A Synopsis of Project on

Sentiment-Augmented Stock Price Forecasting

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Computer Science and Engineering Data Science

by

Mustafa Shaikh (21107045)
Sumit Shahu (21107004)
Pravesh Yadav (21107057)
Ankit Purohit (21107020)

Under the Guidance of

Ms.Ujwala Pagare
Ms.Aavani Nair

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING (DATA SCIENCE)
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W)-400615
UNIVERSITY OF MUMBAI

Academic Year 2024-2025

Approval Sheet

This Project Synopsis Report entitled ***"Sentiment-Augmented Stock Price Forecasting "*** Submitted by ***"Mustafa Shaikh"(21107045), "Sumit Shahu"(21107004), " Pravesh Yadav"(21107057), "Ankit Purohit"(21107020)***is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Computer Science and Engineering Data Science*** from ***University of Mumbai***.

(Ms.Aavni Nair)
Co-Guide

(Ms.Ujwala Pagare)
Guide

Ms. Anagha Aher
HOD, Computer Science and Engineering Data Science

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Sentiment-Augmented Stock Price Forecasting "*** submitted by ***"Mustafa Shaikh" (21107045),"Sumit Shahu" (21107004), "Pravesh Yadav" (21107057),"Ankit Purohit" (21107020)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Computer Science and Engineering Data Science***,to the University of Mumbai,is a bonafide work carried out during academic year 2024-2025.

(Ms.Aavni Nair)
Co-Guide

(Ms.Ujwala pagare)
Guide

Ms. Anagha Aher
HOD, Data Science

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Internal Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

We have great pleasure in presenting the synopsis report on **Sentiment-Augmented Stock Price Forecasting .** We take this opportunity to express our sincere thanks towards our guide **Ms.Ujwala Pagare** & Co-Guide **Ms.Aavni Nair** for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Ms. Anagha Aher** Head of Department for his encouragement during the progress meeting and for providing guidelines to write this report.

We express our gratitude towards BE project co-ordinator **Ms.Ujwala Pagare**, for being encouraging throughout the course and for their guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Mustafa Shaikh
(21107045)

Sumit Shahu
(21107004)

Pravesh Yadav
(21107057)

Ankit Purohit
(21107020)

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)
Mustafa Shaikh(21107045)

(Signature)
Sumit Shahu(21107004))

(Signature)
Pravesh Yadav(21107057)

(Signature)

Ankit Purohit(21107020)

Date:

Abstract

The project aims to develop a real-time stock market prediction system using a hybrid deep learning approach combining RNN variants (LSTM, GRU). The system addresses the challenges of noisy, high-frequency data and aims to provide accurate and low-latency predictions for stock price movements. By integrating real-time data streams from sources like stock market APIs and news sentiment, the system leverages ensemble learning techniques to boost prediction accuracy. An ensemble model is built by combining LSTM, GRU, and traditional machine learning methods like XGBoost, further improving robustness and accuracy. The system is designed to handle the complex, sequential nature of stock market data, providing traders and investors with timely, actionable insights.

To further enhance prediction accuracy, the system incorporates real-time data streams from multiple sources, including stock market APIs and sentiment analysis of financial news. By analyzing both market data and external factors such as news events, the system gains a more holistic understanding of market dynamics, enabling it to predict price movements with greater precision.

The system is designed to operate with low latency, providing traders and investors with timely and actionable insights. The combination of LSTM, GRU, and XGBoost within a hybrid architecture ensures that the model can handle the complex, sequential nature of stock market data, while also adapting to fast-changing market conditions. Ultimately, this project aims to develop a highly accurate and real-time prediction tool that can support better decision-making for market participants, enhancing their ability to respond quickly to market fluctuations and capitalize on trading opportunities.

