Stock Price Prediction Using Machine Learning, Deep Learning, and Transformer-Based Transfer Learning

A thesis submitted in partial fulfillment of the Requirements for the degree of Master of Science in Computer Science

By

Raviteja Pothala

May 2024
The thesis of Raviteja Pothala is approved:

___________________________________  ______________________
Dr. Robert McIlhenny, Ph.D.                      Date

___________________________________  ______________________
Dr. Katya Mkrtchyan, Ph.D.                      Date

___________________________________  ______________________
Dr. Taehyung George Wang, Ph.D., Chair        Date
Acknowledgments

I would like to extend my deepest gratitude to Dr. George Wang, Ph.D., for his remarkable guidance throughout my thesis journey. His consistent support, invaluable counsel, and sharp insights have profoundly impacted the direction and success of my research, for which I am immensely thankful. I am also thankful to Dr. Robert McIlhenny, Ph.D., for his willingness to share his expertise and provide critical feedback, even amidst his other numerous responsibilities. His input has significantly enriched the caliber of my work.

Additionally, my heartfelt thanks go to Dr. Katya Mkrtchyan, Ph.D., for his thorough and constructive critiques, which have been crucial in enhancing my manuscript. His dedication to scholarly excellence and meticulousness have been key factors in my academic development.

My profound appreciation also goes to my parents, whose endless encouragement, support, and faith in my potential have been my constant motivation. Their sacrifices and love have been foundational in my educational path, and I am eternally grateful for their constant support. This accomplishment is a tribute to them, acknowledging their pivotal role in my life.
Dedication

This work is a tribute to everyone who has offered their support and encouragement throughout my journey. To my family, whose unwavering love and faith in my potential have been the bedrock of my pursuits. Your constant support and sacrifices have been the driving force behind my commitment to excel in all that I undertake. I am deeply thankful for your guidance and encouragement, which have molded me into who I am.

To my friends and colleagues, I extend my gratitude for your companionship, solidarity, and understanding during this project. Your support and shared journeys have been a source of invaluable motivation and inspiration.

This work is also in honor of my mentors and educators, whose advice, wisdom, and expertise have played a crucial role in my academic and professional development. Your guidance has inspired me to explore new horizons and aim for excellence in my work.

Finally, this project is dedicated to all those with the ambition to effect change in the world. May this endeavor stand as a symbol of the strength of persistence, passion, and commitment in surmounting obstacles and achieving one’s goals.
Table of Contents

Copyright Page.................................................................ii
Signature Page .................................................................. iii
Acknowledgments.............................................................. iv
Dedication ..................................................................... v
List of Figures .................................................................. ix
List of Abbreviations ........................................................ xi
Abstract ......................................................................... xii
Chapter 1 - Introduction .................................................. 1
  1.1 Background and Context ............................................ 1
  1.2 Problem Statement and Research Gap ........................ 1
  1.3 Research Aims and Objectives .................................... 2
  1.4 Research Questions .................................................. 3
  1.5 Significance and Contributions of the Study .................. 5
  1.6 Scope and Delimitations ............................................. 5
    1.6.1 Scope ................................................................ 6
    1.6.2 Delimitations .................................................... 6
  1.7 Overview of Methodological Approaches ....................... 7
  1.8 Outline of the Proposed Framework ............................ 8
Chapter 2 - Literature Survey ............................................. 10
  2.1 Overview of Stock Price Prediction Challenges ............... 10
  2.2 Review of Traditional and Modern Financial Models ........ 11
  2.3 Role of Machine Learning in Financial Markets ............. 11
  2.4 Deep Learning Techniques for Time-Series Analysis ........ 12
  2.5 Transfer Learning in Financial Forecasting .................... 13
  2.6 Comparative Analysis of Predictive Models in Finance ...... 14
  2.7 Integrating Multiple Data Sources for Enhanced Predictions 15
  2.8 Advances in Algorithmic Trading Strategies .................. 17
  2.9 Evaluation Metrics for Financial Models ........................ 18
  2.10 Transformer Models in Financial Forecasting ................. 19
Chapter 3 - Methodology ................................................... 20
  3.1 Research Design and Strategy ..................................... 20
  3.2 Data Collection and Pre-Processing .............................. 20
3.3 Model Selection and Development ................................................................. 21
3.4 Implementation of Transfer Learning Techniques ........................................... 22
3.5 Validation and Performance Evaluation Methods ............................................ 22
3.6 Block Diagram .................................................................................................. 23
3.7 Limitations and Assumptions .......................................................................... 24

Chapter 4 - System Design and Implementation .................................................... 26
4.1 Data Features and Selection Criteria ............................................................... 26
4.2 Data Preprocessing and Normalization ............................................................. 26
4.3 Application of Machine Learning ..................................................................... 27
  4.3.1 Model Training and Parameter Tuning ....................................................... 28
  4.3.2 Model Integration and Ensemble Techniques ............................................... 28
4.4 Deployment of Deep Learning Models ............................................................... 29
  4.4.1 RNN and LSTM Application ..................................................................... 29
  4.4.2 Application of BI-LSTM and GRU Models ............................................... 30
4.5 Transfer Learning Model Application ............................................................... 31
4.6 Performance Metrics and Model Comparison ................................................... 31

Chapter 5 - Implementation .................................................................................... 33
5.1 Data Set Description ......................................................................................... 33
5.2 Information about the Data .............................................................................. 33
5.3 Feature Description .......................................................................................... 34
5.4 Inspection of Missing Values ............................................................................ 35
5.5 Correlation Matrix ........................................................................................... 36
5.7 Closing Stock Price for Top-5 Stocks over 5 Years .......................................... 39
5.7 Seasonality Decomposition .............................................................................. 41
  5.7.1 Seasonality Decomposition for FB ............................................................. 41
  5.7.2 Partial and Auto Correlation for AAPL ....................................................... 43
  5.7.3 Partial and Auto Correlation for FB ............................................................. 45
5.8 Comparative Analysis of 7 Tech Stocks ............................................................ 47
5.9 Overview of Stocks Over 5 Years ...................................................................... 49
5.10 Daily Return of Top 5 Stock ............................................................................ 50
5.11 Comparative Analysis of Stock Prices .............................................................. 52
5.12 Algorithm Implementation .............................................................................. 52
  5.12.1 ARIMA Model .......................................................................................... 53
List of Figures

Figure 1 Overview of The Methodologies [2] .................................................................................. 7
Figure 2 Stages of Implementation [4] ......................................................................................... 16
Figure 3 Proposed Block Diagram [14] ......................................................................................... 23
Figure 4 Records of the Dataset ................................................................................................. 34
Figure 5 Missing Values Inspection .............................................................................................. 35
Figure 6 Correlation Matrix .......................................................................................................... 36
Figure 7 Top 5 Stock Tickers by Average Trade Volume .............................................................. 38
Figure 8 Closing Stock Price for Top-5 Stocks over 5 Years ......................................................... 40
Figure 9 Seasonality Decomposition for FB ............................................................................... 42
Figure 10 Auto Correlation Plot for AAPL .................................................................................. 44
Figure 11 Partial Auto Correlation of AAPL ............................................................................... 44
Figure 12 Auto Correlation for FB ............................................................................................... 45
Figure 13 Partial Auto Correlation for FB ................................................................................... 46
Figure 14 Comparative Analysis of 7 Tech Stocks ....................................................................... 47
Figure 15 Overview of Stocks over 5 Years .................................................................................. 49
Figure 16 Overview of Daily Return of Stock ............................................................................. 50
Figure 17 Comparative Analysis of Top Tech Stock Price ............................................................ 52
Figure 18 Model Architecture for ARIMA ................................................................................... 54
Figure 19 SARIMAX Model Architecture .................................................................................... 57
Figure 20 Linear Regression Model .............................................................................................. 60
Figure 21 Decision Tree Regressor Model Architecture .............................................................. 62
Figure 22 Basic Architecture of RNN [18] ................................................................................... 64
Figure 23 Model Summary of RNN Architecture ......................................................................... 65
Figure 24 RNN Prediction Plot ..................................................................................................... 67
Figure 25 Simple Architecture of Simple LSTM [7] .................................................................... 68
Figure 26 Simple Architecture of Simple LSTM ......................................................................... 69
Figure 27 Simple LSTM Architecture ......................................................................................... 71
Figure 28 Bi-LSTM Architecture [27] ......................................................................................... 72
Figure 29 Bi-LSTM Architecture ................................................................................................. 73
Figure 30 Basic Architecture of LSTM & GRU [1] ..................................................................... 75
Figure 31 GRU Model Summary ................................................................................................. 76
Figure 32 Summary of Hybrid Algorithm ...............................................................78
Figure 33 Parameters of Transformer Model.......................................................80
Figure 34 Forecasted Plot for the Transformer Model ...........................................81
Figure 35 Summary of GPT-2 Model.....................................................................82
Figure 36 Forecasted Plot for GPT-2 Model ...........................................................83
Figure 37 Summary of TST Model.........................................................................85
Figure 38 Forecasted Plot for TST Model ...............................................................86
Figure 39 Hybrid Model Summary with Transformer Interpretation....................87
Figure 40 Forecasted Plot for Hybrid Model with Transformer.............................88
Figure 41 Performance Evaluation for Apple Stock after Transformer Interpretation....89
Figure 42 Forecasted Plot for Apple Stock after Transformer Interpretation.............90
Figure 43 Performance Evaluation Parameter for Amazon Stock after Transformer Interpretation........................................................................................................90
Figure 44 Forecasted Plot for The Amazon Stock after Transformer Layer Interpretation ....91
Figure 45 Models Comparison..............................................................................93
List of Abbreviations

ARIMA - Auto Regressive Integrated Moving Average
Bi-LSTM - Bi-directional Long Short-Term Memory
GPT - Generative Pre-trained Transformer
GRU - Gated Recurrent Units
IT - Information Technology
LSTM - Long Short-Term Memory
MAE - Mean Absolute Error
ML - Machine Learning
MSE - Mean Squared Error
RMSE - Root Mean Squared Error
RNN - Recurrent Neural Networks
SARIMA - Seasonal Auto Regressive Integrated Moving Average
TST - Time Series Transformer
Abstract

Stock Price Prediction Using Machine Learning, Deep Learning, and Transformer-Based Transfer Learning

By

Raviteja Pothala
Master of Science in Computer Science

In the field of stock market price prediction modelling conventional and existing methodologies are often inadequate, highlighting pressing for the need for innovative approaches for enhancing reliability and accuracy. In this research, the forefront of predicting stock market prices by integrating the strengths of statistical analysis, machine learning models, deep learning neural networks, transformers with transfer learning techniques are explored. The main objective is to explore the models as foundation for transfer learning in stock market forecasts by employing a comprehensive array of techniques. The study starts with an examination of statistical approaches of ARIMA and SARIMA, in time series analysis, providing insights for identifying patterns and seasonality behaviors in stock prices. Further, we explore machine learning algorithms such as linear regression, and parameter tuned decision trees where each model offers a distinct perspective on market dynamics: linear regression is valued for its simplicity and transparency, decision trees for their explicit classification and regression tasks using information gain.

The main share of establishing a foundation model comes from deep learning neural networks includes Recurrent Neural Networks (RNNs), Bidirectional LSTM (BI-LSTM), Gated Recurrent Units (GRUs), Long Short-Term Memory (LSTM) networks which are well to known to capture long range dependencies and intricacies in the stock data, and the impact of
Transformer model, GPT-2, and Time Series Transformers (TST). Furthermore, Transformer models and GPT-2 introduce a new direction in handling sequential data given the feature of multi-head attention to capture long-range dependencies with remarkable efficiency. TST, a variant tailored for time series data, further enhances our analytical toolkit by providing a mechanism specifically designed for capturing temporal patterns. An approach utilizing Transfer Learning technique is also introduced, with hybrid models developed to capture the intricacies in the stock market data.
Chapter 1 - Introduction

1.1 Background and Context

Financial stock market is complex and subject to volatility influenced by lot of factors ranging from geopolitical events, investor, retail sentiments and other factors. In this environment, accurate prediction of stock prices remains a pivotal challenge for investors, traders, and financial analysts. Traditional financial models, while useful, often fall short in capturing the multifaceted nature of market movements. The integration of machine learning, neural networks used for deep learning, and transfer learning techniques, has opened new opportunity for bolstering the predictive accuracy of stock prices. These techniques leverage their architectures capturing long range dependencies identifying complex patterns within seasonal or cyclic time series data. By integrating these innovative approaches, the goal is to develop a more robust predictive framework that can adapt to the rapid changes in the market and provide more reliable forecasts. This integration represents a significant shift from conventional methods, promising to revolutionize the way financial markets are analyzed and how investment decisions are made. The exploration of these advanced techniques in stock price prediction not only holds the potential to improve investment strategies but also contributes to the broader field of financial analysis and forecasting.

1.2 Problem Statement and Research Gap

Despite the significant advancements in financial modeling, the accurate prediction of stock prices remains a daunting challenge due to the inherent unpredictability and complexity of financial markets. Traditional financial models like CAPM assumes stock prices possess full information and perform by using basic economic indicators, failing to capture the intricacies in the time series data of stock and cope with the volatile nature of the stock market. Additionally, these models sometimes fail to account for sudden market shifts caused by unforeseen events like geopolitical changes resulting in shift of investor and retailer sentiment, leading to inaccurate predictions and potential financial losses when employed for trading.

The integration of machine learning algorithms and neural networks used for deep learning has shown promise in addressing some of these limitations by analyzing vast datasets and identifying non-linear patterns within the market. However, there exists a research gap in
effectively combining these techniques leveraging transfer learning to further boost the adaptability and accuracy of predictive models. Transformers based models being the recent proliferation are tried for time series analysis to an extent and Transfer learning, in particular, can leverage pre trained models from one domain to improve predictions by fine tuning in target domain, yet its application in financial forecasting is relatively unexplored and underutilized.

Another aspect of the research gap is the need for a comprehensive framework that not only employs these advanced computational methods but also systematically evaluates their performance in diverse market conditions. Many existing studies focus on individual models or specific market segments, lacking a holistic approach that integrates various predictive techniques to tackle the multifaceted nature of stock price movements.

Addressing this gap requires a detailed examination of how machine learning models, neural network models used in deep learning, transformer models and the technique of transfer learning can be applied for stock price forecast and prediction. This study is involved in developing and evaluating various models for advantages and limitations of each model, and building a foundation model that harnesses a collective potential. This approach promises to bridge the current research gap, offering more reliable and adaptive tools for financial market forecasting and contributing significantly to the field of financial analysis.

1.3 Research Aims and Objectives

This research primarily focuses on advancing the field of stock price prediction by combining neural networks used for deep learning, transformer models and the technique of transfer learning. This endeavor seeks to construct a comprehensive predictive model that not only surpasses the performance of traditional financial models but also addresses the complexities and dynamics of modern financial markets. To achieve this overarching goal, the research is guided by several specific objectives:

➢ Evaluate Existing Models: Critically assessing statistical, machine learning and neural network, transformer models in predicting stock prices. This involves evaluating the model performances on training and testing data and applicability to different stock patterns.
➢ Integrate Advanced Techniques: To analyze and interpret the integration and combination of neural networks within and with transformer layers, identifying how these techniques can complement each other to enhance predictive accuracy. This includes examining various algorithms and their suitability for analyzing financial time series data.

➢ Incorporate Transfer Learning: To estimate the potential of pre-trained model under the technique of transfer learning in financial forecasting, particularly its ability to adapt knowledge from one domain or market to improve predictions in another by filling gap in existing literature.

➢ Develop a Comprehensive Framework: To design and implement a robust predictive framework that synergizes machine learning, deep learning, and transfer learning. This framework should be capable of handling the nonlinearities and unpredictability of stock market data, providing reliable forecasts that can aid in decision-making processes.

➢ Evaluate Performance: Evaluating the performance of the discussed algorithms and techniques across different market scenarios and time periods. This includes assessing its accuracy, reliability, and adaptability in comparison to existing models.

By fulfilling the above mentioned, this work aims to deliver valuable interpretations and tools to the field of financial analysis, aiding investors, analysts, and policymakers in navigating the complexities of stock markets with greater confidence and foresight.

1.4 Research Questions

This research targets to address several pivotal questions that underpin the investigation into enhancing stock price prediction through the use of machine learning algorithms, Neural networks for deep learning, transformers and transfer learning techniques:

➢ How do traditional financial models compare to machine learning, neural networks,
transformers and deep learning techniques fare in prediction accuracy for stock prices?
  o This helps in identifying the traditional methods, machine learning models and neural networks and establish a base model for transfer learning.

➢ In what ways can machine learning algorithms be optimized to better capture the complex dynamics of the stock market?
  o This question explores the customization and refinement of machine learning algorithms to enhance their ability to analyze and predict market movements more effectively.

➢ What are the specific benefits and challenges of applying neural networks like RNNs, Transformers and GRU, LSTM, Bi-LSTM a networks, in the context of financial time series forecasting?
  o This question deals with the capability and efficiency of neural networks and transformer models for handling the temporal nature of time series stock data.

➢ How can transfer learning be effectively applied to stock price prediction, and what are the implications for model adaptability across different markets and conditions?
  o This question deals with the ability of pre-trained models under transfer learning to bolster the accuracy of predictive models by leveraging knowledge from various domains.

➢ What is the impact of integrating machine learning, neural networks for deep learning, Transformers and transfer learning techniques on overall performance and reliability of stock price predictions?
  o This question aims to assess the synergistic effects of combining multiple advanced computational methods on the predictive capabilities of the resulting framework.

