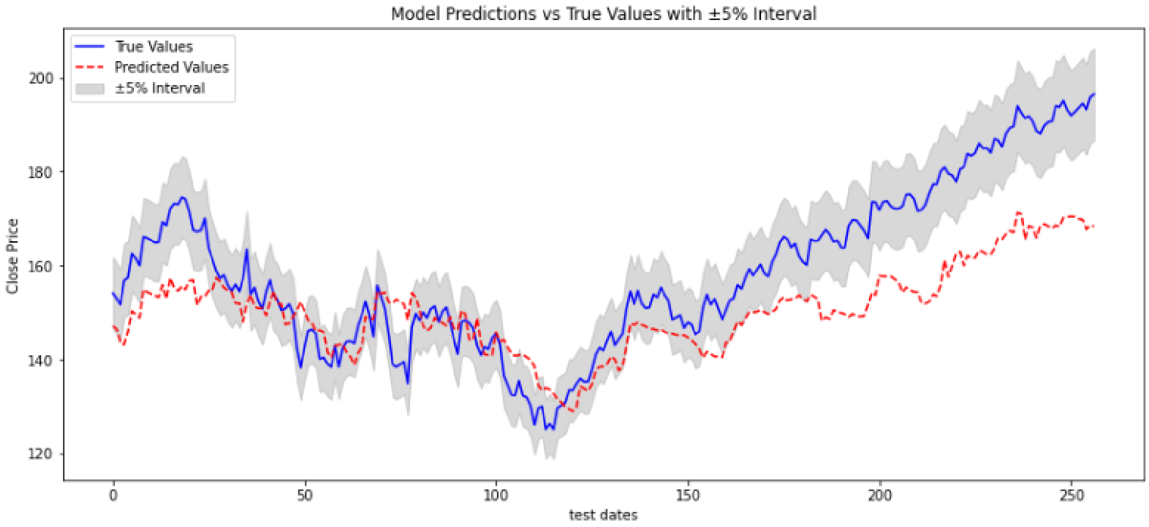
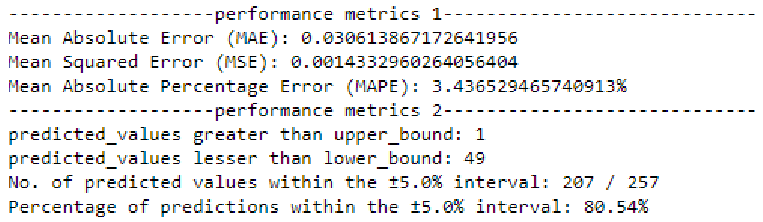


Figure 47: Experiment 8 – Model Prediction vs True Value

In the experiment 8, the model attained a MAPE of 6.8% and 43.9% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the prediction line has got the trends of the actuals, and even got the highs and lows of the true line to some extent, but the second half of the prediction is offset.



```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.030613867172641956
Mean Squared Error (MSE): 0.0014332960264056404
Mean Absolute Percentage Error (MAPE): 3.436529465740913%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 1
predicted_values lesser than lower_bound: 49
No. of predicted values within the ±5.0% interval: 207 / 257
Percentage of predictions within the ±5.0% interval: 80.54%
```

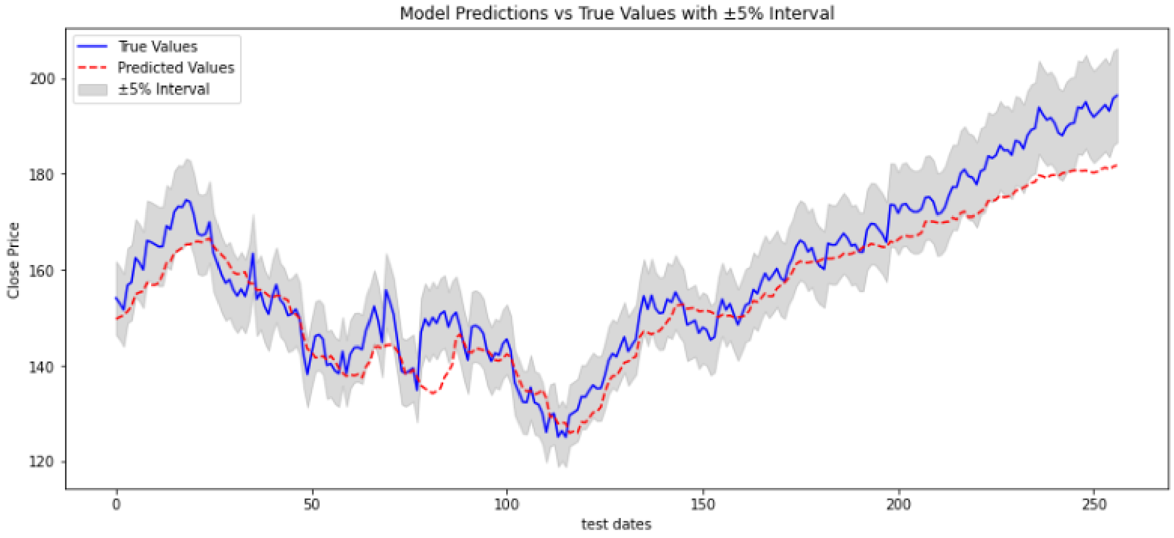


Figure 48: Experiment 9 – Model Prediction vs True Value

In the experiment 9, the model attained a MAPE of 3.43% and 80.5% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the prediction line has got the trends of the actuals, and with the hyper parameter optimization the best model got trained better in such a way that the prediction line in the last quarter of the graph improved to some extent.

4.2.6 LSTM with Dataset_D2: Output 10 & 11

The outcomes of the LSTM model and Hyper parameter optimized best model on the Dataset_D2 (Historical Price + Technical Indicators + Financial News – FinBERT) are as follows,

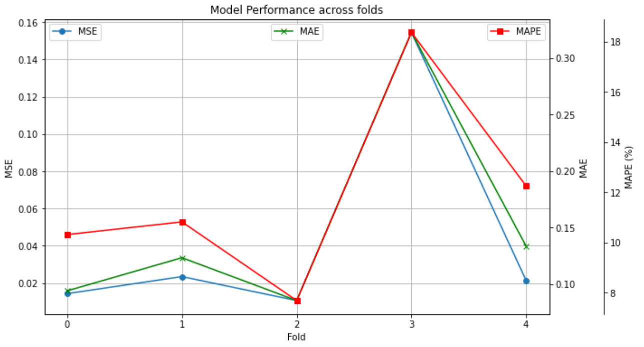


Figure 49: Exp-10 Expanding Window CV performance

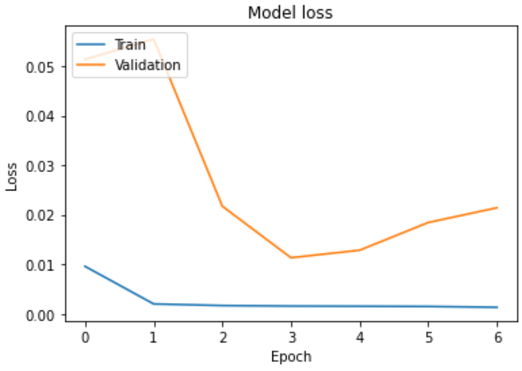
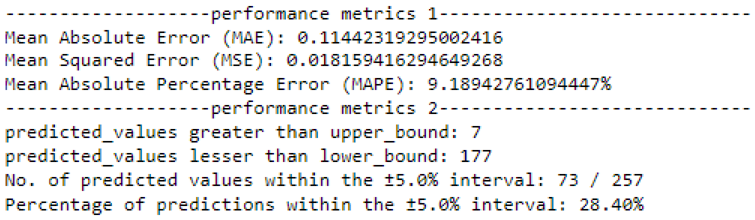