Contents

List of Figures

List of Tables

List of Abbreviations

IDS:	Intrusion Detection System
WSN:	Wireless Sensor Network
MANET:	Mobile Ad-Hoc Network
AODV:	Ad-Hoc On-demand Distance Vector Routing
DSR:	Dynamic Source Routing Protocol
NS2:	Network Simulator 2
ACK:	Acknowledgement
AGT:	Agent
RTR:	Router

Chapter 1

Introduction

The stock market is a key financial system that influences global economic stability and individual investments. Stock prices fluctuate rapidly due to a variety of factors including economic events, news, geopolitical changes, and investor sentiment. Accurately predicting stock prices is difficult due to the market's highly volatile and noisy nature. In real-time, the challenge becomes even harder because millions of trades and price changes happen every second. Traditional financial models struggle with real-time data, facing issues such as scalability, latency in predictions, and the dynamic nature of time-series data. Despite advances in financial technology, the complex and fast-changing stock market demands more sophisticated, adaptable tools for real-time prediction and analysis.

1.1 Motivation

The motivation for developing a stock price forecasting system is to improve decision-making by leveraging advanced models to analyze complex market data and trends in real time. This helps investors make informed decisions, optimize strategies, and gain a competitive edge.

- Modern stock markets are influenced by multiple volatile factors, making accurate predictions challenging.
- Unpredictable market behavior leads to higher risks, affecting decision-making and financial stability.
- Conventional models fail to account for dynamic, non-linear market behaviors, reducing prediction accuracy.
- Investors and financial institutions increasingly seek advanced models that can provide:
  - Timely and reliable market forecasts.
  - Enhanced decision-making.
  - Reduced risk.

1.2 Problem Statement

The goal behind developing a stock price forecasting system lies in the inherent complexity and volatility of financial markets, where prices are influenced by numerous unpredictable factors such as historical trends, market sentiment, news, and global events. Traditional methods struggle to process and interpret vast amounts of real-time data effectively, leading to inaccurate predictions and missed opportunities. There is a clear need for an advanced system that can integrate diverse data sources, utilize sophisticated machine learning models, and provide more accurate, timely predictions to support better investment decisions and risk management.

- **Volatility & Prediction Challenges**: The stock market is highly volatile and difficult to predict in real-time due to fluctuating data, news events, and investor sentiment.
- **Limitations of Existing Models**:
  - Inability to efficiently process real-time data.
  - Struggle to handle complex dependencies and multiple data sources.
  - Result in inaccurate predictions and delayed insights.
- **Real-time Data Processing**: There is a need for continuous data processing to make timely predictions.
- **Accuracy**: Existing models struggle to capture complex patterns in noisy and volatile market data.
- **Multiple Data Sources**: A comprehensive approach is required to combine stock prices, technical indicators, and sentiment analysis from news and social media.
- **Scalability**: The system should handle high-frequency data and make predictions with low latency.

1.3 Objectives

- To Develop and evaluate hybrid LSTM and GRU models for stock price forecasting, leveraging their ability to capture both spatial and temporal patterns in market data.
- To Design and implement an ensemble stacking model that combines LSTM, GRU, and RNN architectures to enhance prediction accuracy and robustness across various market conditions.
- To Integrate sentiment analysis using BERT and VADER techniques to incorporate real-time news data, assessing the impact of market sentiment on stock price predictions.
- To Compare the performance of deep learning models (LSTM, GRU, RNN) and traditional methods (Random Forest, Gradient Boosting) in terms of accuracy, risk reduction, and real-world applicability.

1.4 Scope

A stock forecasting system predicts future stock prices using advanced machine learning models and real-time data from sources like historical prices, news, and social media. It helps investors and analysts make informed decisions, manage risks, and optimize strategies by identifying market trends and price fluctuations. The system is adaptable, scalable, and enhances financial decision-making across various market conditions. Some key future scopes include:

- Real-time forecasting allows traders to adjust strategies in response to sudden market changes or emerging trends.
- Helps firms monitor trading activities in real-time to ensure they follow regulations and quickly identify and report any suspicious behavior.
- Enables ongoing evaluation of investment strategies and market models to refine and enhance forecasting methods.
- Enables the development of automated trading strategies that can execute trades based on real-time data.