Addressing these questions will provide a proper interpretation of the existing challenges and opportunities in financial forecasting, helping developing better predictive models for time series data. The evaluation of several deep learning algorithms, and their hybrid models along with transformer layers combined with technique of transfer learning combined in stock price prediction is also addressed.
1.5 Significance and Contributions of the Study

This study stands at the forefront of financial forecasting by integrating machine learning algorithms, neural networks for deep learning, transformer and the technique of transfer learning to improve the accuracy of prediction in stock price. Its significance lies in its potential to revolutionize how market analysts, investors, and financial institutions make informed decisions by providing a more reliable and nuanced understanding of market dynamics. By transcending the limitations of traditional financial models, this research contributes to the field by offering a comprehensive framework that adapts to the complexities and inherent unpredictability of the financial markets.

The contributions of this study are manifold. Firstly, it provides an in-depth comparative analysis of traditional and advanced predictive models. Secondly, the research explores the application of transfer learning technique in stock price prediction, a relatively uncharted territory that promises significant improvements in model adaptability and performance across different market conditions. Furthermore, by synthesizing machine learning and deep learning methodologies, the study unveils a novel predictive model that leverages the best of both worlds, setting a new benchmark for accuracy and reliability in stock price forecasts.

Ultimately, this study not only advances the academic discourse in financial analysis but also offers practical tools and insights that can be leveraged by the financial industry to navigate the ever-evolving market landscape more effectively.

1.6 Scope and Delimitations

The scope of this work encompasses building and evaluating an advanced predictive framework for stock price forecasting, leveraging the integration of machine learning algorithms, Neural network based deep learning, Transformers and transfer learning techniques. The research focuses on the application of these computational methods to analyze historical stock market data, aiming to bolster the accuracy of stock price prediction. Key areas of investigation include the assessment of traditional financial models, optimization of machine learning algorithms, application of neural networks and their hybrid models with transformer layers to time-series data, and exploration of transfer learning for enhanced model adaptability.
1.6.1 Scope

➢ Comparative Analysis: The study includes a comprehensive comparison between traditional financial models and advanced computational techniques to highlight improvements in predictive performance.
➢ Algorithm Optimization: It involves the customization and parameter tuning of machine learning algorithms and Neural networks algorithms, transformers to suit the specific nuances of stock market data.
➢ Data Utilization: The research leverages extensive historical stock price data containing stock data of nearly 500 companies with seasonality and trends to enrich the predictive models.
➢ Model Integration: The study aims to integrate various predictive models to create a robust framework that capitalizes on the strengths of individual approaches.

1.6.2 Delimitations

➢ Market Focus: The research may focus on specific stock markets or indices, limiting its immediate applicability to other markets or financial instruments without further adaptation.
➢ Data Availability: The study is confined to publicly available data, which may exclude certain proprietary or granular financial data that could enhance model predictions.
➢ Time Horizon: The predictive models developed in this study are designed for short to medium-term forecasts, with the understanding that long-term stock market predictions are inherently more uncertain due to the multitude of influencing factors.
➢ External Factors: While the study may consider external factors like economic indicators and market sentiment, it acknowledges the challenge of quantifying all potential influences, such as unforeseen geopolitical events, which may impact market dynamics.

By delineating these scopes and delimitations, the study sets clear boundaries for its investigative pursuits while acknowledging the constraints within which it operates. The study also accounts for the analysis of seasonality and trends present in the data using statistical analysis and appropriate algorithms.
1.7 Overview of Methodological Approaches

This research employs a CRISP-DM approach (Cross-Industry Standard Process for Data Mining) shown in Figure 1 throughout the work to explore the effectiveness of stock price prediction using machine learning algorithms, Neural Networks, Transformers and transfer learning techniques.

The methodology is structured to re-evaluate the models with parameter tuning and offers flexibility through iterative process to get satisfied evaluation results.

➢ Research Design: The study adopts a quantitative research framework, utilizing empirical data to model, evaluate, and refine predictive models. A comparative analysis approach under evaluation is employed to examine the efficacy of various predictive techniques, ranging from traditional machine learning algorithms to advanced deep learning and transfer learning methods. This design facilitates a thorough comparison, underscoring the capabilities and limitations of each technique and shedding light on their applicability in stock market forecasting.

➢ Data Collection: Central to the research is the acquisition of extensive and representative datasets, to use for training and validation during training process. The
study leverages historical stock price data based on literature survey with seasonality and trends. This data is collected from an open-source community and pre-processed for model input layers fitting to the objectives of the research, encompassing a wide array of market conditions and trends.

➢ **Data Analysis:** This involves analysis of the stock data, applying a variety of statistical techniques like correlation and null hypothesis.

The crucial part of this methodology is the innovative use of advanced analytical techniques to sift through complex datasets, identify underlying patterns, and forecast future stock price movements. This approach not only improves accuracy but also provides a customizable framework for investors and market analysts to perform day to day trading and market movements.

Furthermore, the research examines the integration of these predictive models into broader financial forecasting frameworks, addressing the technical and practical challenges of implementation and offering guidelines for their effective use.

In essence, this methodology is carefully used for evaluation of advanced predictive techniques in stock price forecasting. It aims to deliver valuable insights and actionable strategies to improve investment decisions and risk management in the stock market, contributing to the field of financial analytics and forecasting.

### 1.8 Outline of the Proposed Framework

The proposed framework is designed to bolster stock price prediction by integrating machine learning algorithms, Neural Networks used for deep learning, Transformers and techniques of transfer learning. At its foundation, the framework employs a rigorous data collection process, sourcing historical stock prices, financial indicators, to create a rich, multifaceted dataset. This dataset is then preprocessed to ensure data quality and relevance, including normalization, handling missing values, and feature selection to optimize the input for predictive modeling.

The core of the framework is characterized by a multi-layered analytical approach. Initially, linear regression and optimally tuned version of Decision Trees are applied to establish
baseline predictions. The framework then advances to incorporate deep learning models, including RNNs and LSTMs, Bi-LSTMs and Transformers specifically tailored to capture the sequential nature of stock market data and uncover deep-rooted patterns within it.

A novel aspect of this framework is the inclusion of transfer learning approach, which enable the adaptation of pre trained model on one domain to perform on target domain, delivering reasonable performance with less training across diverse market conditions.

The predictive models undergo rigorous validation and testing to assess their accuracy and reliability. This comprehensive framework not only aims to surpass the predictive capabilities of conventional models but also provides a dynamic tool for investors and analysts, offering deeper insights and more accurate forecasts of stock price movements.
Chapter 2 - Literature Survey

2.1 Overview of Stock Price Prediction Challenges

Predicting stock prices can be lot more challenging with technical analysis of charts, primarily due to volatility and the complex dependencies of financial markets on lot of factors. One of the foremost challenges is the market's inherent volatility, where prices are influenced by an unpredictable mix of economic indicators, company performance data, geopolitical events, and investor sentiment. This makes the task of forecasting prices akin to navigating a landscape that is constantly reshaped by external factors [32].

Another significant challenge is the high degree of noise within financial data. Stock markets generate vast amounts of data daily, much of which may be irrelevant or misleading for predictive purposes. Filtering out this noise to identify meaningful patterns requires sophisticated analytical tools and techniques [29].

CAPM and EMH assumes that all available information is already reflected in stock prices, suggesting limited potential for predictive models to outperform the market average. This theory underscores the difficulty of achieving consistent predictive accuracy [32].

Additionally, the non-linear relationships between influencing factors and stock prices complicate the modeling process. Traditional linear models may not capture these complex dynamics, necessitating advanced deep learning approaches along with transformers that can interpret non-linearity and temporal dependencies [21].

Lastly, the rapid evolution of market conditions means that models must continually adapt to remain relevant, posing a challenge for the development of robust and adaptable predictive frameworks. There is a also need to identify the seasonality and trends in the stock data and apply suitable statistical analysis and algorithms and exploring transfer learning approach to achieve reasonable accuracy of the stock price data with little training by using pre-trained models.
2.2 Review of Traditional and Modern Financial Models

The landscape of financial modeling for stock price prediction spans from traditional models grounded in economic theory to modern computational approaches that leverage advanced analytics. Traditional financial models, such as the Capital Asset Pricing Model and the Efficient Market Hypothesis, rely on the foundational assumption that markets are rational and that prices reflect all available information. These models emphasize the relationship between risk and return, but often fall short in capturing the complexities of real-world market dynamics and investor behavior [26].

On the other hand, modern financial models have embraced machine learning algorithms and Neural Networks to analyze vast datasets, uncover patterns that were previously indiscernible. Techniques such as Linear Regression, still considered somewhat traditional and basic, have been taken over by more sophisticated algorithms like Decision Tree and Neural Networks under Deep Learning. These modern approaches excel in handling the complex relationships inherent in financial data, offering improved predictive capabilities [22].

Neural Networks under Deep Learning like RNNs, GRUs, LSTM and Bi-LSTMs, networks, represents frontier of modern financial modeling. These models are suitable for sequential processing of input data and capture long range dependencies, making them suitable for time-series data analysis. By capturing temporal dependencies and learning from the historical progression of stock prices, deep learning models provide a nuanced understanding of market trends [6].

The evolution from traditional to modern financial models marks a significant shift towards data-driven, algorithmic approaches in stock price prediction, offering enhanced accuracy.

2.3 Role of Machine Learning in Financial Markets

The role of machine learning in financial domain is revolutionary in the analysis and prediction by introducing sophisticated algorithms capable of deciphering complex patterns within vast datasets. Unlike traditional financial models, which often rely on linear assumptions and historical averages, machine learning techniques can adaptively learn from data, capturing non-linear relationships and subtle market signals that may elude conventional analysis [12].
Primary application of machine learning algorithms in financial markets is in risk management and forecasting, where it's used to determine investment returns and risk management, identify market trends, and assess investment risks. Algorithms such as Random Forests which are improvised version of decision tree, Gradient boosting models offer powerful tools for classification and regression tasks, providing insights into future market movements based on historical data [19].

Machine learning also enhances algorithmic trading strategies, enabling the development of models that can execute trades at optimal times based on learned market patterns. This capability not only increases the efficiency and speed of trading operations but also opens new avenues for investment strategies that can dynamically adjust to changing market conditions [11].

Furthermore, machine learning contributes to risk management by analyzing vast arrays of financial instruments and market conditions to pinpoint potential risks. Techniques like clustering and anomaly detection help in identifying unusual market behaviors, which could signify emerging risks or opportunities.

In customer analytics, machine learning personalizes financial services, tailoring products and advice to individual investor profiles by analyzing past behaviors, preferences, and market responses [30].

Overall, machine learning's role in financial markets is transformative, offering a blend of predictive power, operational efficiency, and personalized financial solutions, thereby reshaping how financial entities approach investment strategies, risk management, and customer engagement.

2.4 Deep Learning Techniques for Time-Series Analysis

Various Neural network architectures under deep learning have emerged as particularly potent tools for time-series analysis, especially in domains like finance where sequential nature of data plays a critical role. These techniques leverage neural network architectures designed to process and make predictions based on temporal data, capturing intricate patterns traditional
models might miss [13]. Recurrent Neural Networks are basic deep learning approach for time-series analysis. RNNs are uniquely capable of retaining information from long range of inputs in the sequence. This feature makes RNNs well-suited for analyzing stock prices, currency exchange rates, and other financial time series, enabling them to forecast values based on underlying trends.

Long Short-Term Memory networks, an advanced variant of RNNs, address the limitations of traditional RNNs, particularly their difficulty with vanishing gradient problem. LSTMs can remember information for longer range, making them highly effective for financial time-series analysis where long-term patterns and short-term fluctuations both hold predictive value.

Bidirectional LSTMs (BI-LSTMs) extend LSTM capabilities by processing data bidirectionally. This bidirectional processing allows the model to capture the features of the input data from previous and upcoming data points around the current data point, enabling a more comprehensive grasp of patterns in the time series [20].

Gated Recurrent Units (GRUs) are another variation, offering a simpler alternative to LSTMs with fewer parameters, making them faster to train without significantly compromising the model's performance. GRUs maintain the ability to process long sequences effectively, making them another valuable tool for time-series interpretation and analytics.

These neural networks under deep learning, with their ability to learn from and predict complex, non-linear patterns in time-series data, have become indispensable for advanced financial analysis, offering nuanced insights and more accurate forecasts.

### 2.5 Transfer Learning in Financial Forecasting

Transfer learning technique can become a transformative approach in time series forecasting especially in financial domain, enabling pre-trained models trained on different domain and apply it to another domain with little fine tuning. This methodology is particularly advantageous in financial markets, where data can be scarce, expensive to acquire, or non-stationary, making it challenging to build robust models from scratch [3].
In the context of time series forecasting, transfer learning allows for the pre-training model on different dataset which can be large from related domains and then fine-tuned on the target domain. This will help the fine-tuned model achieve reasonable accuracy with little training from available limited datasets. For instance, a model trained on broad stock market data can be fine-tuned to predict prices for niche markets or individual stocks, leveraging the general patterns learned during pre-training to achieve higher accuracy.

Transfer learning also addresses the issue of non-stationarity in financial data, where market conditions and influencing factors change over time. Models can be quickly adapted to new conditions without the need for extensive retraining, making financial forecasting more responsive to market dynamics.

Moreover, this approach facilitates the inclusion of diverse datasets, such as investor and retailer sentiment, and social media trends, greed and fear indexes into models. By transferring knowledge from these varied sources, models can capture a more comprehensive view of the factors influencing market movements.

Transfer learning not only reduces the heavy training requirements and computational resources required to build high performing forecasting models but also opens up new possibilities for predictive analytics in finance, enabling more nuanced and adaptable forecasting solutions [35].

2.6 Comparative Analysis of Predictive Models in Finance

Various models are developed and their comparative analysis is crucial for identifying the most effective approaches for predicting market movements and making informed investment decisions. This analysis involves evaluating various models based on their predictive accuracy, adaptability to market changes, computational efficiency, and ability to handle the vast and complex nature of financial data.

Traditional statistical models like ARIMA (Auto Regressive Integrated Moving Average) have been staples in financial forecasting, valued for their ability to model time series data and volatility patterns. However, their linear nature, assumption of stationary data often limit their effectiveness in capturing non-linear and dynamic behavior of financial markets [31].
Machine learning models, which are quite famous for classification tasks such as Decision Trees, introduce non-linear modeling capabilities and the ability to handle large datasets with multiple variables. These models can uncover complex relationships between various market indicators and stock prices, offering improved predictive performance over traditional models [23].

Neural Network models under deep learning, including RNNs, GRUs, LSTM, Bi-LSTMs and represent the cutting edge in time series forecasting. Their strength lies in processing sequential data, capturing long-term dependencies, and automatically detecting relevant features from raw data, making them exceptionally suited for analyzing time-series financial data.

The comparative analysis also considers the integration of transfer learning techniques, which can significantly enhance the adaptability and performance of predictive models by leveraging pre-trained networks on diverse datasets.

By systematically comparing these models, one can identify the most suitable models and ways for specific financial forecasting tasks, balancing between accuracy, complexity, and computational demands [10].

2.7 Integrating Multiple Data Sources for Enhanced Predictions

Historical stock prices and financial statements form the backbone of most predictive models, offering insights into a company's performance and market trends. However, these traditional data sources often provide a limited view of the potential factors affecting stock movements. By incorporating additional data types such as inflation, greed and fear index, investor and retailer sentiment, analysts can gain a deeper understanding of the macroeconomic conditions that impact financial markets [15].

Integrating multiple data sources is a pivotal strategy in enhancing the accuracy and robustness of financial predictions. This approach capitalizes on the diverse facets of information that influence market dynamics, going beyond traditional financial indicators to include a broader spectrum of data types.
Furthermore, market sentiment analysis, derived from news articles, social media posts, and analyst reports, adds a qualitative dimension to the predictive models. Sentiment analysis tools can process natural language to gauge the market sentiments. Integrating this sentiment data helps capture the psychological and behavioral aspects of the market, providing a more nuanced view of potential price movements [5].

The approach shown in the Figure 2 is used in this work which starts with business problem understanding, data preparation, modelling evaluation and then integrating multiple datasets with different approaches and techniques to further improve the results.

Alternative data sources, such as satellite imagery, supply chain information, and web traffic data, also offer unique insights into company performance and industry trends. For instance, satellite images of parking lots can indicate retail activity levels, while web traffic data can provide early indicators of consumer interest in a company's products.

By synthesizing these varied data sources, predictive models can achieve a more holistic analysis of the factors driving financial markets. This integration leads to enhanced prediction accuracy, enabling investors to execute more reasonable trades and investments based on a multi-dimensional analysis of market conditions [24].
2.8 Advances in Algorithmic Trading Strategies

Advances in algorithmic trading strategies have significantly transformed the landscape of financial markets, introducing levels of speed, efficiency, and complexity previously unattainable. Central to these advancements are sophisticated algorithms that leverage mathematical models and computational techniques to execute trades at optimal times, based on predefined criteria or real-time market analysis.

One notable advancement is the use of machine learning algorithms, neural networks and transformers in developing predictive models that can analyze vast datasets to identify market trends and price patterns. These models enable algorithms to forecast market movements with greater accuracy, allowing for the execution of more strategic trades. Techniques such as reinforcement learning with human feedback, where algorithms learn optimal policies through feedbacks, have been particularly impactful, enabling the development of self-adjusting strategies that improve over time [25].

High-frequency trading (HFT) represents another significant advancement, where algorithms execute a large number of orders at extremely high speeds, often in milliseconds or microseconds. HFT strategies leverage market inefficiencies and short-term price discrepancies to secure profits, relying heavily on advanced computational infrastructure and ultra-low latency data feeds [17], [9].

Furthermore, sentiment analysis algorithms have gained traction, analyzing news over articles and social media to identify market sentiment and predict its impact on stock prices. This allows traders to incorporate qualitative data into their strategies, adding a layer of market psychology analysis to traditional quantitative models [16].

Quantitative and statistical models, including pairs trading and mean reversion strategies, have also evolved, benefiting from more sophisticated risk management and optimization techniques. These strategies identify and exploit statistical relationships between financial instruments, adjusting their positions based on mathematical models of market behavior [33].