Figure 50: Exp-10 Train loss vs Val loss

In this experiment also, the above chart on the left explains the performance of the model across the folds of the Expanding window cross validation and having the same pattern. The chart on the right shows the train vs validation loss, in which the validation loss has started converging from 3rd epoch.



```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.11442319295002416
Mean Squared Error (MSE): 0.018159416294649268
Mean Absolute Percentage Error (MAPE): 9.18942761094447%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 7
predicted_values lesser than lower_bound: 177
No. of predicted values within the ±5.0% interval: 73 / 257
Percentage of predictions within the ±5.0% interval: 28.40%
```

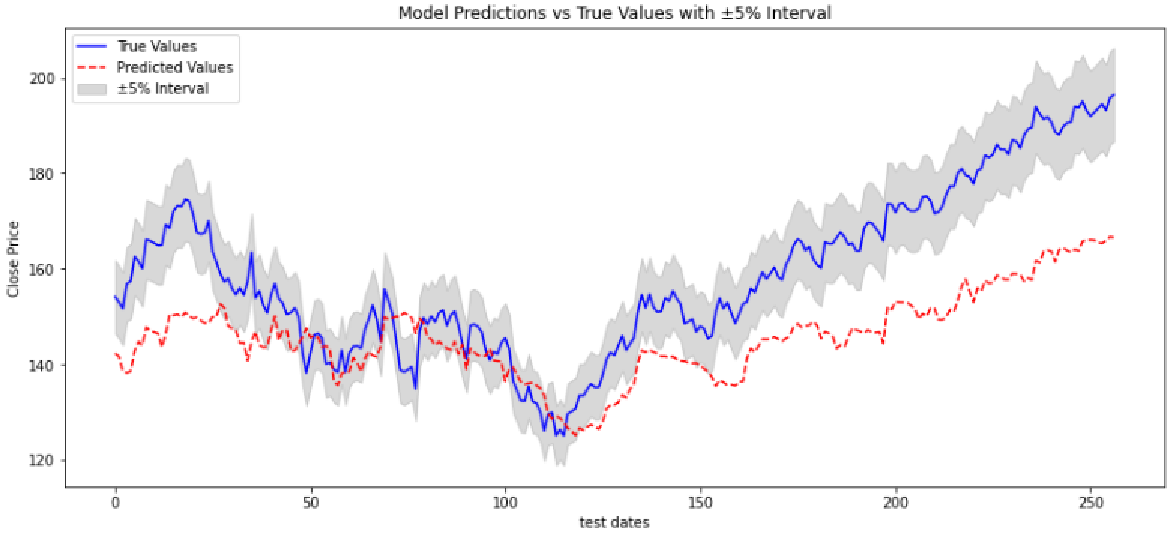
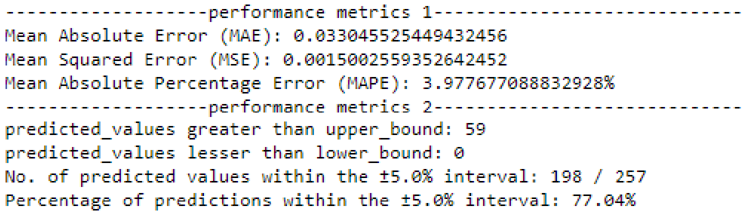


Figure 51: Experiment 10 – Model Prediction vs True Value

In the experiment 9, the model attained a MAPE of 9.1 % and 28.4% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the first half of the test set was predicted well, whereas in the second half the prediction line is diverging from the true line.



```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.033045525449432456
Mean Squared Error (MSE): 0.0015002559352642452
Mean Absolute Percentage Error (MAPE): 3.977677088832928%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 59
predicted_values lesser than lower_bound: 0
No. of predicted values within the ±5.0% interval: 198 / 257
Percentage of predictions within the ±5.0% interval: 77.04%
```

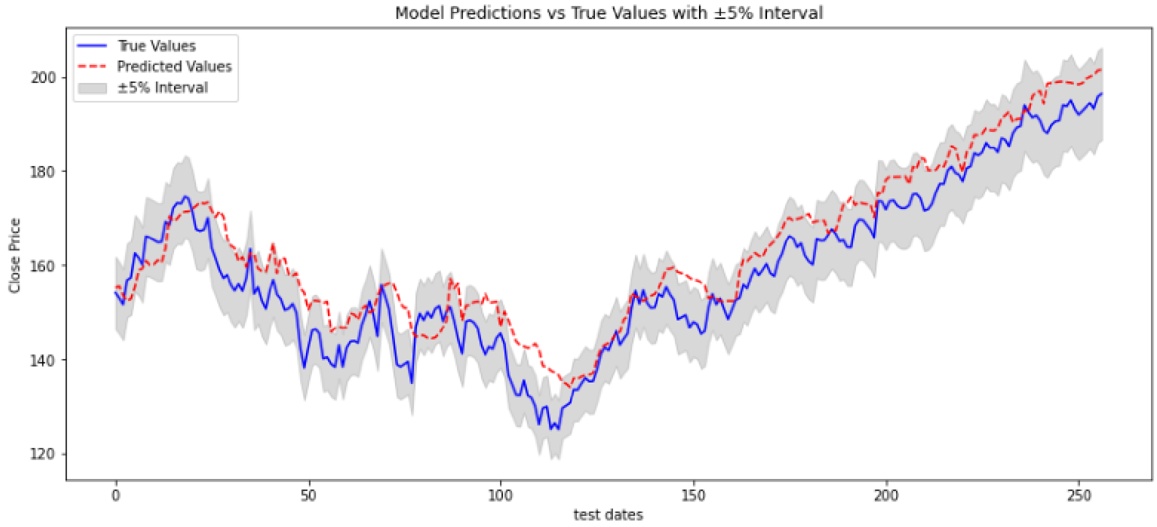
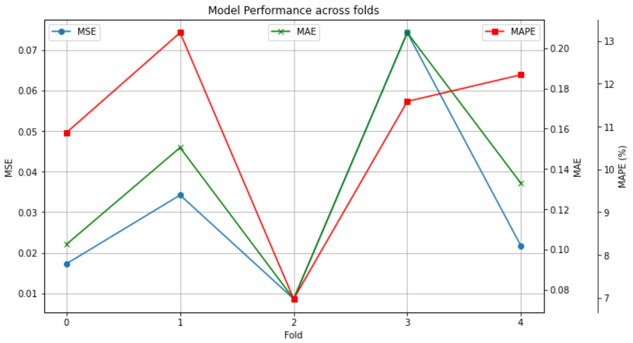


Figure 52: Experiment 11 – Model Prediction vs True Value

In the experiment 10, the best model of hyper parameter optimization has given a better result of MAPE – 3.9% and 77% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the overall trend of prediction is good, one notable point is the prediction line has captured the highs and lows of the true line better than other models and its slightly biased with over estimation.