Chapter 2

Literature Review

Sentiment-augmented stock price forecasting integrates market sentiment derived from news articles, social media, and financial reports with traditional predictive models to enhance forecasting accuracy. This approach recognizes that investor sentiment significantly influences market behavior and can precede price movements. This literature review examines existing methodologies, datasets, and the effectiveness of various sentiment analysis techniques in improving stock price predictions.

2.1 Comparative Analysis of Recent Study

In the paper[1]A stock market prediction approach using Transductive Long Short-Term Memory (TLSTM) and social media sentiment analysis , TLSTM enhances accuracy by focusing on recent data, improving short-term trend predictions. To address class imbalance in sentiment data, the Off-Policy Proximal Policy Optimization (PPO) algorithm adjusts rewards to prioritize underrepresented sentiments. This combination of stock data and sentiment analysis offers more accurate market predictions.

The proposed [2]MS-SSA-LSTM model combines multi-source data, including historical stock trading data and sentiment derived from stock forum comments, for improved predictive accuracy. The model utilizes the Sparrow Search Algorithm (SSA) to optimize the hyperparameters of the Long Short-Term Memory (LSTM) network, enhancing its ability to forecast stock prices. By incorporating sentiment analysis and optimizing LSTM parameters, the model shows improved performance, particularly in predicting short-term market movements, outperforming standard LSTM models by an average of 10.74%. The study emphasizes the importance of multi-source data in improving prediction accuracy, suggesting that investor sentiment plays a critical role in stock price fluctuations

In the research paper[3]A Deep Reinforcement Learning (DRL)-based Decision Support System for automated stock trading. It integrates both past and predicted future stock trends to enhance trading decisions. The model uses Gated Recurrent Units (GRUs) to forecast stock prices and incorporates this information into the state space of the Deep-Q Network (DQN) agent, enabling it to make more informed decisions on whether to buy, sell, or hold stocks. Tested on several stock markets, including Tesla, IBM, Amazon, and Chinese stocks, the model demonstrates improved profitability and efficiency in volatile market environments by optimizing trading strategies dynamically rather than relying on static rules.

The paper titled [4]"Decision Fusion for Stock Market Prediction: A Systematic Review" discusses how decision fusion techniques are applied in stock market prediction using machine learning and deep learning models. It highlights that predictions improve when multiple models (base learners) are combined, a method known as decision fusion or ensemble learning. The review examines trends over two decades, focusing on the characteristics of these base learners and the fusion methods applied. It explores different methods for classification and regression tasks and discusses future directions, including integrating new algorithms and sentiment analysis with decision fusion techniques.

The paper titled [5]"Stock Price Prediction Based on Morphological Similarity Clustering and Hierarchical Temporal Memory" introduces a new method for stock price prediction by clustering similar stocks using Morphological Similarity Distance (MSD) combined with k-means clustering. It then applies Hierarchical Temporal Memory (HTM), an online learning model, to learn patterns from these similar stocks, achieving better prediction accuracy compared to models that do not account for stock similarity. The results show that this method, called C-HTM, outperforms baseline models like LSTM and GRU, especially for short-term stock price predictions.

The paper titled [6]"A Deep Learning-Based Approach for Stock Price Prediction Using Bidirectional Gated Recurrent Unit and Bidirectional Long Short Term Memory Model" presents a comparative study between two deep learning models: Bidirectional Gated Recurrent Unit (BiGRU) and Bidirectional Long Short-Term Memory (BiLSTM). Both models are used to predict stock prices, leveraging forward and backward propagation to capture patterns in time-series data. The BiGRU model outperformed BiLSTM in terms of prediction accuracy, stability, and computational efficiency. The study used stock data from the NIFTY-50 index and demonstrated that BiGRU yielded better results across various metrics such as MSE, RMSE, and R2

The paper [7]discusses the application of machine learning techniques for stock price prediction, emphasizing the limitations of traditional methods like fundamental and technical analysis. It highlights the use of Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models due to their ability to handle sequential data, making them suitable for stock market prediction. By analyzing historical stock prices, these models predict future trends with improved accuracy compared to older methods. The paper also touches on the importance of reliable datasets and compares various models, concluding that RNN and LSTM are effective for stock trend prediction.