Overall, the advances in algorithmic trading strategies have not only enhanced the ability of traders to execute complex strategies but also increased market liquidity and efficiency.
However, they also raise challenges in terms of market regulation, fairness, and the potential for systemic risks. Exercising caution while employing these algorithms to practical applications is advised.

2.9 Evaluation Metrics for Financial Models

Evaluation metrics are used in judging the performance and effectiveness of trained models on testing data, providing quantitative measures to gauge their predictive accuracy, risk management capabilities, and overall reliability. Key metrics are tailored to the specific objectives of the model, whether it's predicting stock prices, assessing portfolio risk, or evaluating the profitability of trading strategies.

For predictive models focusing on stock prices or market trends, metrics such as RMSE, Accuracy, MAE and MSE are commonly used. These metrics calculates the average magnitude of the errors between predicted and true values. The Mean Absolute Percentage Error (MAPE) offers a normalized measure, making it easier to interpret the model's performance in relative terms.

In classification tasks, such as predicting market movements (up or down) precision, classification report, confusion matrix, Sensitivity, and the F1 are pivotal under positive and negative classes. Precision measures the number of true positives among the total positives identified by the model, recall checks for the model's ability to identify the positive classes and minimize false negatives, and the F1 Score includes both false positives and false negatives using harmonic mean between precision and recall, useful in scenarios where both metrics are important [28].

For models involved in risk assessment or portfolio management, metrics like Value at Risk and Conditional Value at Risk assess the possibility of resulting in loss and the expected loss in worst-case scenarios, respectively. These metrics are crucial for understanding and managing financial risk.

Additionally, metrics such as the Sharpe Ratio and Sortino Ratio are essential for evaluating returns for investment strategies using risk management, showing the return earned in excess for risk taken per unit.
Choosing the right evaluation metrics is necessary for accurate results, guiding improvements, and ensuring that the model meets its intended objectives effectively.

2.10 Transformer Models in Financial Forecasting

Transformer models have revolutionized the field of financial forecasting by offering an advanced framework capable of handling complex sequential data with unprecedented efficiency. At the core of Transformer models lies the multi-head attention mechanism which facilitates for positional encoding enabling the positional importance of each token in the input sequence. This feature is particularly beneficial for financial forecasting, where the relevance of historical data can vary significantly over time. The adaptability and scalability of Transformer models make them ideal for analyzing vast datasets common in financial markets, enabling them to capture subtle patterns and dependencies that traditional models might overlook.

The Time Series Transformer (TST) model represents a specialized adaptation of the Transformer architecture, specifically developed for time series data unlike text generation tasks. TST enhances the original Transformer by incorporating domain-specific modifications that better align with the temporal nature of financial data, such as incorporating time-related features. This allows for interpreting time series trends and cycles, offering potential improvements in forecasting accuracy and reliability.

GPT-2, developed by OpenAI, extends the Transformer's capabilities into generative tasks, demonstrating exceptional performance in understanding and generating human-like text. In the context of financial forecasting, GPT-2 can be leveraged to analyze and predict market sentiment by processing vast amounts of textual data from social media and financial news. Its features primarily lies in contextual understanding and generating predictive text makes it a powerful tool for anticipating market movements based on emerging trends and public sentiment.

Together, Transformer models, TST, and GPT-2 represent a cutting-edge approach to financial forecasting, combining deep learning's power with specific adaptations for time series analysis. Their collective strengths offer promising directions for researchers and practitioners seeking to enhance predictive accuracy in the complex and dynamic domain of financial markets.
Chapter 3 - Methodology

3.1 Research Design and Strategy

In this research, the design and strategy is to focus on advancing stock price prediction through machine learning, Neural Network under deep learning, transformers and transfer learning techniques structured in systematically exploring, developing, validating predictive models. This involves a quantitative approach, leveraging empirical data to build and test the efficacy of various algorithms in forecasting stock prices.

Initially, the research design incorporates a thorough literature review to identify existing models, their limitations, and potential areas for innovation. This foundational step ensures that the study builds upon current knowledge while addressing gaps in the field.

Subsequently, the study adopts an experimental research strategy, where different predictive models are developed and rigorously tested under controlled conditions. This involves the collection of a comprehensive dataset, including historical stock prices for building and evaluating the models.

Models are evaluated based on predefined evaluation metrics, such as RMSE, accuracy MAE, MSE which are quite suitable for time series data prediction models, to determine their predictive capabilities. The comparative analysis of these models allows for a mathematical assessment of the models interpretation across various metrics in the prediction of stock data.

This research design and strategy facilitate a systematic exploration of advanced computational techniques in financial forecasting, aiming to contribute novel insights and practical solutions to the field.

3.2 Data Collection and Pre-Processing

Appropriate dataset is crucial and preprocessing of the data are crucial steps for training the models for stock price prediction. The process begins with historical stock data prices for various companies, including opening price of the stock, 24 hour high, and 24 hour low price, 24 hour trading volume for specific ticker and of course, closing price of the day for the stock.
This financial data is often supplemented with market indicators such as interest rates, inflation rates, and economic indices to provide a broader context for the stock market's behavior.

Additionally, alternative data sources like market sentiment, and company earnings reports are collected to incorporate qualitative insights into market sentiment and potential external influences on stock prices.

Once collected, the data preprocessing is done to ensure its quality and usability. This step includes cleaning the data by appropriately dealing with missing values, either by removing them or replacing them with statistical metrics, dealing with outliers or errors. The cleaned data is further normalized or scaled to bring different scales to a comparable level, facilitating more effective analysis by predictive models.

Feature selection and extraction are also critical, where relevant variables are identified and sometimes transformed or combined to better represent the underlying factors that influence stock price movements. Now the model is trained on this featured and cleaned, scaled, normalized data according to the input layer with appropriate sizes for training, testing and validation.

### 3.3 Model Selection and Development

Model selection and development in stock price prediction involve choosing and tailoring algorithms that can effectively capture and analyze the complex patterns within financial markets. The selection process starts by considering a range of models from simple linear regressions, which are useful for their interpretability, to more complex optimized Decision Trees known for handling non-linear relationships and feature interactions.

Deep learning models, particularly RNNs, GRUs, LSTM, Bi-LSTM networks, are chosen for their ability to capture long range dependencies in the input, well-suited for time-series analysis in stock price prediction.
The development phase involves configuring these models with the appropriate hyperparameters, training them on historical data, and iteratively refining them based on their performance. This includes adjusting learning rates, regularization terms, and network architectures to optimize the models' predictive accuracy while preventing overfitting.

The selected models are then rigorously tested and validated using separate datasets to ensure their robustness and generalizability to unseen data, ultimately aiming to develop a reliable predictive framework for stock price forecasting.

3.4 Implementation of Transfer Learning Techniques

The implementation of transfer learning techniques in stock price prediction involves leveraging pre-trained models from related domains to enhance forecasting accuracy without starting from scratch. This process begins by selecting a pre-trained model that has been effectively trained on different but related dataset which can be larger, potentially a different stock data compared to the target stock data in this work. The intuition is that these models have learned generalizable patterns or features that can be applicable to financial time series data.

The next step is to load the pre-trained model and perform fine-tuning on the target stock for specific task of stock price prediction and forecasting. This involves adjusting the model's final layers to tailor its output to financial predictions, while possibly freezing the earlier layers to retain the learned features. The model is then trained on target financial data, allowing it to adapt its learned representations to the nuances of stock market data.

This approach of fine tuning a pre-trained model helps in reducing the training on the target dataset and reduce computational expense for training while potentially improving the model's performance by incorporating broader learned patterns. Transfer learning thus offers a cost-effective and efficient method for enhancing predictive models in finance.

3.5 Validation and Performance Evaluation Methods

Validation and performance evaluation of predictive models in stock price forecasting involve critical methods to ensure accuracy and reliability. Train dataset is sometimes further split for
validation by using techniques like k-fold cross-validation, where the validation data is made into 'k' parts with 'k-1' parts used for training and remaining part for testing, cycling through all parts to ensure comprehensive evaluation. This technique helps mitigate overfitting and provides a more generalized performance metric. Hold-out validation, where dataset is split for test and train with straightforward approach reserving a portion of the data for testing, allowing for evaluation of unbiased model predictiveness on new data. Evaluation metrics MAPE, MAE, RMSE and are used to understand the model accuracy.

These methods and metrics collectively offer a appreciable framework for validating and judging performance of financial forecasting models, ensuring their practical applicability and effectiveness in predicting stock prices.

3.6 Block Diagram

The following procedure shown in Figure 3 is adopted in this work with collection of data preprocessing, visualization, model training, and prediction of target stock values.

![Proposed Block Diagram](image)

Figure 3 Proposed Block Diagram [14]
3.7 Limitations and Assumptions

In any research work, it is necessary to understand and accept the limitations and assumptions that underlie the study. When dealing with the problem of advancing stock price prediction under current techniques discussed, several key limitations and assumptions should be considered:

➢ Data Availability: One significant limitation is the availability and quality of historical financial data. Assumptions are made that the collected data accurately represents market conditions and that any missing or erroneous data can be effectively handled through preprocessing techniques.

➢ Stationarity Assumption: Many models assume stationarity in financial time series data, under null hypothesis which may be rejected later where market dynamics can change over time. This assumption could limit the statistical models like ARIMA in capturing non-stationary trends.

➢ Market Trends and Unseen Events: Predictive models often assume that past market behavior will repeat in the future. However, unforeseen events and market shocks can disrupt these patterns, challenging the model's prediction.

➢ Transfer Learning Assumption: Application of this technique, an assumption is made that knowledge gained from pre-trained models is transferable and relevant to financial data. The effectiveness of this assumption depends on the choice of base models and the similarity of underlying patterns between domains.

➢ Risk Factors: The study may not consider all possible risk factors that can influence stock prices, and assumptions might be made regarding the significance of certain variables.

Recognizing these limitations and assumptions is crucial for interpreting the results of the study and understanding the boundaries within which the predictive models operate. This research also takes into account the limitations on the computational power that can be available for building a model.
3.8 Summary of Methodological Approaches

The methodological approaches employed in advancing stock price prediction using the algorithms and techniques discussed encompass a comprehensive and systematic framework. The research design blends quantitative analysis with empirical data, leveraging historical financial data and market indicators to build and evaluate predictive models. These models range from traditional statistical approaches to cutting-edge deep learning networks, with a focus on capturing the complex patterns within time series data related to financial domain. Transfer learning technique is integrated to enhance model adaptability and performance, leveraging knowledge from pre-trained models in related domains. The research places a strong emphasis on data preprocessing, ensuring the quality and relevance of input to the models. Evaluation metrics encompass a range of statistical measures, including MAE, RMSE, MSE to assess predictive accuracy. Acknowledging limitations and assumptions is fundamental, particularly regarding data availability, stationarity assumptions, and the transferability of knowledge from other domains. In summary, the methodological approaches adopted in this study aim to provide a holistic and rigorous examination of stock price prediction, combining diverse techniques to contribute valuable insights to the field.
Chapter 4 - System Design and Implementation

4.1 Data Features and Selection Criteria

Data features and selection criteria are pivotal considerations in the development of predictive models for stock price forecasting. The choice of relevant features plays a fundamental role in the model's ability to learn from the training data and able to interpret the complexity in the input and deliver better results.

Key data features encompass a wide range of variables, including historical stock prices (open, close, high, low), trading volumes, market indices (e.g., S&P 500), economic indicators (e.g., interest rates, GDP growth), and potentially sentiment data from news articles and social media. Additionally, alternative data sources such as satellite imagery, web traffic data, and supply chain information may be included. The selection criteria for these features are based on their relevance to stock price movements and their ability to enhance predictive accuracy. Feature engineering techniques may involve transforming variables, creating lagged variables to capture time dependencies, and selecting the most informative features through statistical tests or feature importance rankings.

The aim is to strike a balance between including a feature variable that encompass complexity while avoiding too much of multicollinearity. The selected features should collectively provide a holistic representation of the factors that influence stock prices, facilitating the model's ability to deliver better prediction results.

Ultimately, careful consideration of data features and selection criteria is essential for building predictive models that can effectively analyze and forecast stock price movements, contributing to more informed investment decisions in financial markets.

4.2 Data Preprocessing and Normalization

These are critical steps in preparing datasets for predictive modeling in any machine learning task. These will make sure data is consistent and suitable for analysis by machine learning or deep learning algorithms.
Data preprocessing involves several key tasks, such as dealing with missing values by removing them or replacing with statistical averages, dealing with outliers, and skewness. Missing values can be imputed using statistical averages, while outliers, which can skew model predictions, can be dealt by removing them or capping them using percentiles. Data imbalances, where certain classes or data points are underrepresented, may require techniques like oversampling or under sampling to create a balanced dataset.

Normalization is essential for bringing all features to a common scale, preventing variables with larger ranges from dominating the model learning during training. Min-Max scaling, which scales features to a specified range, and Z-score normalization, distributes data around average value of 0 and standard deviation as 1.

Both data preprocessing and normalization are vital for improving the model's learning during training, ensuring that it can effectively learn during training and make predictions on testing dataset better.

4.3 Application of Machine Learning

Machine learning in stock price forecasting represents a pivotal advancement in the financial industry. These models leverage historical data, market indicators, and economic factors to make predictions about future stock price movements.

Primary applications involves in the development of predictive models that can forecast stock prices with varying degrees of accuracy. Algorithms such as Linear Regression, Decision Trees and other improved variants of decision tree are commonly employed to predict patterns and seasonality in financial data. Deep learning models, including RNNs, GRU, LSTM, Bi-LSTM networks, are specifically designed for retaining long range dependencies and for predicting stock prices.

Machine learning models are also used for risk assessment, portfolio optimization, and algorithmic trading. These applications involve sophisticated algorithms that analyze market conditions in real-time, identify opportunities or risks, and make trading decisions autonomously.
Furthermore, sentiment analysis algorithms process social media and financial news and company earnings to gauge market sentiment and assess its impact on stock prices. This qualitative analysis supplements quantitative models, providing a more comprehensive understanding of market dynamics which can considered for future work.

In summary, machine learning offers various features in dealing with financial markets including risk management technique using classification models, risk free investment strategies, optimal times for trading and day trading opportunities.

4.3.1 Model Training and Parameter Tuning

Model training involves optimizing machine learning algorithms by feeding them historical financial data to learn patterns and relationships. During this process, models adjust their internal parameters to minimize prediction errors. Parameter tuning, a critical step, fine-tunes these internal settings to enhance a model's understanding of the data. Techniques like grid search used in this work tries out various combination of parameters for the model and gives out the optimized parameters that the model can run on given data. This iterative training and tuning process ensures that predictive models achieve their highest accuracy and generalizability when applied to stock price forecasting tasks.

4.3.2 Model Integration and Ensemble Techniques

Model integration and ensemble techniques in stock price forecasting involve combining multiple predictive models to enhance overall accuracy and reliability. These involves trying out different neural network layers based on the advantages they provide and can produce robust predictions.

Ensemble methods, such as Bagging and Boosting, create diverse sets of models by training them on different subsets of data or with varied hyperparameters. These models are then combined to provide a collective prediction, often yielding improved results compared to single models.
In addition, model integration can involve the fusion of predictions from various neural networks under deep learning models and transformers, each specializing in different aspects of stock price prediction. This hybrid models provide more comprehensive view of market dynamics.

Ensemble techniques used in Random Forest involves ensemble numerous decision trees with information gain or Gini index by picking samples randomly and then ensemble the optimal trees.

Model integration and ensemble techniques contribute to more robust and stock price prediction, aiding investors in risk free investment strategies.

4.4 Deployment of Deep Learning Models

The usage of neural networks deep learning models in stock price forecasting involves the implementation of trained neural networks for live data predictions and trading in financial markets. After rigorous training and validation, models, such as RNNs, LSTM, and GRUs, hybrid model are used.

These are typically deployed with model integration into trading platforms or financial systems, allowing for the seamless flow of data from data sources for seamless integration of real time data to model market information for continuous analysis. Once deployed, deep learning models continuously process incoming data, making predictions about future stock price movements. These predictions can inform trading decisions, risk management strategies, and portfolio adjustments in real-time. The ability of deep learning models to capture and adapt to changing market conditions makes them valuable tools for automated trading and risk assessment.

4.4.1 RNN and LSTM Application

The deployment of the above two models in stock price forecasting involves the implementation of trained neural networks for market analysis and investment strategies in financial markets. After rigorous training and validation, these are deployed to production. The deployment process involves similar to general deep learning models by integrating them with
data sources or trading platforms or financial systems, allowing for the seamless flow of data between data sources and the neural network. This makes sure that the model receives real time data of market information for seamless learning and analysis.

Once deployed, deep learning models continuously process incoming data, making predictions about future stock price movements. These predictions can inform trading decisions, risk management strategies, and portfolio adjustments in real-time. The ability of models to cope with trends and seasonality in the data and adapt to changing market conditions makes them valuable tools for automated trading and risk assessment.

Overall, the deployment of deep learning models in stock price forecasting empowers financial professionals with data-driven insights and automation capabilities, enhancing decision-making and potentially improving trading performance in dynamic financial markets.

### 4.4.2 Application of BI-LSTM and GRU Models

The application of Bi-LSTM and GRU models in stock price prediction is a testament to these models ability in handling time-series data. BI-LSTM and GRU models are specialized variants of RNNs designed to handle vanishing gradient and with improved learning ways.

BI-LSTM models, by processing data bidirectionally, excel in capturing long-range dependencies in stock price time series. They are well-suited for tasks where understanding the context of past and future data is crucial, enabling more accurate predictions of stock price trends.

On the other hand, GRU models offer a balance between complexity and computational efficiency. They are designed to capture essential temporal information while requiring fewer parameters than traditional LSTM models. GRUs are particularly valuable when computational resources are limited.