4.2.7 LSTM with Dataset_D3: Output 12 & 13

The outcomes of the LSTM model and Hyper parameter optimized best model on the Dataset_D3 (Historical Price + Technical Indicators + Financial News – LLM-ChatGPT3.5) are as follows,



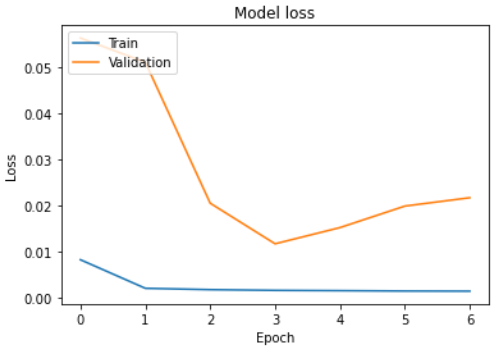
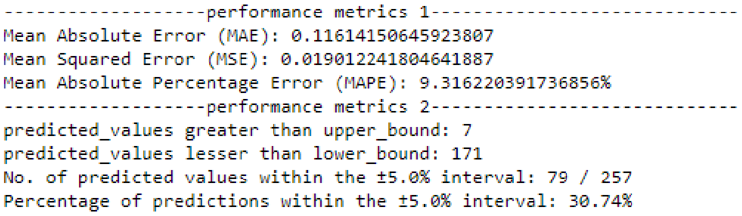


Figure 53: Exp-12 Expanding Window CV performance

Figure 54: Exp-12 Train loss vs Val loss



```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.11614150645923807
Mean Squared Error (MSE): 0.019012241804641887
Mean Absolute Percentage Error (MAPE): 9.316220391736856%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 7
predicted_values lesser than lower_bound: 171
No. of predicted values within the ±5.0% interval: 79 / 257
Percentage of predictions within the ±5.0% interval: 30.74%
```

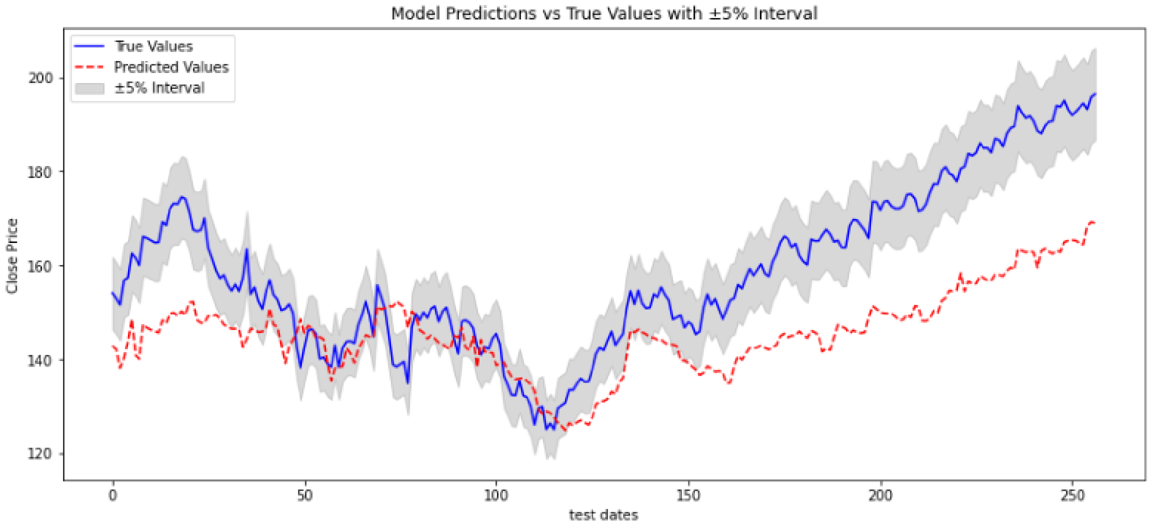
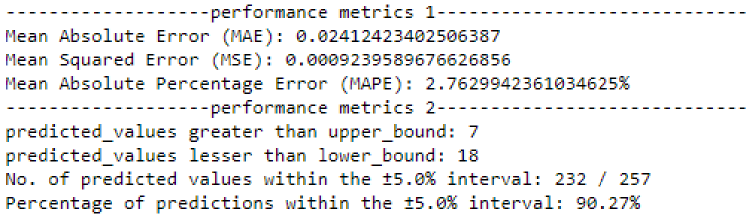


Figure 55: Experiment 12 – Model Prediction vs True Value

In the experiment 12, the model attained a MAPE – 9.31% and 30.7% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the overall trend of prediction is good, but in the second half of the test set the prediction line is diverging.



```
-------------------performance metrics 1----------------------------
Mean Absolute Error (MAE): 0.02412423402506387
Mean Squared Error (MSE): 0.0009239589676626856
Mean Absolute Percentage Error (MAPE): 2.7629942361034625%
-------------------performance metrics 2----------------------------
predicted_values greater than upper_bound: 7
predicted_values lesser than lower_bound: 18
No. of predicted values within the ±5.0% interval: 232 / 257
Percentage of predictions within the ±5.0% interval: 90.27%
```

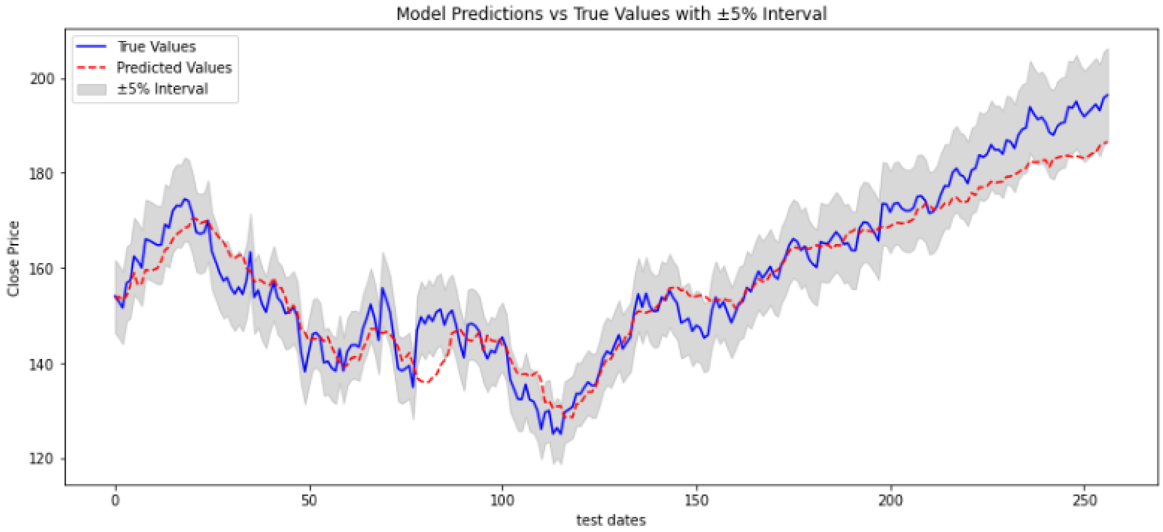


Figure 56: Experiment 13 – Model Prediction vs True Value

In the experiment 13, the best model of hyper parameter optimization using Dataset_D3 has given a best result of MAPE – 2.7% and 90% of its predictions fall within the confidence interval. Additionally, from a visual standpoint, the overall trend of prediction is good, at the last quarter of the test set the prediction line is diverging but within the 5% interval of the true value

4.3 Consolidated Results

In this section let us consolidate all the above results for a better view,

The below table consolidates all the performance metrics captured in the experiments and highlighted MAPE and Percentage of prediction within interval using gradient data bars,