The paper[8] explores the use of sentiment analysis combined with machine learning to predict stock market movements. It highlights the difficulty in accurately predicting stock trends due to the market's inherent complexity. The authors analyze data from social media platforms like Twitter, extracting sentiment features such as polarity and subjectivity. By applying machine learning techniques, including model stacking, they attempt to improve prediction accuracy. The study examines the stock movements of six companies, achieving an accuracy of up to 60%. The research suggests that augmenting textual features with historical stock data enhances predictability but acknowledges the need for more refined methods.

The paper [9]by Sidra Mehtab and Jaydip Sen proposes a hybrid approach for predicting stock prices using machine learning and deep learning, particularly Convolutional Neural Networks (CNNs). It focuses on the NIFTY 50 index from India's National Stock Exchange over a four-year period (2015-2018) to predict stock movements for 2019. The authors use various classification and regression models to forecast stock values, emphasizing the use of CNNs due to their ability to capture complex patterns in multivariate time series data. They test different CNN configurations, including univariate and multivariate inputs, and find that CNN-based models significantly outperform traditional machine learning methods in prediction accuracy. The study highlights CNN's capability to improve stock price forecasting, especially for medium-term investors.

The paper titled [10]"A Prediction Approach for Stock Market Volatility Based on Time Series Data" explores stock market forecasting through time series analysis, focusing on the Indian stock market. It employs the ARIMA (Auto-Regressive Integrated Moving Average) model to predict stock trends for Nifty and Sensex indices using historical data from 2012 to 2016. The study emphasizes the importance of financial forecasting for decision-making and risk management, highlighting how ARIMA helps transform non-stationary data into a stationary format for accurate predictions. Validation methods, such as the Augmented Dickey-Fuller and L-Jung Box tests, confirm the reliability of the predictions, with the results showing a mean error margin of around 5%. This forecasting model is proposed as useful not only in finance but also in various other domains like health and education.

The paper [11]"Predicting Prices of Stock Market using Gated Recurrent Units (GRUs) Neural Networks" explores the application of GRUs, a type of recurrent neural network, for predicting stock market prices. The authors propose modifications to the GRU's internal structure to overcome challenges such as the vanishing gradient problem and local minima, aiming to improve efficiency and reduce computation time. Using real-time stock data from Yahoo Finance, the model is trained and evaluated through mini-batch gradient descent, with accuracy measured via Root Mean Square Error (RMSE). Results show that GRUs effectively learn patterns from historical data and outperform other recurrent networks like LSTMs in certain aspects. The paper concludes with future prospects of developing automated trading agents based on the proposed system.

The paper [12] proposes a deep learning model to predict short-term stock trends using a two-stream Gated Recurrent Unit (TGRU) network. It integrates financial news articles and historical stock prices, leveraging both sentiment analysis and Stock2Vec, a sentiment embedding model trained on financial data. The study conducts two experiments: one using S&P 500 index data and news from Reuters and Bloomberg, and another using the VN-index and VietStock news. The results demonstrate that the TGRU outperforms other models, achieving higher accuracy and effectively predicting stock price directions in both datasets. The model also shows robustness in real-world trading simulations.