Both BI-LSTM and GRU models excel in capturing the nuanced patterns and dependencies inherent in stock price data, making them powerful tools for accurate stock price forecasting and enhancing decision-making in financial markets. Their adaptability to various time-dependent scenarios further solidifies their relevance in the field of finance.


4.5 Transfer Learning Model Application

Transfer learning technique harnesses the power of pre-trained models from one domain and tailors them to address specific challenges in another, such as financial forecasting, including stock price prediction. The process involves fine-tuning pre-trained models to adapt their learned features in patterns of financial time series data.

In stock price prediction, transfer learning offers several advantages. Pre-trained models, often derived from extensive datasets in different domains like image or text processing, capture high-level features that can be valuable for recognizing complex patterns in financial data. By modifying the model's final layers to align with the forecasting task and retraining it on financial data, transfer learning allows for efficient model adaptation.

This approach significantly reduces the size of training data while potentially improving the model's accuracy reasonably with lesser training due to pre training acquired in the source domain. This technique is a cost-effective and efficient strategy for enhancing predictive models, contributing to more accurate stock price forecasts and informed decision-making in the financial sector.

4.6 Performance Metrics and Model Comparison

Stock price prediction through machine learning, neural networks and transfer learning techniques, several performance metrics helps in evaluating the effectiveness of predictive models. These can provide insights into the model's ability and robustness, helping investment and trading strategies in the financial domain.

Important evaluation metrics that are used in time series modelling techniques include MAE, MSE, RMSE. These evaluation metrics are discussed along with their mathematical representations.
Mean Squared Error (MSE)

MSE is a popular evaluation metric, which gives the mathematical values of average of squared values of differences between actual and predicted values.

Lesser the MSE values more accurate the model is, with predictions closely aligning with actual values.

\[ MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y})^2 \]

Root Mean Squared Error (RMSE)

RMSE scales MSE back to original units of data, providing better interpretable error metric of the model. It is mathematically calculated by using the 2\(^{nd}\) root of Mean Square error and helps in ensuring model is not skewed by outliers

\[ RMSE = \sqrt{MSE} = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} (y_i - \hat{y})^2 \]

Mean Absolute Error (MAE)

MAE provides another perspective on a regression model’s accuracy. It is calculating mathematically by the average of modulus of difference between actual and predicted values. This metric provides accuracy for the larger values better to compared to other two due to non-squaring.

\[ MAE = \frac{1}{N} (\sum_{i=1}^{N} |y_i - \hat{y}|) \]
Chapter 5 - Implementation

5.1 Data Set Description

The dataset [34] provided for this project, titled "all_stocks_5yr.csv," encompasses a comprehensive collection of stock market data over a five-year period. This dataset is expected to contain key financial indicators relevant to stock market analysis, including:

- **Date:** The specific trading dates, offering a chronological timeline of the stock data.
- **Open:** Indicating the opening price of stock on that day.
- **High:** This represents the highest price point that the stocks reached during the trading day.
- **Low:** Conversely, this reflects the lowest price point of the stocks on each day.
- **Close:** The closing price of stocks, which is critical for understanding the daily market closure status.
- **Volume:** This denotes the number of shares traded during the day, providing insights into the trading activity and liquidity of the stocks.
- **Name:** The stock ticker symbols, identifying the specific companies to which the stock prices correspond.

This dataset is pivotal for conducting comprehensive time series analyses, exploring trends, seasonality, and other patterns in stock prices. Its depth and breadth make it an invaluable resource for applying and testing the various statistical analysis, machine learning algorithms, and neural networks, transformers proposed in this work.

5.2 Information about the Data

The dataset, "all_stocks_5yr.csv," is a substantial collection of stock market data, encompassing a total of 619,040 entries. It spans a five-year period, providing a rich historical perspective on stock market trends and behaviors. The data is structured into seven distinct columns, each capturing key aspects of stock market activity. These include date information, opening price, 24 hour high price, 24 hour low price, and closing price of the day of stocks, along with the trading volume and the names of the stocks (identified by their ticker symbols). The comprehensive nature of this dataset, with its extensive range of entries and detailed stock-related information, makes it an ideal resource for in-depth analysis and modeling in our
project. The granularity and volume of the data are well-suited to apply sophisticated statistical and machine learning techniques, offering a fertile ground for exploring and developing advanced stock price prediction models.

5.3 Feature Description

In this work, we're analyzing comprehensive dataset containing the stock prices of 505 different companies over five years. Each day's data includes the date, opening price of the stock, 24 hour high and low, closing price of stock, trading volume generated on that day, and the company's ticker symbol. However, due to limited computing resources, we're focusing our study exclusively on Apple Inc., symbolized as "AAPL" as shown in Figure 4. This approach not only simplifies our analysis but also gives us a chance to closely examine the financial trends of a major player in the tech industry.

<table>
<thead>
<tr>
<th>date</th>
<th>open</th>
<th>high</th>
<th>low</th>
<th>close</th>
<th>volume</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 2013-02-08</td>
<td>15.07</td>
<td>15.12</td>
<td>14.63</td>
<td>14.75</td>
<td>8407500</td>
<td>AAL</td>
</tr>
<tr>
<td>1 2013-02-11</td>
<td>14.89</td>
<td>15.01</td>
<td>14.26</td>
<td>14.46</td>
<td>8882000</td>
<td>AAL</td>
</tr>
<tr>
<td>2 2013-02-12</td>
<td>14.45</td>
<td>14.51</td>
<td>14.10</td>
<td>14.27</td>
<td>8126000</td>
<td>AAL</td>
</tr>
</tbody>
</table>

Figure 4 Records of the Dataset

Let's break down the key features of the dataset:

- **Date**: This is the date on which the stock was traded. It's crucial as it allows us to perform a time series analysis and observe trends over a period.
- **Open**: This represents the opening price at which stock trades at start of day. This is a floating-point number that's key for understanding how the market reacts at the start of the trading day, often influenced by overnight news or events.
- **High**: This is the peak price of the stock during the trading session. It's another float value, and it's significant because it shows the maximum potential the stock reached on any given day, reflecting peak investor confidence.
- **Low**: This denotes the day's lowest stock price. Also a floating-point number, it indicates the moments when the market sentiment was not so optimistic.
➢ Close: Closing price of the stock at which the last trade of the day was made. It's crucial because it serves as the final value of the stock for the day and is often used as a reference point for the next day's opening price.

➢ Volume: This is an integer value indicating the total number of shares traded during the day. It gives us an idea of the stock's liquidity and investor interest. High trading volumes can point to strong sentiment, while low volumes might indicate a lack of interest or contentment with the current positions.

➢ Name: This is the ticker symbol, a string that uniquely identifies the traded company. It's important for distinguishing among various companies in the dataset.

Understanding these features in detail is essential for a thorough analysis of the stock market trends, especially when focusing on a specific company like Apple Inc.

5.4 Inspection of Missing Values

When certain fields or records in a dataset lack data, these are identified as missing values. Such gaps in data often result from mistakes during the collection, processing, or recording stages. Addressing these missing values is vital because they can lead to inaccurate analysis and outcomes in modeling. To manage this issue, typical approaches include filling in these gaps with estimated values (imputation), removing records with missing data (deletion), or applying methods that can effectively handle missing data.

The displayed output in Figure 5 shows the number of missing values (either NaN or null) in each column of a Data Frame. For example, there are 11 instances where the 'open' column is missing data, meaning in 11 rows the opening price isn't recorded. In the same way, both 'high'
and 'low' columns have 8 missing entries each, showing that in 8 rows, the data for the highest and lowest prices are not available. Conversely, the columns 'date', 'close', 'volume', and 'Name' all have 0 missing values, indicating complete data in these areas. The term 'dtype: int64' at the end indicates that these missing value counts are expressed as integer numbers. The outliers in the data are reasonable and are maintained for the complexities in the modelling predictability.

5.5 Correlation Matrix

![Correlation Matrix](image)

Correlation matrix shown in figure 6 describes the correlation coefficients between the variables in the dataset. A mask is applied for the upper diagonal of the matrix to avoid the upper triangle of the matrix. The coefficients of this matrix usually range from -1 to 1.
For stocks dataset the following observations can be made from the above correlation matrix for the stock price dataset:

1. Strong Positive Correlations:
   - The coefficient between 'open', 'high' is 0.999184, indicating a strong positive correlation, suggesting that when the opening price increases, the high price tends to increase as well.
   - The similar trend of strong positive correlation is seen between ‘close’ and ‘high’ suggesting that closing price tends to increase with the highest price.
   - The same relation can be drawn between ‘close’ and ‘open’ showing a strong positive correlation, implying that if the open price of the stock is high, closing price of stock will also be tending to be higher which is usually the general case.

2. Negative Correlations:
   - The correlation between ‘volume’ and ‘close’, indicating a negative correlation, suggesting that when the volume is more, the closing price will be less.

3. Low Correlations:
   - The correlation coefficient between 'volume' and 'daily return' is 0.059308, indicating a weak positive correlation, implying that there is a slight tendency for higher trading volume to be associated with slightly higher daily returns.

4. Self-Correlations:
   - The matrix is masked to hide the upper triangle but the diagonal elements will be showing a value of 1.

There may be other relationships between the variables which are complex but not seen in the matrix and the low correlation does not mean the variables are not related and complex relations between the variables can exist.

The variables that can be used for training and testing the model include volume, open, low, high as dependent variable and the close variable as independent variable as per the correlation matrix.
5.6 Top 5 Stock Tickers by Average Trade Volume

The dataset has been refined to improve its structure and clarity and each stock is renamed as "ticks" to calculate the average volume as shown in Figure 7.

- The column formerly named "Name" has been renamed to "ticks" for better representation.
- Any null or missing values present have been removed. This is particularly beneficial for time-series analysis, ensuring data consistency.
- The date column has been converted into a 'datetime' datatype, making time-series operations more straightforward.
- The dataset now has 619,029 entries distributed across seven columns, with no missing values.

Upon analyzing the average volume of stocks traded for each company, it was identified that there are 505 unique ticker symbols in our dataset. The top 5 companies with the highest average traded stock volume are:

1. BAC (Bank of America)
2. AAPL (Apple Inc.)
3. GE (General Electric)
4. F (Ford Motor Company)
5. FB (Facebook, Inc.)
6. MSFT (Microsoft Corporation)
7. AMD (Advanced Micro Devices, Inc.)
8. MU (Micron Technology, Inc.)
9. INTC (Intel Corporation)
10. CSCO (Cisco Systems, Inc.)

Insight: These top 5 companies represent a mix of sectors, from technology to finance to automotive. Their high average trading volumes indicate these companies’ significance in the stock market, reflecting strong investor interest and activity. Such stocks are often pivotal in portfolios, as their movements can influence broader market trends.

5.7 Closing Stock Price for Top-5 Stocks over 5 Years

The visual representations offer a detailed view of the 5-year closing stock prices for five top companies are shown in Figure 8. Key observations include:
○ BAC (Bank of America) and F (Ford Motor Company) stand out with their unique red backgrounds, in contrast to the green used for other companies, highlighting a visual differentiation based on their ticker symbols.
Each chart illustrates the trend in closing stock prices, with special notes marking the peak closing price within the 5-year timeframe.

AAPL (Apple Inc.) demonstrates remarkable growth, with its highest closing price significantly surpassing its initial value. This trend mirrors the company's consistent innovation and leadership in the technology sector.

GE (General Electric) and F (Ford) haven't experienced growth on par with some tech companies, despite their established market presence.

Tech giants like FB (Facebook, Inc.), MSFT (Microsoft Corporation), and MU (Micron Technology) have exhibited substantial growth, particularly Facebook and Microsoft, which attained impressive highs.

The semiconductor industry's competitive landscape is evident in the stock performances of AMD (Advanced Micro Devices) and INTC (Intel Corporation).

CSCO (Cisco Systems) offers insights into the network and communication equipment sector, with its price changes reflecting broader market trends.

These visualizations highlight the stock market's dynamic nature, especially the significant growth potential in the technology sector. The noted peak performances of each stock enable an effective comparative analysis over the observed period.

5.7 Seasonality Decomposition

Understanding and interpreting the seasonality trends in the time series data such as stocks is quite useful in comprehending the patterns and dynamics of the stock behavior. Various events can

5.7.1 Seasonality Decomposition for FB

FB (Facebook) stock can be decomposed into key factors affecting its performance. Firstly, financials, encompassing revenue growth, advertising metrics, and cost management, significantly impact its value. Market sentiment, influenced by user engagement trends and platform innovations, plays a pivotal role. Regulatory actions and privacy concerns can impact user trust and ad revenue. Economic conditions, such as digital ad spending trends and consumer behavior, also shape FB's performance.
Moreover, competition within the social media and tech sector, as well as global political developments, can affect the stock’s trajectory.

- **Observed**: This depicts the actual observed stock prices for FB. We can observe the stock’s journey, reflecting its growth and various market dynamics over the period.
- **Trend**: The trend component provides a smoothed representation of the stock’s overarching direction. For Facebook, the trend has been predominantly upward, signifying steady growth as shown in Figure 9. However, unlike Apple, there are more pronounced peaks and troughs, indicating periods of rapid growth followed by consolidation or slight declines. The upward trend indicates market confidence and the company's continual growth in its sector.
- **Seasonal**: This component captures cyclical patterns in the stock price. For Facebook, as shown in Figure 9, there is evident seasonality with recurring patterns observed throughout the years. These could be influenced by factors such as advertising revenue cycles, major updates or changes to the platform, or other cyclical events pertinent to Facebook.
- **Residual**: The residual component represents the unpredictable fluctuations in the stock price after removing the trend and seasonality. It's the "noise" in the stock's movement, capturing anomalies and unexpected events, perhaps reactions to news or unforeseen market dynamics.
Facebook's stock, like Apple's, exhibits clear trends and seasonality. The pronounced peaks in the trend component might correspond to significant company events, product launches, or other market dynamics. The seasonality might be associated with the company's business cycle or the tech sector's general dynamics. Understanding these components can be invaluable for investors or analysts trying to predict future stock movements or understand past behaviors.

5.7.2 Partial and Auto Correlation for AAPL

Partial and Auto Correlation functions are essential tools in variable correlation analysis. ACF measures the linear correlation between observations at different time lags, helping to identify repeating patterns or seasonality. It actually denotes the correlation between two variables at different points of time in our case. Meanwhile, PACF isolates the correlation between observations at a specific lag, removing the influence of intervening observations. Both ACF and PACF are pivotal for understanding time-dependent structures within data, aiding in the selection of appropriate models and parameters for time series forecasting, where auto correlation plot helps to identify the auto regressive term and partial correlation term can help in identifying the Moving Average term required for statistical analysis.

The following denotes the mathematical representations for the partial and autocorrelation respectively.

Auto Correlation

\[
\rho(k) = \frac{\frac{1}{n-k}\sum_{t=k+1}^{n}(y_t - \bar{y})(y_{t-k} - \bar{y})}{\sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \bar{y})^2} \sqrt{\frac{1}{n-k}\sum_{t=k+1}^{n}(y_{t-k} - \bar{y})^2}}
\]

Partial Auto Correlation

\[
\tilde{y}_t = \phi_{21}\tilde{y}_{t-1} + \phi_{22}\tilde{y}_{t-2} + e_t
\]

The sample Auto Correlations in the single variable context along with its lagged versions for different orders can be calculated using auto correlation and similarly Partial Auto Correlation can be obtained as series using the Partial Auto Correlation using the respective formulae.
The ACF plot shown in the Figure 10 and PACF plot shown in Figure 11 provide insights into the time dependency structure of a time series.

Let’s break down the insights for Apple’s (AAPL) stock:

- **ACF Plot:**
  - The ACF plot provides correlations of the series with its lags. In the case of AAPL, several lags are significantly different from zero, implying that the stock price today is influenced by its past values.
  - The slow decay in the autocorrelation values indicates a strong trend component in the series. This is consistent with the upward trend observed in the decomposition plot.
The presence of significant autocorrelations at multiple lag values suggests that there might be some seasonality or repetitive patterns in the data, though this is not as pronounced as the trend.

- **PACF Plot:**
  - The PACF plot gives the partial correlation of single variable with respect to other variable while keeping one or more variable controlled. Here time series with shorter lags are controlled and partial correlation is drawn out for the time series. This helps identify the extent of the lag in an autoregressive model.
  - The PACF plot shows a sharp cut-off after the first few lags, indicating that the stock price is influenced by its recent past values but not much by the values before that.
  - The significant spikes at certain lags hint at potential seasonality or patterns in that are not seen or interpreted solely by the trend plot shown.

The ACF and PACF plots suggest that AAPL's stock prices are influenced by their recent past and have a strong trend component. The plots provide valuable insights when considering autoregressive models for forecasting or understanding the stock's behavior. For modeling purposes, the significant lags in the PACF might suggest the order of an AR (Autoregressive) model, while the ACF can hint at the order for a MA (Moving Average) model.

### 5.7.3 Partial and Auto Correlation for FB

The performance of FB (Meta Platforms, Inc.) stock can be decomposed into several key factors. Firstly, its financial health, including metrics like revenue growth, advertising effectiveness, and cost management, significantly influences its value. Market sentiment is another crucial factor, heavily reliant on user engagement trends, the success of platform
innovations, and the company's strategic moves. Regulatory actions and privacy concerns can impact user trust and advertising revenue, adding a layer of complexity to FB’s performance. The performance of FB (Meta Platforms, Inc.) stock can be decomposed into several key factors.

Firstly, its financial health, including metrics like revenue growth, advertising effectiveness, and cost management, significantly influences its value. Market sentiment is another crucial factor, heavily reliant on user engagement trends, the success of platform innovations, and the company's strategic moves. Regulatory actions and privacy concerns can impact user trust and advertising revenue, adding a layer of complexity to FB’s performance.

The performance of FB (Meta Platforms, Inc.) stock can be decomposed into several key factors.