Model's Performance on the test split (last 10%) of different datasets										
Sl. No.	Experiment Name	Experiment Description	Performance Metrics 1			Performance Metrics 2				
			MAE	MSE	MAPE	Predicted > Upper Bound	Predicted < Lower Bound	Predictions Within Interval	Total Predictions	% Prediction Within Interval
1	Experiment_1	Dataset_A with RF regressor Model	0.030	0.002	13.03%	19	45	193	257	75.1%
2	Experiment_2	Dataset_A with LSTM Model	0.084	0.009	6.89%	0	169	88	257	34.2%
3	Experiment_3	Dataset_A with hp opti LSTM Model	0.045	0.003	5.36%	97	0	160	257	62.3%
4	Experiment_4	Dataset_B with LSTM Model	0.291	0.104	24.78%	0	244	4	248	1.6%
5	Experiment_5	Dataset_B with hp opti LSTM Model	0.096	0.012	11.41%	109	77	62	248	25.0%
6	Experiment_6	Dataset_C with LSTM Model	0.413	0.188	35.82%	0	248	0	248	0.0%
7	Experiment_7	Dataset_C with hp opti LSTM Model	0.035	0.002	3.99%	18	34	196	248	79.0%
8	Experiment_8	Dataset_D1 with LSTM Model	0.086	0.011	6.87%	19	125	113	257	44.0%
9	Experiment_9	Dataset_D1 with hp opti LSTM Model	0.031	0.001	3.44%	1	49	207	257	80.5%
10	Experiment_10	Dataset_D2 with LSTM Model	0.114	0.018	9.19%	7	177	73	257	28.4%
11	Experiment_11	Dataset_D2 with hp opti LSTM Model	0.033	0.002	3.98%	59	0	198	257	77.0%
12	Experiment_12	Dataset_D3 with LSTM Model	0.116	0.019	9.32%	7	171	79	257	30.7%
13	Experiment_13	Dataset_D3 with hp opti LSTM Model	0.024	0.001	2.76%	7	18	232	257	90.3%

Table 2: Stock Price Prediction - consolidated results

The below table consolidates the results of classification report for NLP models on financial_phrasebank dataset,

Classification Report for NLP models on financial_phrase bank dataset						
Model No.	Model	class	Precision / NPV	recall / Sensitivity	f1-score	accuracy
1	Bi-LSTM word2vec	0	0.78	0.71	0.75	88.9%
		1	0.92	0.97	0.94	
		2	0.85	0.78	0.81	
2	FinBERT	0	0.89	0.93	0.91	91.6%
		1	0.91	0.99	0.95	
		2	0.96	0.74	0.84	
3	ChatGPT3.5	0	0.85	0.98	0.91	92.2%
		1	0.96	0.94	0.95	
		2	0.87	0.85	0.86	

Table 3: Classification Report for NLP models on financial_phrase bank dataset

5. Discussion

5.1 Evaluation of Objectives

In this section all the five research question were answered one by one.

RQ1: Justification and Comparative Analysis of Model Selection: Why are the LSTM model chosen over a traditional model like the RF regressor for stock price prediction?

When I started this project with a literature survey, I understood that LSTM is doing better in stock price prediction using machine learning method. Before choosing LSTM for my parametric study, I wanted to compare the performance of LSTM with some other tradition machine learning models say RF Regressor.

In order to answer this question, we need to compare the results of Experiment 1 and 3, that is hyperparamter optimized best RF Regressor model and hyper parameter optimized best LSTM on the same Dataset_A (Historical Data + Technical Indicators)

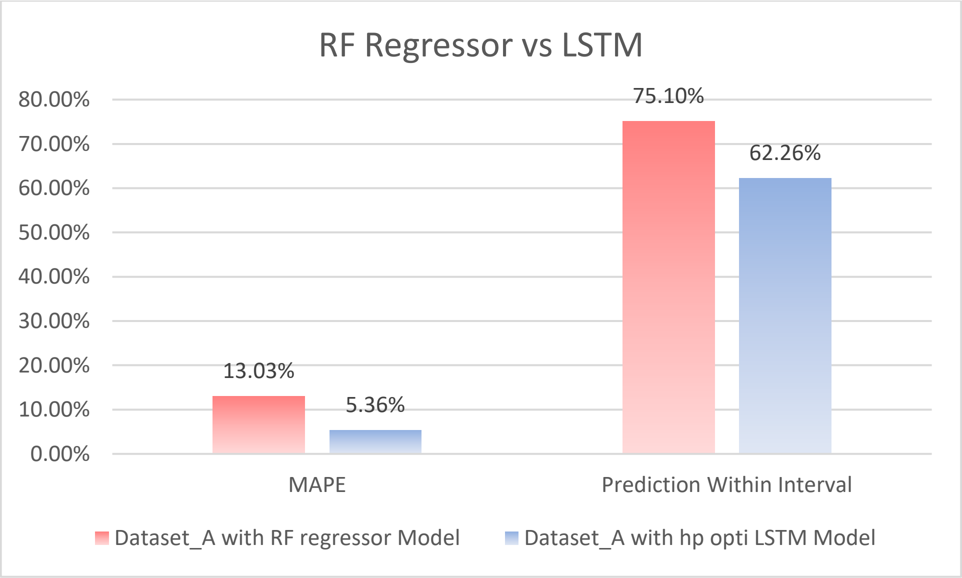


Figure 57: Results of RF Regressor vs LSTM on Dataset_A

While the LSTM model showcased a lower MAPE compared to the RF regressor, the latter outperformed LSTM in achieving a higher percentage of predictions within the specified confidence interval. Drawing upon Performance Metrics 1 and 2, it is not straightforward to declare a definitive superior model. However, a visual analysis, as seen in Figures 33 and 37, unveils a notable divergence: the RF regressor struggles to capture the trend during the test set's concluding 50 days, likely contributing to its higher MAPE, while LSTM, inherently structured to navigate through sequential data patterns, aptly traces the trends from the test dataset’s commencement to conclusion, albeit with a tendency to overestimate. It's crucial to note that applying LSTM to stock price prediction, even with ten years of historical data (over 2500 instances), may not leverage its full potential due to a somewhat limited data scope. This data limitation inadvertently benefits the RF regressor, which excels in aggregating decision tree outputs. Hence, although neither model markedly outperforms the other, LSTM seems to offer a degree of reliability in projecting future unknown data, with the RF regressor presenting certain challenges.

RQ2: Impact of Feature Diversification on Predictive Accuracy: How does the combination of features (technical indicators, financial indicators/ratios, and sentiment from financial news) influence the accuracy and reliability of stock price predictions across different datasets?

The initial hypothesis put forward that a sequential enrichment of indicators - from Technical Indicators (TI) to TI + Financial Indicators (FI) to a comprehensive blend of TI, FI, and News sentiments, would proportionally enhance the predictive capabilities of the model. Let us delve into how the experimental outcomes support this hypothesis.