Table 2.1: Comparative Analysis of Recent Studies

Sr. No	Title	Author(s)	Year	Methodology	Drawback
1	Stock Market Prediction With Transductive Long Short-Term Memory and Social Media Sentiment Analysis	Ali Peivandizadeh, Sima Hatami, Amirhossein Nakhjavani, Lida Khoshsima, Mohammad Reza Chalak Qazani, Muhammad Haleem, Roohallah Alizadehsani	2024	Combines TLSTM (Transductive LSTM) for stock price prediction with social media sentiment analysis. Uses Off-Policy Proximal Policy Optimization (PPO) to handle class imbalances.	Over-reliance on sentiment from social media.Complex architecture requires significant computational resources. Data imbalances in sentiment analysis can still affect the model's performance.
2	A Stock Price Prediction Model Based on Investor Sentiment and Optimized Deep Learning	Guangyu Mu, Nan Gao, Yuhan Wang, Li Dai	2023	MS-SSA-LSTM model integrates multi-source data, such as: Investor sentiment from forums , Stock trading data. Optimizes LSTM (Long Short-Term Memory) using the Sparrow Search Algorithm. Uses a sentiment dictionary to improve prediction accuracy.	model heavily depends on sentiment data Sentiment categorization is limited to positive and negative emotions. Lacks integration of more complex emotional aspects such as fear, anger, etc. .
3	A Deep Reinforcement Learning-Based Decision Support System for Automated Stock Market Trading	Yasmeen Ansari, Sadaf Yasmin, Sheneela Naz, Hira Zaffar, Zeeshan Ali, Jihoon Moon, Seungmin Rho	2022	Deep Reinforcement Learning (DRL) framework using Deep-Q Networks with Gated Recurrent Unit (GRU) forecasting. The model uses both past and future stock trends to make trading decisions.	Model performance is volatile and may vary with different training attempts. Lacks robustness in highly volatile markets, and future price predictions are sometimes inaccurate, affecting trading decisions.
4	Decision Fusion for Stock Market Prediction: A Systematic Review	Cheng Zhang, Nilam N. A. Sjarif, Roslina B. Ibrahim	2022	Decision fusion: Combines forecasts from multiple models using base learners and fusion methods	Few studies use decision fusion.Lack of diversity in ensemble models.Minimal integration of sentiment analysis.

Sr. No	Title	Author(s)	Year	Methodology	Drawback
5	Stock Price Prediction Based on Morphological Similarity Clustering and Hierarchical Temporal Memory	Xingqi Wang, Kai Yang, Tailian Liu	2022	K-means clustering with Morphological Similarity Distance (MSD) to group stocks, followed by Hierarchical Temporal Memory (HTM) for prediction	. The method may not generalize well to long-term predictions.Multivariate input had minimal benefit for HTM. Potential underfitting or overfitting of baseline models due to lack of parameter tuning.
6	A Deep Learning-Based Approach for Stock Price Prediction Using Bidirectional Gated Recurrent Unit and Bidirectional Long Short Term Memory Model	Md. Ebtidaul Karim, Sabrina Ahmed	2021	BiGRU (Bidirectional Gated Recurrent Unit) with activation layer andBiLSTM (Bidirectional Long Short Term Memory).	Increased complexity with more hidden layers. Trainable parameters grow with additional layers, which can increase computation cost. Limited to comparison between BiGRU and BiLSTM models.
7	Literature Survey on Stock Price Prediction Using Machine Learning	Anusha J Adhikar, Apeksha K Jadhav, Charitha G, Karishma KH, Mrs. Supriya HS	2020	Machine Learning: Recurrent Neural Networks (RNN), Long Short Term Memory (LSTM). Data collected from Yahoo Finance.	Python libraries used were not optimal for training speed. Need for optimization of neural network parameters.
8	Augmented Textual Features-Based Stock Market Prediction	Salah Bouktif, Ali Fiaz, Mamoun Awad	2020	Sentiment analysis using tweets and stock data combined with machine learning techniques such as SVM, Naive Bayes, ANN, and XGBoost with model stacking.	Limited by the text features extraction approach, Detailed analysis of sentiments is needed to improve prediction accuracy