![Partial Autocorrelation for FB](image)

**Figure 13 Partial Auto Correlation for FB**

Firstly, its financial health, including metrics like revenue growth, advertising effectiveness, and cost management, significantly influences its value. Market sentiment is another crucial factor, heavily reliant on user engagement trends, the success of platform innovations, and the company's strategic moves. Regulatory actions and privacy concerns can impact user trust and advertising revenue, adding a layer of complexity to FB’s performance.

The ACF and PACF plots shown in Figure 12 and Figure 13 shows the time dependency of a time series with seasonality and non-stationary nature. In the context of Facebook’s (FB) stock:

- **ACF Plot:**
  - Just like AAPL, ACF plot drawn for FB indicates presence of several lags significantly different from zero. This means the FB stock price today is influenced by its past values, suggesting a certain level of autocorrelation.
  - The slow decline in the autocorrelation values indicates a strong trend component in the series. This is consistent with our previous observation of FB's stock showing a general upward trend over time.
  - The extended range of significant autocorrelations suggests a potential seasonal or repetitive pattern in the data, although this isn't as sharp as the overarching trend.
• PACF Plot:
  o The PACF plot gives the partial correlation of single variable with respect to other variable while keeping one or more variable controlled. Here time series with shorter lags are controlled and partial correlation is drawn out for the time series. This helps identify the extent of the lag in an autoregressive model.
  o For FB, the PACF plot showcases significant correlation spikes at the initial lags, which then taper off. This suggests that while recent past values influence the stock price, the effect diminishes for older values.
  o The presence of significant correlations at select lag values could indicate potential seasonality or cyclical patterns, though these seem to be more subtle when compared to the trend.

Facebook's stock prices, like Apple's, show patterns of autocorrelation, influenced strongly by recent past values. The ACF and PACF plots are instrumental when considering modeling techniques like ARIMA for forecasting, as they can guide the choice of parameters. The significant lags in PACF might indicate the order for an AR (Autoregressive) model, while the ACF hints at the potential order for a MA (Moving Average) model.

5.8 Comparative Analysis of 7 Tech Stocks

The visual comparison of average daily stock prices of seven leading technology companies over five years reveals the following insights as shown in Figure 14:

![Comparative Analysis of 7 Tech Stocks](image)

Figure 14 Comparative Analysis of 7 Tech Stocks
• Apple Inc. (AAPL) shows a strong upward trend, indicating solid market performance. Notably, there was a substantial rise in its stock value between 2016 and 2017, a period emphasized in the visualization.

• Facebook (FB) displays a growth pattern similar to Apple's, particularly in the same timeframe, underscoring the dominance of major tech firms in the stock market.

• Both Micron Technology (MU) and Advanced Micro Devices (AMD), key players in the semiconductor sector, exhibit fluctuating prices. However, they also demonstrate growth phases, especially noticeable for Micron Technology around 2016-2017.

• Intel Corporation (INTC) maintains a more consistent price level with slight variations, reflecting its stable presence in the market.

• Microsoft (MSFT), as an established figure in the tech industry, shows a trend of growth, mirroring its ongoing innovation and adaptability in the market.

• Cisco Systems (CSCO) presents a trend that, while somewhat variable, generally remains stable, indicative of its well-established role in the networking and communications field.

• The span from February 2016 to December 2017, specifically marked in the chart, is of particular interest. This phase marked significant growth for multiple tech firms, most notably Apple, Micron Technology, and Facebook.

The visualization effectively highlights the dynamic nature and growth potential of the technology sector, accentuating the significant role these tech giants play in the stock market. The resilience and expansion potential of the tech industry, particularly during the boom phase encapsulated in the highlighted period, are clearly demonstrated, underscoring the sector's impact on global financial markets.

The plot shows the significant trend in the increase for the all the stocks over the years. The tech stocks of FB and AAPL has increased drastically and maintained significant gap with the other five stocks which is evident from the plot.

This helps in selecting the target stock for prediction and modelling analysis. Although there is no thumb rule that highest performing stocks should be taken into account for price modelling. We can also choose low performing stocks for analysis and modelling but the high performing stock is subject to more Fear and Greed in the market usually with fluctuations in the returns of the stock.
5.9 Overview of Stocks Over 5 Years

The horizontal bar chart shown in Figure 15 provides an insightful comparison of stock price growth for ten prominent companies over five years, yielding these observations:

- **Apple Inc. (AAPL)** and **Advanced Micro Devices (AMD)** stand out with remarkable growth rates exceeding 300% and 500%, respectively. This highlights Apple's innovation-led market leadership and AMD's rising prominence in the semiconductor sector.

- Close behind are **Micron Technology (MU)** and **Microsoft Corporation (MSFT)**, each showing nearly or above 200% growth, indicative of their consistent progress in their sectors.

- **Facebook, Inc. (FB)** also exhibits strong growth, approaching the 200% mark. This reflects its expanding reach and diversification in revenue, reaffirming its status among tech leaders.

- In contrast, **General Electric (GE)** and **Ford Motor Company (F)** are shown in red, denoting negative growth. GE's struggles span multiple business areas, while Ford contends with shifts in consumer tastes and industry trends.

- **Bank of America (BAC)**, **Intel Corporation (INTC)**, and **Cisco Systems (CSCO)** display modest growth, consistent with their stable positions in the market.

- The chart uses color coding for quick visual reference: green bars indicate positive growth, and red bars represent declines. Each bar is annotated with specific growth percentages, facilitating an easy comparison of company performances.

![Figure 15 Overview of Stocks over 5 Years](image-url)
The dominance of the technology sector is clear, as most tech firms exhibit substantial growth over the five-year period. Conversely, traditional industries like automotive and conglomerates have encountered difficulties, reflecting the dynamic and ever-changing landscape of the global market.

5.10 Daily Return of Top 5 Stock

The visual data shown in Figure 16 illustrates the daily financial returns of three key companies over a five-year period, providing the following insights:

- **Market Fluctuations:** The charts reveal varying degrees of volatility among the companies. For example, AAPL (Apple) and AMD (Advanced Micro Devices) show significant price swings, typical of the often-volatile tech industry.
Highest Returns: Each company's top daily return is marked with a green dashed line. Notably, AMD's chart shows one of the highest peaks, reflecting its potential for high returns but also high risk.

Lowest Returns: Marked by red dashed lines, the lowest returns highlight the most challenging days for these stocks. GE (General Electric) and Ford Motor Company, for instance, show some pronounced dips, pointing to specific struggles they faced during those times.

Return Distribution: Histograms give a visual representation of return frequencies. A wider distribution, like AMD's, suggests greater unpredictability, whereas a tighter distribution, seen in stocks like INTC (Intel), suggests more stability.

Average Returns: Indicated by yellow dashed lines in the histograms, the mean daily returns help gauge overall profitability. Positive averages, particularly in the cases of AAPL and MSFT (Microsoft), suggest a generally profitable trend. Detailed annotations provide exact figures for a more thorough comparison.

Impact of External Events: Certain sharp increases or decreases in the charts may be linked to external factors, such as product launches, financial reports, or global economic shifts.

While tech stocks like AAPL, MSFT, and AMD demonstrate notable growth and potential gains, they also come with inherent volatility. Conversely, established corporations like GE have shown less favorable daily returns, mirroring their operational challenges. These insights are crucial for investors to align their strategies with their risk tolerance and investment timeframe.

As discussed before, the high performing stocks tend to draw lot of attention resulting in more volume, liquidations, high sell and buy pressures from the retail investors due to fear of missing out with greed resulting in the volatility which is true for FB and APPL stocks. The spikes in the daily returns happens when there is high fear of missing out and greed and the lows in the graphs can be due to sell pressure when there is heavy greed which can be well interpreted by the market sentiments better. Also, the stocks of AMD, MSFT also tend to show some volatility due to popularity among the retail investors. Intel has stable returns due to less popularity on the streets of the market among the retail users and shows a stable graph. The graphs for the other stocks can also be drawn for similarly and various conclusions can be drawn in similar way.
5.11 Comparative Analysis of Stock Prices

A line plot for comparison of the top tech stock prices is shown in the graph form in Figure 17. The following can be observed from the graph plot:

- 'FB' stock stands out as notably expensive among the seven tech stocks analyzed as shown in Figure 17.
- Conversely, 'AMD' emerges as comparatively inexpensive to purchase compared to others.
- 'FB' and 'AAPL' exhibit higher volatility, evident from the chart, distinguishing them as more volatile than the remaining stocks.

5.12 Algorithm Implementation

This section delves into various algorithms that can be employed to find insights about the time series, its dependencies and seasonality and trends lying the data. Understanding the patterns in the stock prices data is important for forecasting, anomaly detection, and investment strategies in various industrial sectors.
5.12.1 ARIMA Model

The ARIMA (Auto Regressive Integrated Moving Average) model is a leading method for time series analysis, adeptly capturing the complexities of time-dependent data. This model skillfully combines three essential elements: the Auto Regressive (AR) portion, which utilizes past data points; the Integrated (I) section, employing differencing to achieve data stationarity; and the Moving Average (MA) aspect.

ARIMA stands for Auto Regressive Integrated Moving Average. This model is statistical approach of estimating market movement and sentiment by calculating differences between different time points rather than from the actual. This model can be completely understood from below:

Auto Regression (AR): This helps in finding the relationship between a current observation with respect to set of lagged observations and uses lagged points to predict future values. The ‘p’ in ARIMA is for Auto regression and 18 value is taken in this work from auto correlation plot.

Integrated (I): The ‘d’ parameter in ARIMA refers to number of differencing steps. The time series usually assumed to be stationary. If not, the d value indicates the number of differencing terms required to make the data stationary.

Moving Average (MA): The q value in ARIMA refers to moving average. This reflects the relationship between a data point and the residual error applied to lagged observations.

In the AR part, defined by the parameter 'p', the model predicts future values based on past observations. The 'I' component, denoted by 'd', indicates how many times the time series needed to be differencing to make it stationary, a key step for the model's effectiveness and precision. The MA segment, labeled as 'q', incorporates historical forecast errors as part of the prediction process, offering a corrective dynamic for upcoming forecasts.

The ARIMA model's intricate integration of these components makes it exceptionally adept at interpreting time series data, identifying trends both short and long terms in data. Its adaptability makes it applicable across various fields, from finance to healthcare to academia,
offering critical insights and predictions. The challenge lies in choosing values of p, d, and q specific nuances of the time series data, often using techniques like grid search and cross-validation for optimization. When properly parameterized, the ARIMA model performs as a powerful and dependable tool for effective time series analysis.

The ARIMA model information, coefficients and diagnostic results shown in Figure 18 can be interpreted as follows:

![Figure 18 Model Architecture for ARIMA](image)

### Model Information

In this analysis, we employed an ARIMA model configured with the parameters (18, 1, 2), which indicates the following specifics:

- It includes 18 autoregressive terms.
- The dataset underwent a single differencing step to ensure stationarity.
- The model incorporates 2 moving average terms.
- For model estimation, a blend of the conditional sum of squares (CSS) approach and maximum likelihood estimation (MLE) was utilized.
• This model was applied to a dataset consisting of 1,258 data points.

Parameter Coefficients

• The table of coefficients provides the values of the AR and MA terms. The corresponding `P>|z|` values are the p-values associated with each term, which test the null hypothesis assuming the data is stationary.

• For instance, the AR term `ar.L1.D.close` has a coefficient of -0.3904 and a p-value of 0.015. The p value less than 0.05 shows that the null hypothesis cannot be valid and time series is not stationary.

• Parameters with high p-values (e.g., > 0.05) might not be statistically significant and could potentially be removed to simplify the model. For example, `ar.L3.D.close` has a p-value of 0.860, indicating it may not be significant.

Model Diagnostics

• The roots of the ARIMA model are provided in the table at the end. All of the modulus values of the roots are greater than 1. If any roots had a modulus equal to or less than 1, the model would be non-stationary and would not be valid.

Accuracy Metrics

• Mean Absolute Error (MAE): An MAE of 1.52 suggests that average distance for the predicted and actual values is 1.52.

• Root Mean Squared Error (RMSE): An RMSE of 1.5606 implies that the model was off by 1.5606 units on average. This metric penalizes larger errors more than the MAE.

• Mean Squared Error (MSE): The MSE is 2.4357, which is simply the square of RMSE.

Summary

The chosen ARIMA (18, 1, 2) model seems to be quite complex with a lot of AR terms. While some terms are statistically significant, others might not be adding much value to the model.
This could potentially lead to overfitting, indicating that the model is performing well on training but failing to identify the complex patterns in the test data or unseen data. The accuracy metrics suggest that the model's predictions are reasonably good. However, it would be beneficial to compare these metrics with other potential models or forecasting methods to determine the best approach.

5.12.2 SARIMA Model

The Seasonal ARIMA (SARIMA) model is an advanced variation of the ARIMA model, which can handle time series with seasonality. SARIMA can integrate both types of seasonality, offering a robust approach for analyzing and predicting data with periodic patterns. The SARIMA model is typically represented as SARIMA(p, d, q) (P,D,Q)s, where the non-seasonality are denoted by (p, d, q), and the seasonality are (P,D,Q), with 's' seasonality cycle length.

In this model, SARIMA(p, d, q), functions similarly to the regular ARIMAs, addressing non-seasonal trends and patterns. Conversely, the seasonal section, (P, D, Q)s, is specifically tailored to manage the seasonality of the data. The autoregressive (AR) component (P) analyzes the correlations of an observation with its preceding seasonal points.

Seasonal differencing (D) is employed to achieve stationarity in the seasonal data, thereby eliminating seasonal trends. The seasonal moving average (MA) portion (Q) considers previous errors in seasonal forecasts. SARIMA's ability to simultaneously deal with non-stationary and seasonal data makes it highly versatile and effective in analyzing complex time series.

By encompassing both seasonal and non-seasonal dynamics, SARIMA emerges as a potent tool for forecasting in areas where seasonality is significant, including sectors like retail, finance, and meteorology. The seasonality in the data facilitates the uses of SARIMA model in further analysis by finding p, d, q. If the data is seasonal then d will be 1. p and q can be taken from the Auto correlation and partial correlation plots. With appropriate tuning of its parameters, SARIMA can adeptly capture and predict the nuanced patterns in seasonal time series data, proving to be an invaluable resource in forecasting endeavors.
The SARIMAX model information, coefficients and diagnostic results shown in Figure 19 can be interpreted as follows:

**Model Information**

- The model used is a SARIMAX (Seasonal ARIMA with Exogeneous variables) model with parameters (18, 1, 2). This implies that:
  - 18 autoregressive terms are used.
  - The data has been differenced once to achieve stationarity.
  - 2 moving average terms are used.
  - The method used for fitting the model is the outer product of gradients (OPG).
  - The model was applied on a dataset with 1259 observations.

**Parameter Coefficients**

- The table of coefficients provides the values of the AR and MA terms. The corresponding `P>|z|` values are the p-values associated with each term, which test null hypothesis.
• For instance, the AR term \`ar.L1\` has a coefficient of -0.3934 and a p-value of 0.007. This suggests that the term is significant when 5% is the significance level.
• Parameters with high p-values (e.g., > 0.05) might not be statistically significant and could potentially be removed to simplify the model. For example, \`ar.L3\` has a p-value of 0.944, indicating it may not be significant.

Model Diagnostics

• The Ljung-Box test (Q) checks for white noise residuals. A p-value (Prob(Q)) close to 1.0 suggests the residuals are white noise, which is desirable. The p-value here is 0.95, which is quite high, indicating that information in the series is picked by the model quite well.
• The Jarque-Bera (JB) test checks normal distribution of residual in the time series data. A low p-value (Prob(JB)) means not normally distributed. Here, the p-value is 0.00, indicating the residuals may not be normally distributed.
• Heteroskedasticity (H) tests for constant variance in the residuals. A value significantly different from 1 suggests non-constant variance. Here, H is 2.25, which might indicate non-constant variance in the residuals.

Model Accuracy Metrics

• Mean Absolute Error (MAE): An MAE of 1.1562 on average the predicted values are 1.1562 units away from actual values.
• Root Mean Squared Error (RMSE): An RMSE of 2.4689 implies that the model was off by 2.4689 units on average. This metric penalizes larger errors more than the MAE.
• Mean Squared Error (MSE): The MSE is 6.0955.

In summary, the chosen SARIMAX(18, 1, 2) model seems to be quite complex with a lot of AR terms. While some terms are statistically significant, others might not be adding much value to the model. This could potentially lead to overfitting. The diagnostic tests suggest that while the model copes with the patterns of the data quite well (as shown in the Ljung-Box test), the residuals may not be normally distributed and could have non-constant variance. The accuracy metrics suggest that the model performance is reasonably close to the actual
values, but given the RMSE value, there might be occasional large deviations. Again, comparing these metrics with other models or forecasting methods can help in determining the best approach.

5.12.3 Linear Regression

Linear regression stands as a basic yet crucial predictive tool in stock price forecasting. It aims to find a line that best fits between the stock price and various independent variables like trading volumes, historical prices, or other financial indicators with minimum errors between actual value and predictions by the model.

The essence of this model is to create a regression line that best aligns with the data, striving to minimize the differences between actual and predictions made by the model. This method helps identify market trends and potential future price levels, offering essential insights for investors and analysts.

Linear regression lines are commonly used in technical analysis to determine optimal trading with technical analysis of top and bottom using the future price movements indicated by the model. Although linear regression performs reasonably well, one should recognize the limitations of linear regression in the context of stock price prediction.

Since the stock markets are influenced by variety of external factors, many of which might not have a linear relationship with stock prices the linear regression fails in that case. Additionally, the stock market is known non-stationary nature, which can challenge the foundational assumptions of linear regression models.