To navigate through this, we compare the results of experiments 3, 5, 7, and 9, which respectively utilize hyperparameter-optimized LSTM models across Dataset_A (TI), Dataset_B (TI + FI), Dataset_C (TI + FI + News), and Dataset_D1 (TI + News). The comparison of results unfolds as follows:

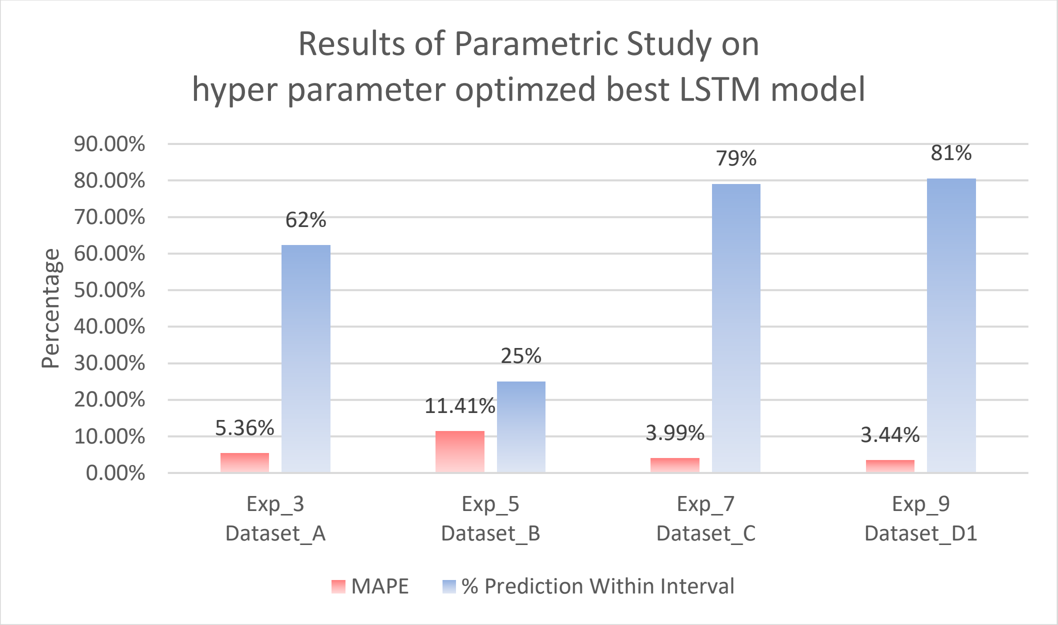


Figure 58: Results of Parametric Study

Dataset_A (Exp_3): A predictive aptitude is noticeable with TIs alone, securing 62% of predictions within the confidence interval and registering a 5.36% Mean Absolute Percentage Error (MAPE). This underscores the chosen TIs as potent predictive features.

Dataset_B (Exp_5): Introducing FIs to TIs, unexpectedly, dented the performance and plummeting to 25% of predictions within the confidence interval and elevating MAPE to 11.41%. The incorporation of FIs seemingly paralysing the model's performance, potentially attributed to the restrained data, featuring merely a single instance per annum.

Dataset_C (Exp_7): Augmenting the feature set with news sentiments propels a significant recovery in model performance, soaring to capture 79% of predictions within the confidence interval, complemented by a tapered MAPE of 3.99%. This not only rectifies the performance dip witnessed with FIs but remarkably eclipses the TI-only model (Exp_3), spotlighting the pivotal influence of news sentiments on predictive precision and reliability.

Dataset_D1 (Exp_9): Stripping FIs from the previous combination, the model, now powered by TI and news sentiments, achieves a spectacular 81% of predictions within the confidence interval and a minimal 3.44% MAPE. This highlights that excluding FIs not only nullifies their previously noted adverse impact but also slightly optimizes the model’s performance relative to all prior configurations.

Among the various feature combinations explored, the combination of TI and News sentiments decisively outshines all others, clinching the foremost position with a laudable 81% of predictions within the confidence interval and a mere 3.44% MAPE. Is this the best? No, we tried out 3 sophisticated NLP models on arriving the sentiments of the news, which will be discussed in the RQ4.

RQ5: Optimization and Model Robustness: How does hyperparameter optimization impact the performance of LSTM models across various experiments with diversified datasets, and does the optimized model consistently outperform its non-optimized counterpart across all datasets?

Analyzing the model performance data from experiments 2 to 13 sheds light on how hyperparameter optimization influences LSTM model results across various datasets, precisely A, B, C, D1, D2, and D3.

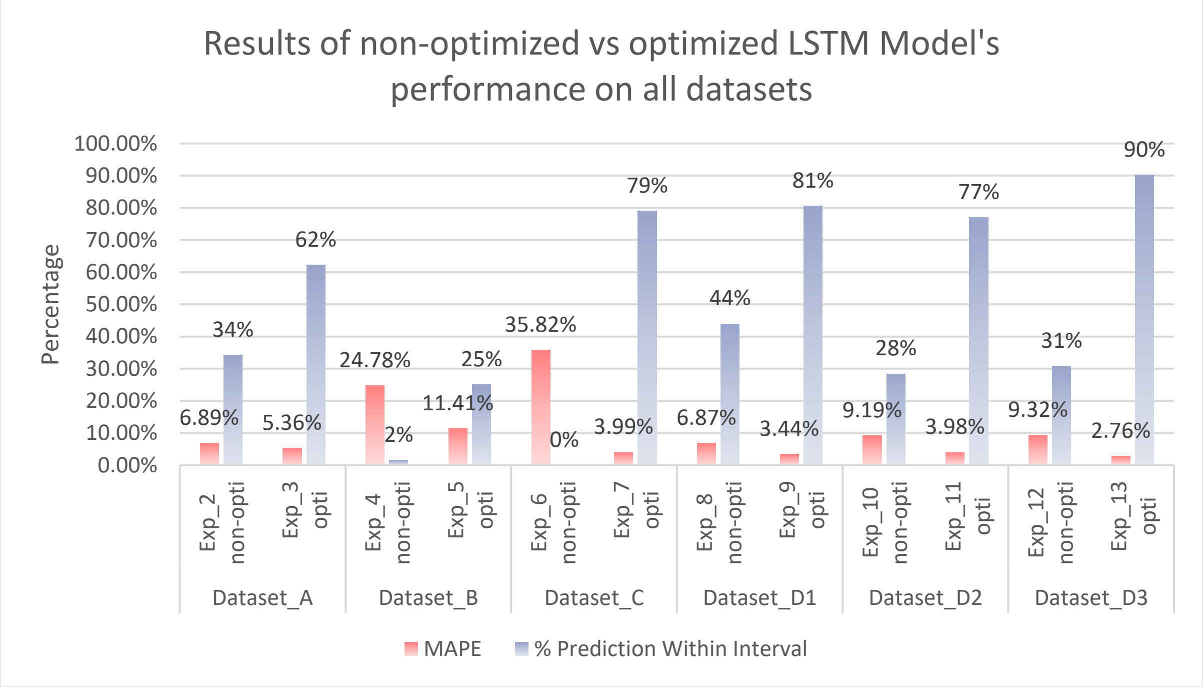


Figure 59: Results of non-optimized vs optimized LST models

Comparative Analysis: Optimized vs Non-Optimized LSTM Models

Dataset_A (Exp_2 vs Exp_3): Optimization improves performance significantly: percentage of predictions within the interval jumps from 34% to 62%, while MAPE reduces from 6.89% to 5.36%.

Dataset_B (Exp_4 vs Exp_5): Here too, optimization enhances the performance, increasing the predictions within the interval from a mere 2% to 25% and reducing the MAPE from 24.78% to 11.41%.

Dataset_C (Exp_6 vs Exp_7): A similar pattern is seen. Optimizing drastically boosts predictions within the interval from 0% to 79% and slices the MAPE from 35.82% to a minimal 3.99%.

Dataset_D1 (Exp_8 vs Exp_9): The trend of improvement continues predictions within the interval rise from 44% to a substantial 81% post-optimization, and MAPE decreases from 6.87% to 3.44%.

Dataset_D2 (Exp_10 vs Exp_11): Again, optimization provides clear benefits, increasing the predictions within the interval from 28% to 77% and reducing the MAPE from 9.19% to 3.98%.

Dataset_D3 (Exp_12 vs Exp_13): The performance improvement is compelling here too, with predictions within the interval soaring from 31% to 90% and MAPE trimming down from 9.32% to 2.76% after optimization.