Sr. No	Title	Author(s)	Year	Methodology	Drawback
9	Stock Price Prediction Using Convolutional Neural Networks on a Multivariate Time-series	Sidra Mehtab, Jaydip Sen	2019	Machine Learning: Logistic Regression, KNN, Decision Tree, Bagging, Boosting, Random Forest, ANN, SVM. Deep Learning: CNN with three approaches: univariate, multivariate, and multi-model. Data from NIFTY 50 index (2015- 2019).	Difficulty in predicting rapid stock price changes. Performance varies with data size and model settings. Higher prediction errors (RMSE) in some CNN configurations.
10	A Prediction Approach for Stock Market Volatility Based on Time Series Data	Sheikh Mohammad Idrees, M. Afshar Alam, Parul Agarwal	2019	ARIMA (Auto-Regressive Integrated Moving Average) model for time series data prediction	Limited to univariate data, assuming no external factors like social or economic events.
11	Predicting Prices of Stock Market using Gated Recurrent Units (GRUs) Neural Networks	Mohammad Obaidur Rahman, Md. Sabir Hossain, Ta-Seen Junaid, Md. Shafiul Alam Forhad, Muhammad Kamal Hossen	2019	Gated Recurrent Units (GRUs) for stock price prediction Mini-batch gradient descent Real-time dataset from Yahoo Finance	Local minima problem and time complexity of stochastic gradient descent were reduced but still may arise.
12	Deep Learning Approach for Short-Term Stock Trends Prediction Based on Two-Stream Gated Recurrent Unit Network	Dang Lien Minh, Abolghasem Sadeghi-Niaraki, Huynh Duc Huy, Kyungbok Min, Hyeonjoon Moon	2020	Two-Stream Gated Recurrent Unit (TGRU) Network, Sentiment analysis with Stock2Vec, financial news data, and technical indicators.	Complexity of TGRU model, requiring long training times and computational resourcesLimited to daily stock movements rather than intra-day trading.

Chapter 3

Project Design

Project design refers to the process of defining and organizing all the key aspects of a project before execution. It involves planning how the project will be carried out to achieve its goals and objectives efficiently. Project design serves as a blueprint or roadmap for managing the project and includes details on scope, deliverables, resources, timeline, and methodology.

3.1 Proposed System Architecture

The web application for stock forecasting will utilize a combination of advanced machine learning techniques to deliver accurate and timely predictions. It will primarily employ Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models to analyze historical stock data, leveraging their strengths in capturing temporal patterns and long-term dependencies inherent in financial time series. Simultaneously, a Recurrent Neural Network (RNN) will process real-time news headlines gathered through web scraping. This RNN model will perform sentiment analysis, generating sentiment scores based on the tone and content of the news, which can significantly influence stock market movements. By incorporating real-time sentiment data, the system captures external factors that may not be reflected in historical data alone.