Despite these limitations, when linear regression is used carefully and combined with other analytical tools, it can provide valuable insights for forecasting stock prices. The dependent variables are volume, high, low, open and close is the target variable which is generally true for stock price prediction.
The train and test data are split between dependent and independent variables with 80% and 20% respectively in size for the models that follow.

```python
from sklearn.linear_model import LinearRegression

model = LinearRegression()
model.fit(X_train, y_train)

y_train_pred = model.predict(X_train)
y_test_pred = model.predict(X_test)

model_coef = model.coef_[0]
model_intercept = model.intercept_

model_coef, model_intercept
```

Figure 20 Linear Regression Model

**Model Overview**

A Linear Regression model shown in Figure 20 was developed to forecast AAPL’s closing stock prices, using the number of days since the dataset's inception as the primary variable.

**11.3.2 Model Parameters**

- The `days_since_start` coefficient is around 0.0636, indicating an expected increase of about $0.0636 in AAPL’s closing price for each additional day.
- The intercept is set at roughly 79.68, representing the predicted closing price of AAPL stock at the beginning of the dataset.

**Performance Metrics**

- The Mean Absolute Error (MAE) stands at 23.30, reflecting an average discrepancy of $23.30.
- The Mean Squared Error (MSE), at 619.85
- A Root Mean Squared Error (RMSE) of 24.90, similar to the MAE, quantifies the average error but emphasizes larger discrepancies more.
In summary, the linear regression model, train set is from 80% of the data size and validated on the 20%, effectively identified AAPL’s long-term price trend but struggled with short-term market variations. Although it accurately traced the general direction of AAPL stock prices, the substantial MAE and RMSE suggest limited efficacy for precise short-term predictions or intricate investment strategies. Given the stock market's complex and fluctuating nature, this underscores the need for more sophisticated modeling approaches and a broader set of features for more accurate predictions in stock price forecasting.

5.12.4 Decision Tree Regressor

The Decision Tree algorithm, a non-parametric tool used in supervised learning, is highly effective for both classification and regression tasks, such as predicting stock prices. In stock price forecasting, the decisions are taken homogenously with information gain or Gini Index.

This method is particularly beneficial for data with complex, non-linear relationships, thanks to its hierarchical nature, enabling it to identify intricate patterns within the historical stock data. It considers a range of features, including trading volumes, moving averages, and other technical indicators, to forecast stock price trends or movements.

Key advantages of using Decision Trees in stock price prediction is their straightforward and transparent nature. Each decision pathway from the tree's root to a leaf symbolizes a distinct decision-making rule, making the process easily interpretable. However, Decision Trees are prone to variability and can be overly sensitive to minor fluctuations in the dataset, potentially leading to various tree structures from similar data addressed by models like Random Forest, which combine the predictions from multiple Decision Trees and interpreting all such tree outputs into single output.

The decision tree is most commonly used for the classification tasks where each decision is taken with decreasing entropy and increasing Information gain going down the decision tree. However, the decision tree can be used for stock price prediction as shown in the subsequent discussions.
While Decision Trees offer an intuitive approach to forecasting stock prices, it's crucial to fine-tune and validate the model thoroughly to avoid overfitting and ensure its applicability to new, unseen data.

We employed a Decision Tree Regressor for our model shown in Figure 21, using grid search to optimize its hyperparameters. The optimal settings identified are:

- `max_depth`: Unlimited, allowing the tree to expand fully until it perfectly classifies the training data or hits another stop criterion.
- `min_samples_split`: Set to 2, indicating the least number of samples necessary for splitting a node.
- `min_samples_leaf`: 1, indicating 2 samples to leaf node for optimal performance.

**Overview of Model Performance**

The Decision Tree Regressor captures AAPL stock price trends, albeit with notable volatility. Predictions show abrupt changes, a common trait of decision trees in continuous numerical predictions. This happens as the tree divides the data into distinct segments based on features, leading to a stepped pattern in the forecasted values.
Evaluation Metrics

- Mean Absolute Error (MAE) stands at 24.5, predicted values are far from actual prices by an average of $24.5.
- Mean Squared Error (MSE) is 731.74, highlighting a heavier penalty on larger errors and suggesting some predictions were considerably inaccurate.
- Root Mean Squared Error (RMSE) is 27.05, offering a perspective on error magnitude in the same units as the stock prices.

Concluding Observations

Although the Decision Tree Regressor shows potential, its stepwise prediction pattern may not be ideally suited for smooth stock price forecasting. The model’s tendency to reflect volatility could be a sign of overfitting to training data. Exploring ensemble techniques like Random Forest or Gradient Boosted Trees could be advantageous, as they might average out these fluctuations and yield more consistent, smoother predictions.

5.12.5 RNN Algorithm

Recurrent Neural Networks (RNNs) are a specialized type of neural networks tailored for predicting sequences, which makes them ideal for time series analysis, such as forecasting stock prices. RNNs are unique because they have a memory-like capability, enabling them to retain information over range of data points which is critical for understanding and learning the time-based dependencies inherent in sequential data. At the heart of an RNN is a cell that processes input at each time step, delivers an output, and passes key information to the subsequent cell in the sequence. This process of recursive information transmission equips RNNs to capture and learn patterns over time, a vital aspect of making accurate predictions in time series.

Basic RNNs, however, can face challenges in holding information of long-term dependencies because vanishing or exploding gradients. To address these challenges, Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs) are proposed.
These improved architectures include gating mechanisms that manage the retain or forget information at each time step. This modification significantly enhances the neural network to process learn from extended sequences of data.

![Diagram of a Recurrent Network](image)

Figure 22 Basic Architecture of RNN [18]

In the realm of stock price forecasting, RNNs shown in Figure 22 and their advanced variants are quite suitable for modeling the sequential type of financial time series data. By learning the intricate patterns and dependencies in historical stock prices, these networks can provide forecast prices, aiding investors and traders for optimal trading strategies. Despite their potential, careful preprocessing of data, hyperparameter tuning, and model validation are essential to build effective and reliable forecasting models using RNNs.

RNNs are suitable for stock price forecasting because of their capability in picking temporal trends in the time series data. The layers of RNNs can account for sequential data inheriting previous state and account for predict next point in time series data based on the previous points. These are trained internally through back propagation technique leading to a reducing loss function internally through feedbacks. Sometimes the RNNs are subject to overfitting and regularization techniques like dropouts are used to prevent this scenario.

However, RNNs are subject to problem called vanishing gradient where the gradient vanishes inside the neural network of RNNs during the training data. This happens when the gradient during back propagation becomes exponentially smaller due to multiplications involved in the back propagation algorithm. This results in lesser capabilities in contextual learning and capturing long range dependencies.
Model Design and Structure

Our model employs a Recurrent Neural Network (RNN) framework shown in Figure 23 with Simple RNN layers. The model's structure includes:

Model: "sequential_99"

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple_rnn_111 (SimpleRNN)</td>
<td>(None, 60, 50)</td>
<td>2600</td>
</tr>
<tr>
<td>dropout_455 (Dropout)</td>
<td>(None, 60, 50)</td>
<td>0</td>
</tr>
<tr>
<td>simple_rnn_112 (SimpleRNN)</td>
<td>(None, 60, 50)</td>
<td>5050</td>
</tr>
<tr>
<td>dropout_456 (Dropout)</td>
<td>(None, 60, 50)</td>
<td>0</td>
</tr>
<tr>
<td>simple_rnn_113 (SimpleRNN)</td>
<td>(None, 50)</td>
<td>5050</td>
</tr>
<tr>
<td>dropout_457 (Dropout)</td>
<td>(None, 50)</td>
<td>0</td>
</tr>
<tr>
<td>dense_97 (Dense)</td>
<td>(None, 1)</td>
<td>51</td>
</tr>
</tbody>
</table>

Total params: 12,751
Trainable params: 12,751
Non-trainable params: 0

Figure 23 Model Summary of RNN Architecture

- An initial `Simple RNN` layer with 50 neurons, designed to output sequences for subsequent RNN layers.
- Each `Simple RNN` layer, a `Dropout` layer to prevent overfitting by randomly deactivating few inputs.
- A `Dense` layer for producing the final predicted value.
- Trainable parameters are 12,751.

Duration of Training

The training was conducted over 50 epochs, with each epoch taking about 1 to 3 seconds. This leads to an estimated total training duration of 50 to 150 seconds, roughly 1 to 2.5 minutes. Variations in time per epoch may result from factors like computational resource availability, background operations, or the complexity of the data processed in each batch.
Overview of Training

Throughout the 50 epochs, there was a consistent decrease in loss, as measured by Mean Squared Error (MSE), suggesting an enhancement in the model accuracy. The loss stabilized at approximately 0.0033 by the final epoch.

Evaluation Metrics

- Mean Squared Error (MSE) at 4.9196.
- Mean Absolute Error (MAE) at 1.6928, providing a straightforward average error value.
- Root Mean Squared Error (RMSE) at 2.2180, offering an error estimate in the same unit as the stock prices.

Concluding Thoughts

The RNN model shows promising results in stock price prediction, with an MAE suggesting an average deviation of about $1.69 from the true values. The consistent loss reduction during training further affirms its effectiveness. However, its real-world application hinges on performance with new data. Despite impressive metrics, accurately forecasting stock prices remains challenging due to the stock market's inherent unpredictability. The training time indicates the model's relative complexity yet efficient operability, making it suitable for quick, iterative forecasting in dynamic environments like stock markets.

RNN with Hyperparameter Tuning

Following an extensive hyperparameter optimization, the model achieved its peak performance with the following settings: 64 neurons in each Simple RNN layer, a dropout layer with rate of 0.2, learning rate at 0.001, and an extended training duration of 100 epochs with batch size of 32.

This fine-tuning led to a significant reduction in loss, reaching 0.00153615185909296, a marked improvement from earlier iterations. The model's structure consists of three Simple RNN layers, followed by dropout layer for avoiding overfitting, with dense layer for output predictions. Overall, the model comprises 20,801 trainable parameters.
The training of this refined model, spanning 50 epochs, demonstrated a steady decrease in loss, with each epoch averaging between 1 to 3 seconds. Consequently, the total training time ranged from about 50 to 300 seconds, or approximately 1.7 to 5 minutes. The model's final evaluation metrics after training were: a MSE of 4.6041, MAE of 1.5931, and a RMSE of 2.1457. These results show that the model, on average, about $1.59 off from the actual values. The prediction vs actual plot for the model is shown in the Figure 24.

![AAPL Stock Price Prediction](image)

Figure 24 RNN Prediction Plot

In conclusion, the meticulous process of hyperparameter tuning has substantially reduced error values in price prediction. The resulting model is not only more precise but also efficient in training time, balancing effectiveness with speed. This underscores the potential of utilizing a recurrent neural network with carefully selected hyperparameters for stock price forecasting. Nonetheless, considering the unpredictability of market forces and external variables in investment decision-making these models should be employed proactively and carefully.

5.12.6 Simple LSTM

Long Short-Term Memory (LSTM) networks, a specialized type of RNNs, exhibits in learning from sequential data by preserving long-term dependencies. Unlike traditional RNNs facing gradient issues, LSTMs effectively tackle vanishing and exploding gradients, ideal for time series forecasting like stock prices. Comprising memory cells with input, output, and forget gates, LSTMs regulate information flow:
- Input Gate: Controls incoming data storage in cell.
- Forget Gate: Retain or discard information from the cell.
- Output Gate: Governs amount of cell state output for subsequent layers.

LSTMs shown in Figure 25 excel in retaining and recalling long-term information, proving valuable for tasks reliant on context or sequence, such as predicting stock prices affected by past trends. In stock price forecasting, LSTMs process past price sequences to predict future values, adjusting weights during training to minimize prediction disparities. While potent, LSTMs necessitate meticulous tuning due to computational demands and susceptibility to overfitting with noisy financial data.

LSTMs are introduced over RNNs to hold long range dependencies in the input data which can designed to retain or forget with gating cells and memory cells along with Cell state, Input, Output and Forget gate with sigmoid activation function with the range of values from 0 to 1. The output gates uses sigmoid or hyperbolic activation function along with tanh function which ranges from -1 and 1. The forget gate also uses the sigmoid function for cell state which helps to maintain and discard the information. This cell state is major difference between RNNs and LSTM which helps in maintaining the value of gradient during propagation and prevent it from diminishing.
Thus, their implementation often involves combined strategies and validation techniques to ensure dependable and accurate stock price predictions, aiding in investment strategies.

![Figure 26 Simple Architecture of Simple LSTM](image)

**Model Architecture:**

The model comprises a Sequential structure, housing 3 LSTM layers shown in Figure 26 tailored for sequential data, aiming to capture long-term dependencies. Dropout layers follow each LSTM to curb overfitting, culminating in a single dense output layer suited for regression tasks, containing 1 neuron. With 83,009 trainable parameters, the model holds complexity.

**Training**

Utilizing a regression-type loss function, the model's convergence is evident, commencing at a loss of 0.0357 and steadily decreasing to 0.0011 by the 50th epoch. Although trained for 50 epochs, further training might enhance performance if overfitting is managed.

**Evaluation**

Metrics:

- MSE at 6.0911
- MAE 1.869
- RMSE at 2.4680 depict the model's predictive capability.
**Insights**

The model's complexity, demonstrated by 3 LSTM layers, suggests an ability to grasp intricate patterns. Its improving loss over epochs signifies learning potential, though longer training could benefit. Execution times vary, possibly requiring optimizations for consistency. In conclusion, this model exhibits promise for sequential data prediction. To enhance it further, validating on separate datasets, extending training, and refining hyperparameters could bolster performance.

**5.12.7 Stacked LSTM**

Stacked LSTMs, an advanced extension of the LSTM model, elevate forecasting accuracy by deepening the network's complexity. This architecture comprises multiple LSTM layers, enhancing the network's learning capacity by extracting intricate temporal representations from sequential data. In stock price prediction, Stacked LSTMs adeptly capture short-term trends and long-term dependencies, offering more precise forecasts. Yet, their increased complexity demands meticulous tuning to avoid overfitting, alongside heightened computational requirements for training.

In stacked LSTMs, the long-range dependencies are captured using multiple layers of LSTM stacked together. The output of hidden states of LSTM are given to the subsequent input layers of LSTM. This helps not only in capturing long-range dependencies but also in predicting by learning sophisticated patterns in the input data. They also provide leverage of avoiding underfitting through multiple layers but they come at the expense of higher computational expense compared to single LSTM layer models. They also offer the advantage of hyperparameter tuning which provide dropout rates and number of units crucial in achieving optimal parameters and prevent underfitting and overfitting.

The stacked LSTMs layers help in capturing relevant features as the data goes the input layers and are designed to maintain long term memory and also to learn nonlinear patterns and predict the time series data with more accuracy. All these comes with the computational expense during the training due to multiple layers involved.
Despite these challenges, with careful regularization and hyperparameter optimization, Stacked LSTMs excel in tasks reliant on understanding intricate temporal dynamics, delivering superior predictive performance.

Figure 27 Simple LSTM Architecture

- **Model Structure**: The model shown in Figure 27 is built like a tower with nine layers that help it understand information better. Each layer is like a different floor in a building.
- **Complexity**: It has a lot of settings, around 314,000 of them! That's like having a ton of knobs to adjust.
- **Training Process**: During practice, it got better over 100 rounds. At first, it made big mistakes, but then it improved steadily, like getting better at a game by practicing.
- **Performance**: When it was tested, it made mistakes about 8.70 times on average. That might seem like a lot, but it's pretty good considering what it's trying to do.
- **Insights**: The tower of layers helps the model learn in a special way, but too many layers can sometimes make things confusing. So, it takes breaks to avoid making too many mistakes.
- **Future Improvements**: To get even better, it might need to practice stopping earlier or changing its settings. It could also compare itself to other similar models to see if it's doing well or if it can do even better.
5.12.8 Bi-LSTM

Bi-directional Long Short-Term Memory networks, known as Bi-LSTMs, are a tweak on regular LSTMs. They learn from the input sequence in two ways: one looks at it forward, like reading a book, while the other goes backward, like reading the same book from the last page to the first. This way, a Bi-LSTM learns bidirectionally from both past and future data points around a current data point, getting a better grip on the data.

In stock price prediction, Bi-LSTMs shine because they catch tricky patterns and links in the time-based data that one-way models might miss. This two-sided learning helps the model understand how things change over time, making predictions more trustworthy. For example, if there are hints in the data that tell us where stock prices might go, a Bi-LSTM can spot and use these clues effectively.

Yet, the added complexity of Bi-LSTMs shown in Figure 28 comes with a downside—they can easily get too fixated on the details, especially when handling smaller sets of data. To avoid this, tweaking their settings just right, using tricks to control this obsession, and checking their performance carefully become super important. These steps help prevent them from focusing...
too much on the specifics and instead build a strong, trustworthy model for predicting stock prices. Although it's a bit tricky, when Bi-LSTMs are set up and trained well, they become really good at guessing stock prices. They're like detectives—spotting all the complex and twisty connections hiding in the financial data.

![Image of Bi-LSTM Architecture](image.png)

**Figure 29 Bi-LSTM Architecture**

### Model Setup

This model, called a Bi-LSTM, shown in Figure 29 looks at sequence data in both directions. This helps it understand what happened before and what might happen next.

### Training

It practiced for 50 rounds. At first, it made bigger mistakes, but over time, it got much better. By the end, it was making very small errors, showing it learned well.
Performance

Model performed on the test data pretty good and the following conclusions can be drawn:

➢ The mistakes it made, on average, were about 6.38 (MSE). Lower numbers mean it was closer to getting things right.
➢ It was usually around 1.93 units off with its guesses (MAE). This shows how wrong it was on average.
➢ The data points were concentrated around 2.53 units from the expected values (RMSE).

Comparison with Another Model

Compared to a different model called Stacked LSTM, this Bi-LSTM did better:

➢ It made fewer mistakes than the Stacked LSTM in all three measures (MSE, MAE, and RMSE).