Performance Enhancement: Across all datasets, hyperparameter optimization invariably elevates the model's predictive capability, evidenced by an upward shift in predictions within the confidence interval and a concurrent reduction in MAPE.

Consistency in Optimization Impact: Strikingly, the optimized model consistently outperforms its non-optimized counterpart, substantiating the robustness and efficacy of the optimization process across diverse datasets.

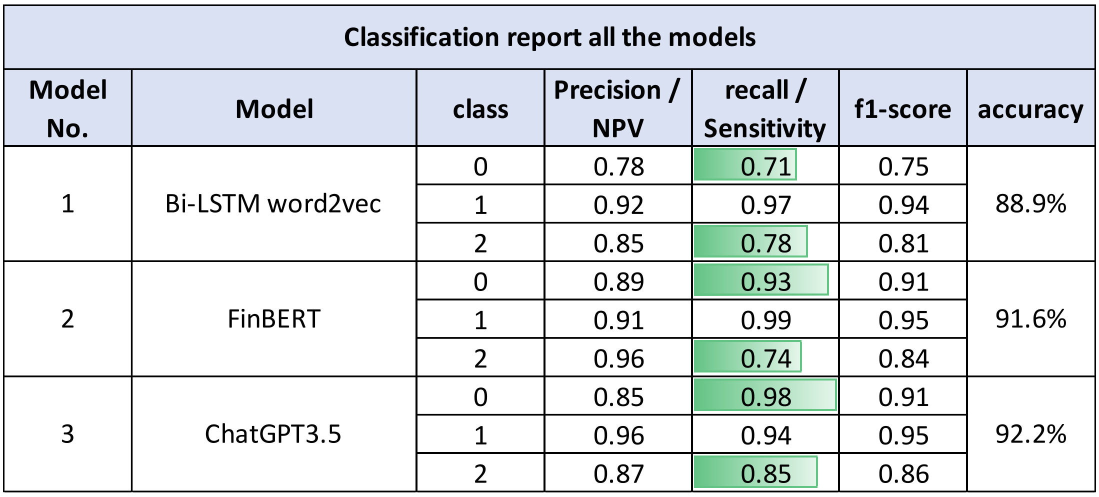
Model Robustness: The disparity in performance metrics between non-optimized and optimized models is substantial, indicating that the robustness of the LSTM models is notably amplified post-optimization.

Through a thorough analysis of experiments across various datasets, it is transparent that hyperparameter optimization substantially and consistently enhances the predictive accuracy and robustness of LSTM models. Consequently, an optimized model not only boosts the reliability and credibility of predictions but also showcases versatility and consistency in outperforming non-optimized versions across a plethora of datasets with diversified features.

RQ3: Comparative Model Efficacy in Sentiment Analysis: How do different NLP models (Bi-LSTM with word2vec, FinBERT, and LLM) compare in terms of accuracy and effectiveness when applied to sentiment analysis of financial news, and how does their performance translate when applied to unlabelled New York Times dataset?

Best model for sentiment analysis on financial_phrasebank dataset.

This could be a straightforward comparison of accuracies of all three models, where Bi-LSTM word2vec has got 88.9%, FinBERT 91.6% and ChatGPT3.5 also 92.2%. From this it could be difficult to say that ChatGPT3.5 has outperformed others, whereas other 2 models are also very close to the first one. Other than accuracy there is another reason why ChatGPT is better than the other models is the way it handled the “False Negative” particularly in the deciding class of negative and positive classes. That is, Recall Values / sensitivity = True Positive / (True positives + False Negatives) which indicates that sensitivity of the model in predicting the wrong classes. Please find the recall values of all three models consolidated here under, wherein I have made a gradient fill only on the recall values of class 0 and 1.



Classification report all the models						
Model No.	Model	class	Precision / NPV	recall / Sensitivity	f1-score	accuracy
1	Bi-LSTM word2vec	0	0.78	0.71	0.75	88.9%
		1	0.92	0.97	0.94	
		2	0.85	0.78	0.81	
2	FinBERT	0	0.89	0.93	0.91	91.6%
		1	0.91	0.99	0.95	
		2	0.96	0.74	0.84	
3	ChatGPT3.5	0	0.85	0.98	0.91	92.2%
		1	0.96	0.94	0.95	
		2	0.87	0.85	0.86	

Table 4: Consolidated Classification Report for Sentiment Analysis

From the above table it is clear that ChatGPT has a better recall value for both the deciding classes that is negative and positive as 0.98 and 0.85 respectively, whereas the second best was FinBERT with 0.93 and 0.74 (which is 11% lesser than the best one). So, now we can say ChatGPT has performed better than the other two models.

how does their performance translate when applied to unlabelled New York Times dataset?

Since this dataset doesn't have labels to validate and compare, we have tried a visual validation as discussed in the section 4.1.4 which is not conclusive, so we tried an indirect method of validation by pulling the sentiments individually to the LSTM stock price prediction (i.e. Dataset D1, D2 and D3) and comparing the result of price prediction to have the best one, which will be seen in the RQ4.

With the above discussion on the first phase of the question, the best NLP model on financial_phrasebank data, we have concluded that ChatGPT is doing better, but we cannot have that in our mind when it comes to the NYTimes dataset which might be general news statements corpus, on which word2vec could have done better. This also a reason for the RQ4.

RQ4: Influence of Sentiment Analysis on Predictive Modeling: To what extent does incorporating sentiment analysis of financial news, obtained through different NLP models, enhance the performance of LSTM models in predicting stock prices?

In order to answer this question, we compare the results of experiments 9, 11, and 13 which utilize hyperparameter-optimized LSTM models across Dataset_D1 (TI + News-word2vec), Dataset_D2 (TI + News-FinBERT), Dataset_D3 (TI + News-ChatGPT3.5). The comparison of results unfolds as follows:

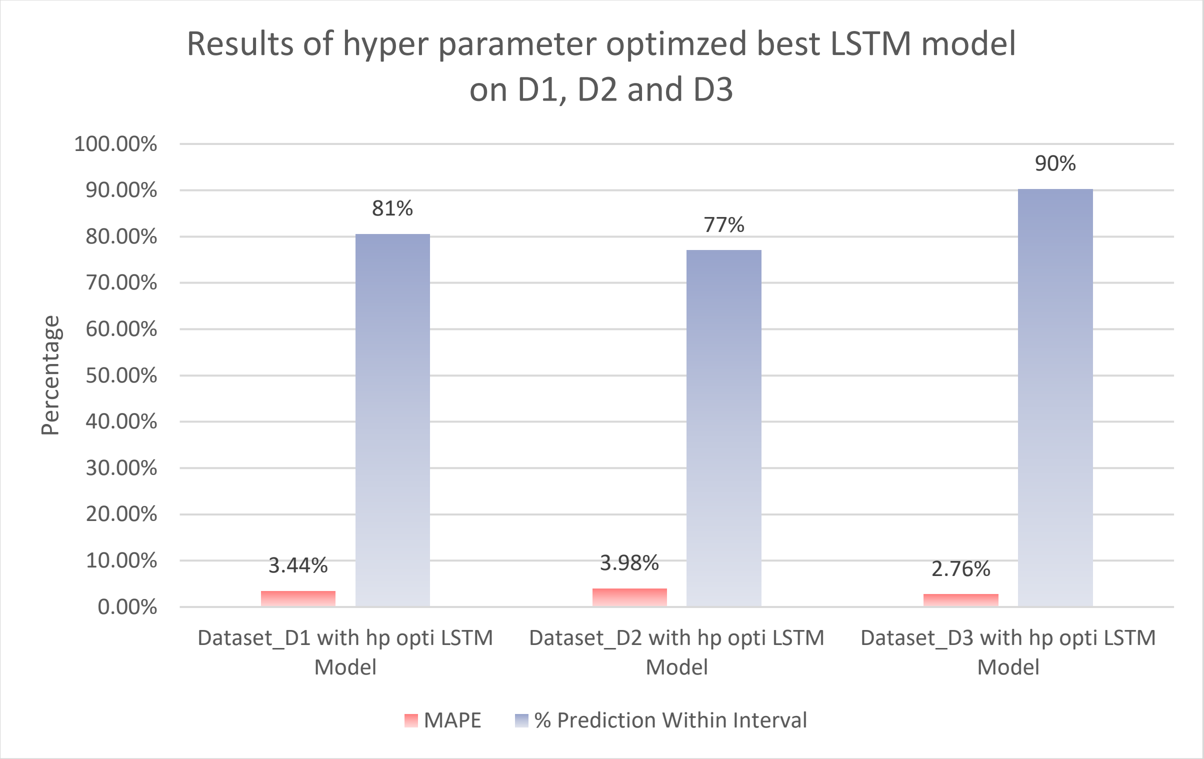


Figure 60: Examination of Influence of Sentiment Analysis on Stock Price Prediction

Dataset_D1 (Exp_9 | TI + News-word2vec): In this model, 81% of the predictions are within the confidence interval where MAPE stands at 3.44%.

Dataset_D2 (Exp_11 | TI + News-FinBERT): In this model, the percentage of predictions within confidence interval has fallen to 77% with a raise in MAPE to 3.98%.

Dataset_D3 (Exp_13 | TI + News-ChatGPT3.5): In this model, we got the highest predictions within the confidence interval of 90% with MAPE of 2.76%.

From the above results it is clear that the performance of the sentiments from ChatGPT3.5 has outperformed with 90% of prediction within confidence interval and 2.76% of MAPE. This is contrasted by word2vec, which stands second at 81% with a similar MAPE of 3.44%, and FinBERT, trailing at 77% and 3.98% respectively.

A visual inspection of the prediction line against the true values (Figures 49, 53, and 57) reveals that the Exp_11 model utilizing FinBERT for sentiment analysis demonstrates superior capability in capturing stock price highs and lows compared to the other two models. Notably, in the final 50 days of the test set, the Exp_9 and Exp_13 models' predictions diverge from the true values, while the Exp_11 model does not show such divergence. This suggests the Exp_11 model better retains predictive accuracy towards the end of the timeframe examined. Additional analysis and experimentation may be reasonable to further investigate these dynamics and potentially identify other influential factors.

Conclusively, the combination of Technical Indicators supplemented with News sentiments processed by ChatGPT 3.5 towered over other model-dataset combinations explored in this thesis, with a commanding 90% of predictions nested within the confidence interval and an impressive MAPE of 2.76%.

5.2 Comparing with the Literature Review:

In the paper “Stock Price Prediction Using News Sentiment Analysis”, Mohan, Mullapudi, Sammeta, Vijayvergia, & Anastasiu (2019) has achieved **MAPE of 2.03** in their LSTM model with stock prices and textual polarity data, and in another paper “Stock Prediction by Integrating Sentiment Scores of Financial News and MLP-Regressor: A Machine Learning Approach” by Maqbool, Aggarwal, Kaur, Mittal, & Ganaie (2023), they have used MLP-regressor with FLAIR sentiment score and got a best **MAPE of 1.48**, but they have used different company, different news data and different architecture. whereas we got our best **MAPE as 2.76%** in the hyperparameter optimised best LSTM model with Technical Indicators and sentiments from NYTimes dataset derived by ChatGPT3.5. In order to improve we may need to improve the hyperparameters or we may need to run more trails to try some more combinations in the LSTM model or we may need to fine tune the prompt of the ChatGPT3.5 for even better results.

6. Evaluation, Reflections, and Conclusions

6.1 Evaluation of the Study

6.1 Evaluation of the Study

This study presented an in-depth examination of stock price prediction for Apple Inc. using Long Short-Term Memory (LSTM) networks enhanced with varied combinations of predictive features. The core focus involved investigating the incremental integration of technical indicators, financial ratios, and sentiment analysis of financial news to determine their impact on forecasting accuracy. And, the research analyzed which of the three NLP techniques – Bi-LSTM with word2vec, FinBERT and Large Language Models (LLM) like ChatGPT3.5 are performing better in the sentiment analysis of financial news for stock price prediction.

The key objectives were achieved through systematic experiments analyzing multiple datasets. The research revealed that augmenting LSTM models with news sentiment analysis significantly improves predictive performance. In particular, sentiment derived from LLM like ChatGPT3.5, when combined with technical indicators, yielded the best results.

A major contribution is the novel incorporation of a decade's worth of company-specific news data from the New York Times for sentiment analysis. The use of latest technology like LLM-ChatGPT3.5 on 10 years of company-specific news data is also novel. Such domain-targeted sentiment analysis with sizable news data has not been commonly pursued in prior academic literature.

Hyperparameter optimization was consistently shown to enhance model robustness across diverse datasets. The study also provided comparative analysis of different NLP techniques for financial sentiment analysis on labeled and unlabeled data.

However, the scope was restricted to only one company's data. More extensive experiments across stocks from varied industries could better validate the versality of the proposed approach. Overall, within the defined scope, the study successfully achieved its objectives and presented insightful techniques for integrating predictive signals into stock forecasting models.

6.2 Reflections on the Research Process

Undertaking this dissertation research project has been a profoundly enriching learning experience on both academic and personal fronts. The opportunity to pursue a self-driven investigative study while applying classroom concepts has served as an invaluable bridge to real-world research.

One of the most meaningful lessons has been the importance of thoroughly reviewing academic literature to identify gaps and chart a unique research path. The extensive literature survey enabled me to strategically position my work to address limitations around comparative feature analysis and financial news sentiment incorporation. However, gathering, reading, comprehending and synthesizing vast amounts of literature was undoubtedly challenging.

The research process underscored the criticality of meticulous planning and organization. Maintaining a methodical workflow while toggling between coding, writing, and experimentation was crucial yet demanding. Debugging errors in code or results required tenacity. Managing timelines to coordinate intermediate deliverables with a long-term vision demanded diligence.

I had limited financial and stock trading domain knowledge before starting this project, which posed an initial challenge. With the knowledge I have gained through this research, some aspects I could modify are, I would annotate a considerable sample of the NYTimes dataset (around 25% or 500 sentences) with the help of financial analysts or domain experts. This could enrich the sentiment analysis on this unlabeled data.

Analyzing results to derive cogent, data-driven insights stretched my analytical thinking capabilities. Objectively interpreting the outcomes to engage in evidence-based discussion without confirmation bias or overextrapolation was intellectually testing. Communicating the technical intricacies of the models and methods comprehensibly yet precisely presented a steep learning curve. However, observing how the systematic experiments illuminated the research questions was incredibly rewarding.

On the whole, undertaking this dissertation has expanded my skills and knowledge across the research lifecycle - from ideation, planning, and organization to literature analysis, coding, experimentation, results interpretation and communication. The self-discipline, perseverance, analytical thinking and troubleshooting required have contributed immensely to my personal and intellectual growth. The culmination of this project provides great satisfaction along with a foundation to build upon through future research endeavors.