Overall Impressions

This Bi-LSTM model looks really promising:

➢ It got better with practice and made fewer mistakes.
➢ Its ability to look in both directions helped it notice patterns that a one-way model might miss.
➢ Considering time from both sides seems to be a smart move for these kinds of predictions.

In conclusion, this Bi-LSTM model outperformed the Stacked LSTM on this dataset. It suggests using Bi-LSTMs for similar tasks or fine-tuning this model for even better results.

5.12.9 Gated Recurrent Unit (GRU)

Gated Recurrent Units (GRUs) are a type of RNNs made to understand sequences in the input data better. They fix a problem in regular RNNs where some information gets lost, making it hard to learn from far-away connections. GRUs are good at catching these long patterns, making them useful for things like guessing future events in a series of data.
GRUs shown in Figure 30 are like simpler versions of another kind called LSTMs but still great at learning long-term stuff. In a GRU, there are two gates—think of them as controllers. One decides what old stuff to forget, while the other decides what old stuff is worth keeping for the future. These gates help GRUs choose what to remember and what to ignore, making them pretty smart at understanding sequences of data.

In predicting stock prices, GRUs shine. They're really good at understanding how one price links to another, picking up on the trends and patterns that help predict where prices might go. Stock prices are pretty chaotic, but GRUs strike a balance—they're fast to learn and still great at capturing the important stuff in the data.

But, like any smart tool, GRUs need to be set up just right and checked carefully to avoid getting too caught up in the details, especially in the unpredictable world of finance. When they’re tuned and trained well, though, they become a powerful tool for folks wanting more accurate predictions in the stock market.

GRUs like LSTMs are also advantageous in coping and holding the long range dependencies in the time series data compared to RNNs by overcoming the gradient reduction problem through update gates and reset gates. These are less computationally expensive compared to LSTM due to lesser complexities due to architecture of GRU.
Model Setup

The GRU model is shown in Figure 31 has four GRU layers, each with a dropout layer to avoid getting too fixated on details. It ends with a single-neuron dense layer, fitting for this kind of problem. In total, there are 87,809 settings to tweak.

```
Model: "sequential_104"
Layer (type)        Output Shape    Param #
gru (GRU)           (None, 60, 64)   12864
dropout_477 (Dropout) (None, 60, 64)  0
gru_1 (GRU)         (None, 60, 64)   24960
dropout_478 (Dropout) (None, 60, 64)  0
gru_2 (GRU)         (None, 60, 64)   24960
dropout_479 (Dropout) (None, 60, 64)  0
gru_3 (GRU)         (None, 64)      24960
dropout_480 (Dropout) (None, 64)     0
dense_102 (Dense)   (None, 1)       65
Total params: 87,809
Trainable params: 87,809
Non-trainable params: 0
```

Figure 31 GRU Model Summary

Training Performance

It practiced for 50 rounds. At first, it was making big mistakes generating big errors, but it got much better as the epochs increased. By the end, it was making really small errors, showing it learned well.

Evaluation Metrics

- On average, the model's guesses were around 2.28 units off from the actual values (MAE).
- The squared version of these errors was about 8.94 (MSE).
- The data points were around 2.99 units away from where they were expected to be (RMSE).
Insights

➢ The GRU model is great at dealing with sequences and time-based data. The way it's built, with multiple GRU layers, seems to be working well, steadily getting better at learning.

➢ It's doing pretty well, but when we compare its errors to another model called BiLSTM, the Bi-LSTM did slightly better on this dataset.

➢ Using dropout layers after each GRU helped the model avoid getting too obsessed with the details, making it better at predicting new data.

➢ If you're looking for a simpler model that's still pretty good, the GRU might be a good choice. But if you want the most accurate predictions, the Bi-LSTM did a bit better here.

Conclusion

Choosing between GRU and Bi-LSTM depends on what you need for your project and how much computing power you have. Testing both on new data can help understand which one works better in the long run.

5.12.10 Hybrid Algorithm (RNN, LSTM, Bi-LSTM & GRU)

A special combined algorithm brings together RNN, LSTM, Bi-LSTM, and GRU. Each model brings something great: RNN tracks time really well, LSTM learns long stuff better, Bi-LSTM looks in both directions, and GRU is fast and still learns the long things. All these combined make a super smart model for guessing stock prices accurately.

But, making this special mix needs a lot of planning, tough practice, and careful checking to avoid getting too into the details. It's like finding the perfect balance between being really smart and making good guesses in the ever-changing world of financial guessing games. The order of layers is made according to input and output layers of each architecture in trial and error method.
Model Architecture

The given model shown in Figure 32 is a sequential model with a combination of various types of recurrent layers:

<table>
<thead>
<tr>
<th>Layer</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>simple_rnn_117 (SimpleRNN)</td>
<td>(None, 60, 64)</td>
<td>4224</td>
</tr>
<tr>
<td>dropout_481 (Dropout)</td>
<td>(None, 60, 64)</td>
<td>0</td>
</tr>
<tr>
<td>gru_4 (GRU)</td>
<td>(None, 60, 64)</td>
<td>24960</td>
</tr>
<tr>
<td>dropout_482 (Dropout)</td>
<td>(None, 60, 64)</td>
<td>0</td>
</tr>
<tr>
<td>lstm_360 (LSTM)</td>
<td>(None, 60, 64)</td>
<td>33024</td>
</tr>
<tr>
<td>dropout_483 (Dropout)</td>
<td>(None, 60, 64)</td>
<td>0</td>
</tr>
<tr>
<td>bidirectional_12 (Bidirectional) (None, 60, 128)</td>
<td>66048</td>
<td></td>
</tr>
<tr>
<td>dropout_484 (Dropout)</td>
<td>(None, 60, 128)</td>
<td>0</td>
</tr>
<tr>
<td>gru_5 (GRU)</td>
<td>(None, 60, 64)</td>
<td>37248</td>
</tr>
<tr>
<td>dropout_485 (Dropout)</td>
<td>(None, 60, 64)</td>
<td>0</td>
</tr>
<tr>
<td>gru_6 (GRU)</td>
<td>(None, 64)</td>
<td>24960</td>
</tr>
</tbody>
</table>

Figure 32 Summary of Hybrid Algorithm

- Simple RNN layer
- GRU layers
- LSTM layer
- Bidirectional layer wrapping another LSTM layer with 64 units, effectively doubling the output to 128 units due to its bidirectional nature
- A dense layer at the end, which suggests this is a regression model.
- The model includes dropout layers between each recurrent layer to prevent overfitting.

190,529 Trainable parameters.

Training

Trained for 100 epochs. The loss, which seems to be Mean Squared Error (MSE), starts at 0.0241 in the first epoch and decreases consistently over time, reaching 0.0011 by the 50th epoch indicating model is improving over training epochs.
Performance Metrics

The performance of model on presumably a test set is provided below:

- Mean Absolute Error (MAE): 1.7614
- Mean Squared Error (MSE): 5.3414
- Root Mean Squared Error (RMSE): 2.3112

Insights

- Model Complexity: The given model is quite complex with a mix of Simple RNN, GRU, LSTM, and Bidirectional LSTM layers. Such a combination might be designed to capture different temporal dynamics in the data.
- Training Convergence: The model's loss consistently decreases over the epochs, suggesting that it is converging and learning patterns from the training data. However, the need to be cautious about potential overfitting arises given the model's complexity. Regularization techniques, like dropout layers, are rightly used to mitigate this.
- Performance: The model's MAE of 1.7614 indicating an average error by a value of approximately 1.7614 (in closing price). RMSE being higher than the MAE indicates that there are some predictions with larger errors, as RMSE penalizes larger errors more heavily.
- Comparison with Previous GRU Model: Compared to the previously discussed GRU model (with an MAE of 2.2776 and RMSE of 2.9908), this mixed architecture model has better performance. It has both a lower MAE and RMSE, suggesting it is making more accurate predictions on average and has fewer large errors.
- Recommendation: Given that the complex model with mixed architectures performs better than the simpler GRU model, it might be beneficial for the given problem. However, it's essential to ensure the model's ability by testing it on diverse unseen or new data. Also, due to its complexity, it might be more computationally extensive.

In summary, the hybrid model demonstrates promising results and outperforms the simpler GRU model. However, careful consideration regarding computational efficiency and generalization capability is necessary before deploying such model in production environment.
5.12.11 Transformer Model

The Transformer model, with its innovative attention mechanism, has significantly impacted forecasting, especially in the financial domain. Its ability to process sequential data and identify relevant patterns across vast timelines sets it apart from traditional forecasting methods. This model excels in managing the complexities of financial time series data, capturing long-term dependencies and subtle market signals that earlier models might miss. The Transformer's scalability and adaptability to large datasets allow for a more nuanced analysis of financial trends, leading to potentially more accurate and insightful forecasts. Its core mechanism, focusing on the importance of different data points within a sequence, enables it to adjust dynamically to the ever-changing landscape of financial markets. Transformer has been emerging as excellent model for financial analysts and researchers, offering a sophisticated approach to predict market movements with greater precision.

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>input_1 (InputLayer)</td>
<td>([None, 60, 1])</td>
<td>0</td>
</tr>
<tr>
<td>tf.<strong>operators</strong>.add (TFOp Lambda)</td>
<td>(None, 60, 64)</td>
<td>0</td>
</tr>
<tr>
<td>multi_head_attention (Multi iHeadAttention)</td>
<td>(None, 60, 64)</td>
<td>16640</td>
</tr>
<tr>
<td>dropout_6 (Dropout)</td>
<td>(None, 60, 64)</td>
<td>0</td>
</tr>
<tr>
<td>layer_normalization (Layer Normalization)</td>
<td>(None, 60, 64)</td>
<td>128</td>
</tr>
<tr>
<td>sequential_2 (Sequential)</td>
<td>(None, 60, 64)</td>
<td>16576</td>
</tr>
<tr>
<td>dropout_7 (Dropout)</td>
<td>(None, 60, 64)</td>
<td>0</td>
</tr>
<tr>
<td>layer_normalization_1 (Layer Normalization)</td>
<td>(None, 60, 64)</td>
<td>128</td>
</tr>
</tbody>
</table>

Figure 33 Parameters of Transformer Model

The above information shown in Figure 33 outlines the architecture and performance of a Transformer model trained on a time series forecasting task, presumably related to stock market
prediction or a similar financial domain. The model architecture is complex, featuring multiple layers that leverage the Transformer's core mechanisms, such as multi-head attention, dropout for regularization, and layer normalization for stabilizing the learning process. Notably, the model includes sequential layers, possibly indicating the use of feedforward neural networks within the Transformer blocks to process the data after attention mechanisms have been applied. The training process spans 20 epochs, showing a consistent decrease in loss, indicating effective learning and model improvement over time. The initial loss starts relatively high but undergoes a significant drop by the second epoch, suggesting that the model quickly adapts to the patterns in the data. This rapid decrease in loss early in the training process shows model's ability to capture relevant features from the time series data.

The final performance metrics provided are MAE, MSE, and RMSE, with values of 0.3148, 0.1521, and 0.3900, respectively. These metrics indicate the model's forecasting accuracy, with lesser error values representing improved performance. The relatively low values of these metrics suggest that the model has good level of accuracy in predicting future values of time series. Specifically, the RMSE, which gives a sense of the average prediction error magnitude, indicating predicted values are relatively closer to the true data points on average, marking its utility for practical financial forecasting tasks. The following plot shown in Figure 34 contains the actual vs predicted values of the Transformer model.

Figure 34 Forecasted Plot for the Transformer Model
5.12.12 GPT-2 Model

The GPT-2 model, a creation by OpenAI, represents a leap forward in natural language processing and has intriguing applications in stock price forecasting. Unlike traditional models that primarily analyze numerical data, GPT-2's prowess lies in its ability to comprehend and generate text-based information. This capability makes it uniquely suited for analyzing the huge textual data in financial markets, such as earnings reports, and social media commentary, which significantly influence stock prices.

GPT-2's effectiveness in stock price forecasting stems from its deep learning architecture, which utilizes transformers to process data. This allows it to capture the nuanced sentiment and contextual information embedded in text, offering insights into market trends and investor sentiments that are not readily apparent in numerical data alone. By training GPT-2 on a diverse corpus of financial texts, it can gauge the dependency of stock prices on news events, providing it to predict changes in stock prices based on emerging news stories or shifts in public sentiment.

Incorporating GPT-2 into stock price forecasting models presents an innovative approach that complements traditional quantitative analysis. This analysis can improve the accuracy of forecasts, providing investors and analysts with a more bird view of the dynamics in stock prices at play.

<table>
<thead>
<tr>
<th>Model: &quot;tfgpt2_model_3&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Layer (type)</td>
</tr>
<tr>
<td>Transformer (TFGPT2MainLayer multiple)</td>
</tr>
</tbody>
</table>

Total params: 28445184 (108.51 MB)
Trainable params: 28445184 (108.51 MB)
Non-trainable params: 0 (0.00 Byte)

Figure 35 Summary of GPT-2 Model

The provided information details in Figure 35 the training process and performance evaluation of a GPT-2 model, specifically configured for a forecasting task, likely within the financial domain given the context. This model, identified as "tfgpt2_model_3," consists a substantial parameter count of 28,445,184, highlighting its complexity and potential for capturing nuanced patterns in data.
Over the course of 20 epochs, the model undergoes training with a notable, consistent decrease in loss from 1.2059 to 0.0903. This steady decline in loss signifies effective learning and adaptation to train data, suggesting the model's increasing proficiency in forecasting based on input it was provided. The rapid improvements in the initial epochs followed by smaller, incremental gains in later epochs are indicative of the model to detect most of the trends early on, with subsequent epochs refining its predictions.

Upon evaluation, the model achieves a MAE of 0.2035, MSE of 0.0690, and RMSE of 0.2627. These metrics shows model ability in prediction:

- A MAE of 0.2035 suggests that, on average, the model's predictions deviate from the actual values by a relatively small margin.
- The relatively low MSE of 0.0690 indicates a good fit to the data, with fewer large deviations.
- RMSE offers a similarly scaled measure of errors as the original data, making it easier to interpret. The RMSE of 0.2627 further confirms the model's efficacy, indicating that model ability shown in below Figure 36 are, on average, within a 0.2627 unit range of the true values.

![Figure 36 Forecasted Plot for GPT-2 Model](image)
These performance metrics, particularly in conjunction with the detailed training loss reduction, provide insightful evidence of the GPT-2 model's capability shown in Figure 36 to accurately forecast future values in its application domain. Its performance underscores the potential of advanced neural network architectures like GPT-2, offering valuable tools for tasks such as stock price forecasting where precision is crucial.

5.12.13 Time Series Transformer (TST) model

The Time Series Transformer (TST) model is transformer, specifically tailored for time series data analysis. TST model focuses on capturing the temporal dynamics and dependencies inherent in time series data. This specialization makes it particularly adept at understanding the sequential nature of data points over time, a crucial aspect of forecasting tasks such as finance, weather prediction.

The TST model incorporates key elements of the Transformer, multi head attention mechanism which helps in capturing the data points at various time points. This feature helps the TST to focus on the most relevant information for making predictions, improving its accuracy and efficiency. Additionally, the TST model can allow for the incorporation of multiple related data streams into the forecasting process.

Key strengths of the TST model is its capability in dealing with long time series of data without losing the context, a common challenge in time series analysis. This capability, combined with its flexibility in modeling complex temporal relationships, positions the TST as a excellent tool for time series forecasting, with enriched performance and deeper insights.

The TST Transformer is similar to other Transformer but is more specifically designed for time-series data modelling compared to other Transformer which are mostly used for text generation tasks and in Large Language Models.

The information outlines the architecture shown in Figure 37, training progress, and performance metrics of a Time Series Transformer (TST) model specifically designed for time series forecasting.
This model, "functional_35," demonstrates a complex architecture optimized for handling sequential data, as evidenced by its use of multi-head attention layers, dropout for regularization, layer normalization for stability, and sequential layers presumably for processing sequences in a deep learning context.

The training process reveals an initial sharp decrease in loss from 8.4559 to 0.1417 within the first two epochs, indicating a rapid learning phase, showing model quickly adapts to the
underlying trends in the dataset. Subsequent epochs show a more gradual reduction in loss, stabilizing around 0.0668 by the 20th epoch. This progression suggests that the model efficiently detects the patterns of the time series data, with diminishing returns on learning as it approaches an optimal state.

Performance metrics post-training, including MAE, MSE, and RMSE, shows model accuracy. An MAE of 0.1895 and an RMSE of 0.2372, both relatively low values, indicate a high degree of accuracy in the model's predictions compared to actual outcomes. The MSE of 0.0563 further corroborates this, suggesting that the model predicted values are generally close to the true values, with errors being minor on average.

Overall, the TST model's architecture and its performance metrics highlight its effectiveness in time series forecasting with prediction vs actual plot shown in Figure 38. The rapid initial learning and subsequent fine-tuning of the model across epochs demonstrate its capability to discern and adapt to complex temporal patterns in data. The low error metrics post-training reflect its precision in stock price prediction, making it a potentially excellent tool for applications requiring high accuracy based on historical sequential data.
5.12.14 Transformer-Interpretation with Transfer Learning

Although the evaluation metrics for the transformer-based models are more accurate, the figures plotted between predicted and actual values for all three transformers shows that the transformer models are finding it difficult to interpret and follow the intricacies the time series data. This can be understood since the transformer are designed for more different use-cases but with hyper-tuning, it is still possible to obtain better results.

Hybrid Model including a transformer layer:

In this section, consider the hybrid model using LSTM, BI-LSTM, GRU, but a transformer layer in between is introduced, and trained with obtained parameters and interpret its results and performance.

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
<th>Connected to</th>
</tr>
</thead>
<tbody>
<tr>
<td>input_2 (InputLayer)</td>
<td>[(None, 300, 2)]</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>simple_rnn_1 (SimpleRNN)</td>
<td>(None, 300, 64)</td>
<td>4288</td>
<td>input_2[0][0]</td>
</tr>
<tr>
<td>dropout_6 (Dropout)</td>
<td>(None, 300, 64)</td>
<td>0</td>
<td>simple_rnn_1[0][0]</td>
</tr>
<tr>
<td>gru_3 (GRU)</td>
<td>(None, 300, 64)</td>
<td>24960</td>
<td>dropout_6[0][0]</td>
</tr>
<tr>
<td>dropout_7 (Dropout)</td>
<td>(None, 300, 64)</td>
<td>0</td>
<td>gru_3[0][0]</td>
</tr>
<tr>
<td>lstm_1 (LSTM)</td>
<td>(None, 300, 64)</td>
<td>33024</td>
<td>dropout_7[0][0]</td>
</tr>
<tr>
<td>dropout_8 (Dropout)</td>
<td>(None, 300, 64)</td>
<td>0</td>
<td>lstm_1[0][0]</td>
</tr>
<tr>
<td>multi_head_attention_1 (MultiHead)</td>
<td>(None, 300, 64)</td>
<td>16640</td>
<td>dropout_8[0][0]</td>
</tr>
<tr>
<td>dropout_9 (Dropout)</td>
<td>(None, 300, 64)</td>
<td>0</td>
<td>multi_head_attention_1[0][0]</td>
</tr>
<tr>
<td>gru_4 (GRU)</td>
<td>(None, 300, 64)</td>
<td>24960</td>
<td>dropout_9[0][0]</td>
</tr>
<tr>
<td>dropout_10 (Dropout)</td>
<td>(None, 300, 64)</td>
<td>0</td>
<td>gru_4[0][0]</td>
</tr>
<tr>
<td>gru_5 (GRU)</td>
<td>(None, 64)</td>
<td>24960</td>
<td>dropout_10[0][0]</td>
</tr>
<tr>
<td>dropout_11 (Dropout)</td>
<td>(None, 64)</td>
<td>0</td>
<td>gru_5[0][0]</td>
</tr>
<tr>
<td>dense_1 (Dense)</td>
<td>(None, 1)</td>
<td>65</td>
<td>dropout_11[0][0]</td>
</tr>
</tbody>
</table>

Figure 39 Hybrid Model Summary with Transformer Interpretation

This model shown in Figure 39 represents a sophisticated sequence processing architecture, leveraging a stack of recurrent layers including Simple RNN, GRU, LSTM, and a Multi-Head Attention mechanism to predict a continuous outcome. The diversity of recurrent units—each with its unique way of handling memory and temporal information—combined with dropout layers for regularization, suggests the model is designed to capture complex time-series patterns while mitigating overfitting.
The architecture begins with a Simple RNN layer, known for its simplicity and efficiency but also its limitations in capturing long-term dependencies. The GRU layers that follow are an evolution, designed to better retain information over longer sequences. The LSTM layer further enhances the model's ability to remember information for long durations. The inclusion of a Multi Head Attention layer is particularly noteworthy, enabling the model to focus different time series points at a time parallel making predictions.

The model training process shows a consistent decrease in loss over epochs, indicating effective learning and improvement in prediction accuracy over time. The final reported metrics—R2 Score of 0.9483, MAE of 0.0373, MSE of 0.0022, and RMSE of 0.0474—highlight its high predictive accuracy and low error rates, which are indicative of a well-tuned model capable of capturing the underlying patterns in the data with a high degree of precision.

The progression in loss reduction and the strong performance metrics suggest the model's capabilities and training regimen are nicely suited for the task at hand, potentially offering valuable insights and predictions for time-series data analysis as shown in Figure 40.

![Actual vs Predicted Values](image)

Figure 40 Forecasted Plot for Hybrid Model with Transformer

The progression in loss reduction and the strong performance metrics suggest the model's capabilities and training regimen are nicely suited for the task at hand, potentially offering valuable insights and predictions for time-series data analysis as shown in Figure 40.
Transfer Learning with hybrid model using LSTM and BI-LSTM:

In this approach, a new dataset [8] is taken and the hybrid models are trained on the new which is different but relatable dataset of stock prices. Here, a model is built by training on GOOGLE stock data and then the model is saved and retrained with the target AAPL dataset on its final layer. The evaluation metrics performed are shown below.

In this refined approach, the focus shifts towards leveraging a hybrid model uniquely trained across two distinct yet related datasets of stock data. Initially, the model is meticulously trained on Google stocks, establishing a robust foundational understanding of stock market dynamics and trends. Following this initial phase, the model undergoes a strategic retraining process, specifically targeting its final layer with data from Apple (AAPL) stocks. This innovative method of layer-specific retraining allows for the transfer and adaptation of learned patterns from Google's stock to precisely predict Apple's stock movements, harnessing the similarities in market behavior between the two tech giants while fine-tuning the model to capture the unique characteristics of Apple's stock.

Apple,
Mean squared error : 0.002625489704751763
Root Mean squared error : 0.051239532635961496
Mean Absolute error : 0.03630144816506908
r2 score : 0.9754915905579639

Figure 41 Performance Evaluation for Apple Stock after Transformer Interpretation

The performance of the model, after being retrained with Apple's stock data, is quantified through several key evaluation metrics, showcasing its predictive prowess. MSE stands at 0.002625489704751763, indicating a relatively low average differences between the predicted and true values. RMSE is calculated to be 0.051239532635961496. This suggests that, on average, from Figure 41, the model's predictions deviate from the actual stock prices by a margin of approximately 5.12%. MAE a direct metric of the average magnitude of errors in predictions, is observed to be 0.03630144816506908, underscoring the models ability in capturing the stock price movements without being significantly affected by large errors.

Remarkably, the model achieves an r2 score of 0.9754915905579639 when applied to Apple's stock data. This score, approaching the upper limit of 1, signifies an exceptionally high level of variance in the stock prices that the model successfully captures, highlighting its effectiveness and precision in predicting Apple's stock price movements based on the learned
patterns from Google's stock data. This hybrid training approach not only exemplifies the adaptability and efficiency of the model in handling stock price predictions as shown in below Figure 42 across different companies but also sets a precedent for the potential of transfer learning and fine-tuning in financial market analysis.

Figure 42 Forecasted Plot for Apple Stock after Transformer Interpretation

Transfer Learning with hybrid model using LSTM and BI-LSTM and Transformer:

In this, the same above approach is used but the model is included a transformer layer this time to test for the ability of transformer to evaluate its accuracy and the results are shown below.

Adopting the same hybrid modeling approach, the focus is now shifted towards Amazon stock data, where the model, initially trained on Google stock information, is further fine-tuned by retraining its final layer with Amazon-specific data. This strategy leverages the foundational knowledge acquired from Google's stock patterns to enhance the model's predictive capabilities for Amazon's stock, effectively utilizing transfer learning to adapt to the nuances of Amazon's market behavior while retaining the core analytical insights gained from Google.

![Closing price comparison for Apple dataset](image)

Figure 43 Performance Evaluation Parameter for Amazon Stock after Transformer Interpretation

Upon evaluation, the model exhibits exceptional performance metrics with Amazon's stock data, reflecting its refined predictive accuracy. MSE is recorded at 0.0011235933497587248, suggesting a minimal the errors between the predicted and actual stock prices. RMSE, offering a more straightforward interpretation by measuring the average prediction error, is found to be 0.0335200439999521.
This shows the model's prediction shown in Figure 43 typically deviate from the true stock prices by an average of approximately 3.35%, illustrating the model's high accuracy level. MAE, is reported to be 0.02591407944610717. This metric highlights the model's effectiveness in closely following the trends of Amazon's stock prices with a relatively low error magnitude.

Furthermore, the model achieves an r2 score of 0.982339518209957 for Amazon's stock data. This score, nearly reaching the perfect fit indicator of 1, signifies that the model is capable of explaining a very high percentage of the fluctuations in the stock prices, underscoring its exceptional predictive performance.

This approach, utilizing a hybrid model trained across datasets from two different corporations (initially Google, then Amazon), not only demonstrates the model's versatility and robustness in stock price prediction across varying market conditions but also showcases the power of specialized retraining in applicability of predictive models in the financial sector as shown in Figure 44. The model shows better evaluation metric results compared to the model with LSTM, Bi-LSTM layers but finds it difficult to interpret the complexities in the prediction. This is due to limited to parameter tuning we have performed but can perform much better if suitable parameters are selected for the transformer layer.
5.13 Model Comparison

Table 1 below constitutes of the evaluation metrics Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE) for the models discussed so far. The table shows that the deep learning models are performing better when it comes to time series data compared to decision tree and linear regression. Further modification of deep learning models like stacked structures and combining different architectures of deep learning gave better results. The Transformer models introduced gave satisfactory results in the evaluation metrics but the actual prediction values tend to be less accurate which can be improved with selection of suitable parameters. Transfer Learning using hybrid model with transformer layer has given better evaluation metrics compared to all the other models followed by the Transfer Learning using hybrid model with transformer layer. However, the prediction plot for the former shows it is failing to interpret the complexities in the stock price data compared to the latter. This can be improved by selecting better parameters for the transformer layer which can be obtained with hyper parameter tuning.

<table>
<thead>
<tr>
<th>Model</th>
<th>Mean Absolute Error (MAE)</th>
<th>Mean Squared Error (MSE)</th>
<th>Root Mean Squared Error (RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARIMA</td>
<td>1.521</td>
<td>2.4357</td>
<td>1.5607</td>
</tr>
<tr>
<td>SARIMA</td>
<td>1.1562</td>
<td>6.0955</td>
<td>2.4689</td>
</tr>
<tr>
<td>Linear Regression</td>
<td>0.3389</td>
<td>0.4718</td>
<td>0.6869</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>24.0999</td>
<td>731.7433</td>
<td>27.0508</td>
</tr>
<tr>
<td>RNN</td>
<td>1.5931</td>
<td>4.6041</td>
<td>2.1457</td>
</tr>
<tr>
<td>LSTM</td>
<td>1.7239</td>
<td>5.1182</td>
<td>2.2623</td>
</tr>
<tr>
<td>Stacked LSTM</td>
<td>2.2691</td>
<td>8.701</td>
<td>2.9497</td>
</tr>
<tr>
<td>Bi-LSTM</td>
<td>1.9339</td>
<td>6.3812</td>
<td>2.5261</td>
</tr>
<tr>
<td>GRU</td>
<td>2.2776</td>
<td>8.9446</td>
<td>2.9908</td>
</tr>
<tr>
<td>Hybrid</td>
<td>1.7614</td>
<td>5.3414</td>
<td>2.3112</td>
</tr>
<tr>
<td>Transformer</td>
<td>0.314780</td>
<td>0.152113</td>
<td>0.390017</td>
</tr>
<tr>
<td>GPT-2</td>
<td>0.203485</td>
<td>0.069037</td>
<td>0.262750</td>
</tr>
<tr>
<td>TST</td>
<td>0.189463</td>
<td>0.056280</td>
<td>0.237234</td>
</tr>
<tr>
<td>Hybrid with Transformer</td>
<td>0.037271</td>
<td>0.002248</td>
<td>0.047415</td>
</tr>
<tr>
<td>TL using Hybrid</td>
<td>0.036301</td>
<td>0.002625</td>
<td>0.051239</td>
</tr>
<tr>
<td>TL using hybrid with Transformer</td>
<td>0.025914</td>
<td>0.001123</td>
<td>0.033520</td>
</tr>
</tbody>
</table>

Table 1 Model Comparison
The analysis of forecasting models shown in the Figure 45 across various methodologies showcases a wide spectrum of performance metrics. Traditional statistical methods, such as ARIMA and SARIMA, deliver moderate accuracy levels. Notably, SARIMA exhibits a higher Mean Squared Error (MSE) than ARIMA, potentially pointing towards issues of overfitting or an increased sensitivity to data outliers.

Among statistical models, Linear Regression distinguishes itself with relatively lower error rates, indicating a strong fit for the dataset in question. In contrast, tree-based models like Decision Trees and Random Forests present considerably higher error metrics, suggesting these models may be overfitting or not well-suited to the dataset's complexity. Neural network approaches, including various configurations of RNNs, LSTM, and GRUs, yield mixed outcomes.

While the basic LSTM model shows improvement over the simpler RNN, it is outperformed by more sophisticated setups. Specifically, GRU, Bi-LSTM, and Stacked LSTM models register higher error values, implying potential challenges in capturing the dataset's temporal dynamics fully.
The Hybrid model improves upon single-method approaches but doesn't achieve optimal performance. However, the Transformer model marks a substantial advancement, showcasing its superior handling of sequential data. Moreover, GPT-2 and the Temporal Fusion Transformer (TST) models demonstrate exceptionally low error rates, underlining the effectiveness of transformer-based architectures in time series forecasting. The TST model, in particular, achieves the least MAE and RMSE, accentuating its specialized design for temporal data analysis.

When incorporating these methodologies with a hybrid approach, following nuanced performances are observed:

- The basic Hybrid model shows promising results but with room for improvement.
- A more sophisticated application, combining the Hybrid model with Transformer techniques, significantly reduces error metrics, highlighting the synergy between these models.
- Transfer Learning (TL) techniques applied to the Hybrid model for specific datasets (like stock data) further refine its predictive accuracy, as evidenced by even lower error rates.
- The application of TL to a Hybrid model enhanced with Transformer architecture yields the best performance metrics among the reviewed models, emphasizing the power of combining advanced machine learning techniques for time series analytics.

In summary, the evolution from basic statistical and tree-based models to advanced neural network and transformer-based architectures, culminating in the strategic application of Hybrid models with Transfer Learning, represents a significant progression in forecasting accuracy and model sophistication. This trajectory suggests that the integration of advanced methodologies, particularly transformer-based models fine-tuned for specific tasks, represents the cutting edge in time series forecasting.

5.14. Results and Discussion

The comparative study of forecasting models reveals a distinct superiority of transformer-based architectures, namely the TST and GPT-2 models, in handling time series forecasting tasks. Their advanced design enables them to adeptly capture temporal dependencies, leading to their standout performance characterized by low error rates across all evaluated metrics. These
models excel because of their capacity to identify and learn long-range patterns and complexities within time series data, a critical capability for precise forecasting. Conversely, traditional models such as ARIMA and SARIMA, despite their utility, lag behind, mainly because of their difficulties with the non-linear and intricate data structures typical of financial time series. Decision trees and random forests significantly underperform, likely due to overfitting issues in complex data landscapes. Among neural network-based approaches, a spectrum of outcomes is observed, with simpler RNN models being eclipsed by LSTMs and GRUs whose results are further bettered by the hybrid model. This analysis underscores the pivotal role of deep learning. Bi-LSTMs have bettered out stacked LSTM might be due to bidirectional learning capability. Transformers, has shown significant accuracy and robustness in financial time series forecasting, marking a clear preference for these advanced methodologies over traditional and simpler neural network models. The hybrid models with transformer layer turned to be having better evaluation metrics when compared to hybrid model with deep learning layers but they lacked in interpreting the complexities in the time series data as compared to latter which can be improved with further parameter tuning. The similar trend of results is shown by the transfer learning models between deep learning hybrid model and the transformer hybrid model which can be improved with further hyper parameter tuning.
Chapter 6 - Conclusion and Future Work

6.1 Conclusion

The comparative analysis of various forecasting models highlights a significant evolution in predictive capabilities, moving from traditional statistical methods to advanced neural networks and transformer-based architectures. Traditional models like ARIMA and SARIMA, while providing a baseline of accuracy, show limitations in handling complex datasets, as indicated by their moderate to high error metrics. The surprisingly robust performance of Linear Regression underscores its effectiveness in certain scenarios, yet it is overshadowed by the high error rates of tree-based models like Decision Trees and Random Forests, which suggest overfitting or a misalignment with the dataset's complexity.

The exploration into neural networks reveals a spectrum of outcomes, with LSTM showing promise over simpler RNNs, yet both are surpassed by the nuanced capabilities of GRU, Bi-LSTM, and Stacked LSTM models. However, these models still grapple with fully capturing the temporal dynamics of the datasets, as reflected in their higher error rates.

A pivotal turn is observed from transformer-based models, particularly the Transformer, GPT-2, and TST models, which exhibit exceptionally low error metrics but less effectiveness when compared to other models in predicting complex time series data which can be improved with better hyper parameter tuning. This marks a breakthrough in the field, highlighting the advanced potential of transformers in time series forecasting. The TST model, tailored for temporal data, stands out for its lowest error rates, showcasing the specialized advantages of this approach.

Integrating these insights with hybrid and transfer learning techniques further refines forecasting accuracy. The nuanced application of a Hybrid model with Transformer enhancements, followed by strategic transfer learning applications, demonstrates a significant leap forward, achieving unprecedented precision in predictions. This progression not only emphasizes the efficacy of combining multiple advanced techniques but also sets a new standard for time series forecasting, paving the way for reliable, accurate and adaptable predictive models across domains.
6.2 Future Work

The future scope of this project is set to embrace transfer learning, a strategic approach aimed at enhancing model performance by leveraging pre-trained models on new, but related tasks. This methodology promises to significantly reduce the need for extensive data collection and computational resources, making it an efficient strategy for tackling tasks with limited datasets. By applying transfer learning, there is an anticipation, not only of acceleration in the development cycle of predictive models but also a notable improvement in their accuracy and generalizability. This forward-looking approach paves the way for more sophisticated and adaptable models in predictive analytics, opening new avenues for research and application. The accuracy that the transfer learning performed using transformer layers combined with especially Bi-LSTM layers developed with little hyper-parameter tuning showing significant results, when provided proper hyper-parameter tuning can deliver promising and reliable results.
References


[34] Zimuel Enrico.Elasticsearch-PHP-Examples/Data/All_stocks_5yr.CSV at Main · Elastic/Elasticsearch-PHP-Examples, github.com/elastic/elasticsearch-php-examples/blob/main/data/all_stocks_5yr.csv.