Let me know if you would like me to modify or expand on any part of this reflection. I can provide more details if needed. Please feel free to suggest any improvements.

6.3 Conclusions

This dissertation presented a comprehensive study focused on integrating machine learning and natural language processing techniques for stock price forecasting. The research systematically analyzed the effects of incorporating technical indicators, financial ratios, and news sentiment analysis on the predictive accuracy of Random Forest and LSTM models for Apple stock prices.

The key conclusions derived from the extensive comparative experiments are as follows:

- LSTM networks show promising capabilities in modeling the temporal relationships in stock price data for prediction, outperforming RF in capturing trends despite no model distinctly surpassing the other.
- Enriching LSTM models with news sentiment analysis significantly enhances predictive performance compared to using just technical and financial indicators. Adding financial indicators was a dip in the model's performance.
- Sentiment analysis using Large Language Models like ChatGPT3.5, when combined with technical indicators, delivers the highest predictive accuracy.
- In the sub-task of determining the best NLP model for sentiment analysis of financial news from the financial_phrasebank dataset, ChatGPT3.5 emerged superior with 92.2% accuracy.
- For the main task of stock price prediction, the best performance came from the combination of technical indicators and news sentiments derived by ChatGPT3.5, yielding 2.76% MAPE and 90% predictions within +/-5% confidence band.
- Hyperparameter optimization consistently improves the robustness of LSTM models across diverse datasets, underscoring its importance.

The study makes notable contributions by providing unique insights into blending predictive signals and presenting an extensive methodology for integrating news sentiment analysis into stock price forecasting using state-of-the-art NLP models.

While promising results have been achieved within a defined scope, further research across more stocks and industry sectors is imperative to validate the versatility of the techniques explored in this dissertation. Overall, this research contributes significantly to the literature and offers a springboard for future work in this intricate domain of stock market prediction.

6.4 Recommendations for Future Work

While this dissertation makes notable contributions, there remain several promising avenues for extending this research:

- To further enhance the accuracy of stock price prediction, additional predictive features like macroeconomic factors could be incorporated. The literature review indicated variables including interest rates, unemployment rates, and market volatility indices can potentially improve forecasts.
- In this work, Large Language Models (LLM) with few-shot prompt fine-tuning delivered strong results for sentiment analysis. However, optimizing the prompt engineering process merits deeper exploration, as excessive prompt complexity can sometimes confuse models. Research into more streamlined prompts could be worthwhile.
- For the unlabeled New York Times dataset, manual annotation by financial domain experts on a sizable subset could enrich sentiment analysis. Distinct expert-annotated samples would provide a comparison benchmark and facilitate training domain-targeted models.
- Long Short-Term Memory (LSTM) networks were used extensively in this research. Investigating transformer architectures like Temporal Convolutional Networks (TCN) or Transformers could offer an interesting comparison and potentially improve predictive performance.

Expanding the experimental scope to incorporate more stocks across diverse industry sectors and markets would better validate the generalizability of the models and techniques explored in this dissertation.

Glossary

- TI – Technical Indicators.
- FI – Financial Indicators.
- FN – Financial News
- NYT – New York Times
- LSTM – Long Short Term Memory
- RF – Random Forest
- Bi-LSTM – Bi direction LSTM

References

Araci, D. (2019). *FinBERT: Financial Sentiment Analysis with Pre-trained Language Models FinBERT: Financial Sentiment Analysis with Pre-trained Language Models*. [online] Available at: https://arxiv.org/pdf/1908.10063v1.pdf [Accessed 14 May 2023].

Bhandari, H.N., Rimal, B., Pokhrel, N.R., Rimal, R., Dahal, K.R. and Khatri, R.K.C. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications*, p.100320. doi:https://doi.org/10.1016/j.mlwa.2022.100320.

Boozer, B., Rainwater, L. and Lowe, S. (2007). financial crisis. *Journal of Finance and Accountancy*, [online] 21, pp.1–10. Available at: https://digitalcommons.jsu.edu/cgi/viewcontent.cgi?article=1142&context=fac_res [Accessed 7 Oct. 2023].

Fridson, M.S., Alvarez, F. and Netlibrary, I. (2002). *Financial statement analysis : a practitioner's guide*. New York: John Wiley & Sons.

Heo, J. and Yang, J.Y. (2016). Stock Price Prediction Based on Financial Statements Using SVM. *International Journal of Hybrid Information Technology*, 9(2), pp.57–66. doi:https://doi.org/10.14257/ijhit.2016.9.2.05.

Hota, J., Chakravarty, S., Paikaray, B. and Bhoyar, H. (2022). *Stock Market Prediction Using Machine Learning Techniques*. [online] ACI’22: Workshop on Advances in Computation Intelligence, its Concepts & Applications at ISIC 2022, May 17-19, Savannah, United States. Available at: https://ceur-ws.org/Vol-3283/Paper86.pdf.

Huang, A.H., Wang, H. and Yang, Y. (2022). FinBERT : A Large Language Model for Extracting Information from Financial Text†. *Contemporary Accounting Research*. doi:https://doi.org/10.1111/1911-3846.12832.

Malo, P., Sinha, A., Korhonen, P., Wallenius, J. and Takala, P. (2013). Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4), pp.782–796. doi:https://doi.org/10.1002/asi.23062.

Maqbool, J., Aggarwal, P., Kaur, R., Mittal, A. and Ganaie, I.A. (2023). Stock Prediction by Integrating Sentiment Scores of Financial News and MLP-Regressor: A Machine Learning Approach. *Procedia Computer Science*, 218, pp.1067–1078. doi:https://doi.org/10.1016/j.procs.2023.01.086.

Mohan, S., Mullapudi, S., Sammeta, S., Vijayvergia, P. and Anastasiu, D.C. (2019). *Stock Price Prediction Using News Sentiment Analysis*. [online] IEEE Xplore. doi:https://doi.org/10.1109/BigDataService.2019.00035.

Müller, A.C. and Guido, S. (2017). *Introduction to machine learning with Python : a guide for data scientists*. Beijing: O’reilly.

Murphy, J.J. (1999). *Technical analysis of the financial markets : a comprehensive guide to trading methods and applications*. New York: New York Institute Of Finance.

Sen, J. (2018). *Stock Price Prediction Using Machine Learning and Deep Learning Frameworks XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE Stock Price Prediction Using Machine Learning and Deep Learning Frameworks*.

Vargas, M.R., dos Anjos, C.E.M., Bichara, G.L.G. and Evsukoff, A.G. (2018). Deep Leaming for Stock Market Prediction Using Technical Indicators and Financial News Articles. *2018*

International Joint Conference on Neural Networks (IJCNN). doi:https://doi.org/10.1109/ijcnn.2018.8489208.

Vijh, M., Chandola, D., Tikkiwal, V.A. and Kumar, A. (2020). Stock Closing Price Prediction using Machine Learning Techniques. *Procedia Computer Science*, 167(167), pp.599–606. doi:https://doi.org/10.1016/j.procs.2020.03.326.

Yang, J., Wang, Y. and Li, X. (2022). Prediction of stock price direction using the LASSO-LSTM model combines technical indicators and financial sentiment analysis. *PeerJ Computer Science*, 8, p.e1148. doi:https://doi.org/10.7717/peerj-cs.1148.

Zhang, W., Deng, Y., Liu, B., Jialin Pan, S. and Bing, L. (2023). *Sentiment Analysis in the Era of Large Language Models: A Reality Check*. [online] Available at: https://browse.arxiv.org/pdf/2305.15005.pdf [Accessed 6 Oct. 2023].