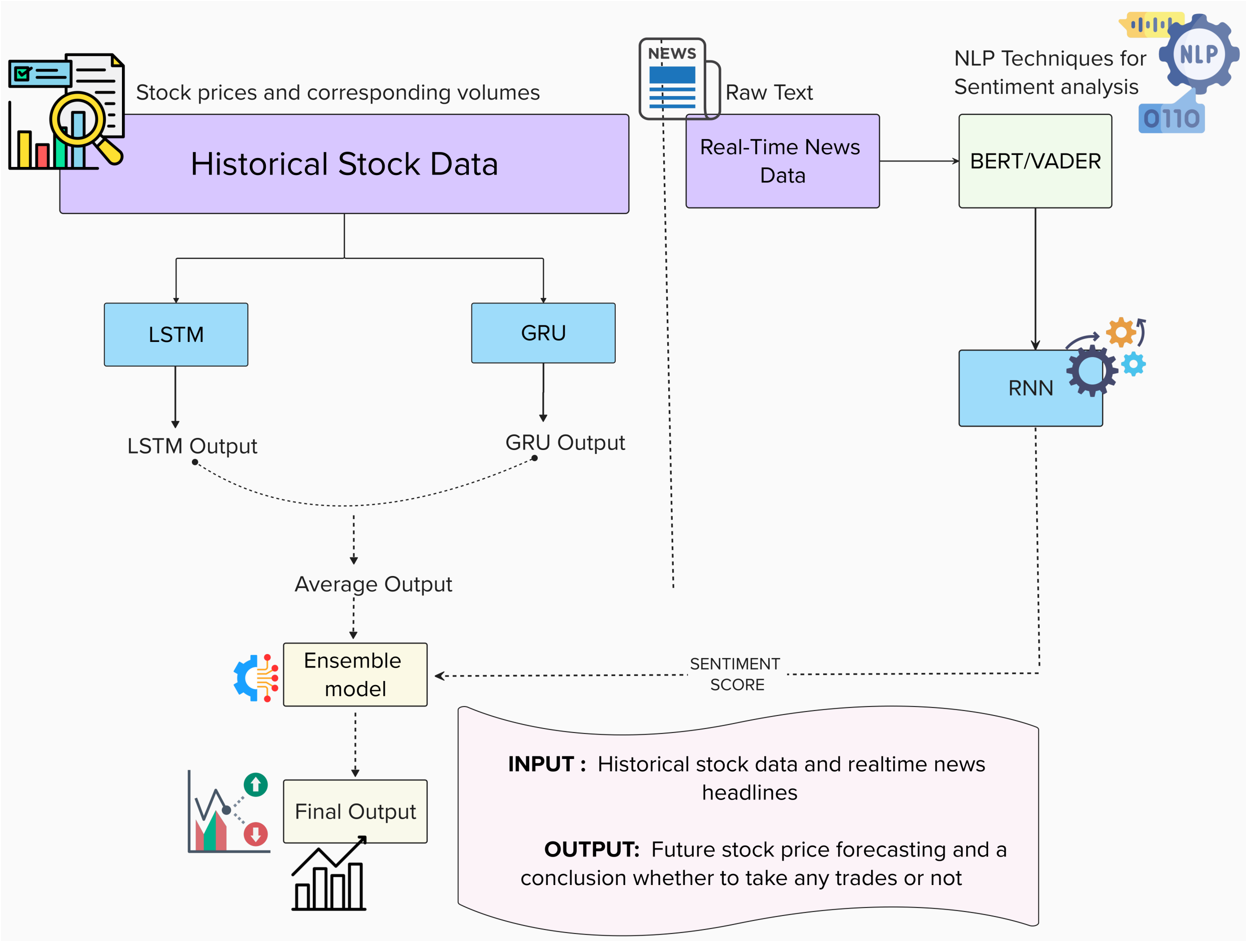


Figure 3.1: System Architecture

An ensemble model will then integrate the outputs from the LSTM, GRU, and RNN sentiment analysis to produce a more robust and comprehensive stock forecast, offering investors and traders a well-rounded prediction tool that accounts for both quantitative data and qualitative market sentiment. This integration aims to improve prediction accuracy, helping users make informed trading decisions in real time.

Recurrent Neural Networks (RNNs) are designed to process sequential data by maintaining a hidden state that captures information from previous inputs, making them suitable for tasks like time series forecasting and language modeling. However, RNNs often struggle with long-range dependencies due to issues such as vanishing gradients. Long Short-Term Memory (LSTM) networks address this limitation by introducing memory cells and gates (input, output, and forget gates) that regulate the flow of information, allowing them to retain long-term dependencies more effectively. This makes LSTMs particularly useful in applications like natural language processing and speech recognition. Gated Recurrent Units (GRUs) offer a simpler alternative to LSTMs by combining the input and forget gates into a single update gate, resulting in fewer parameters while still performing well on similar tasks. Overall, while RNNs serve as a foundational architecture for sequential data, LSTMs and GRUs are advanced models that excel in capturing long-term dependencies.

3.2 Data Flow Diagrams(DFD)

The workflow of the real-time stock price prediction system starts by collecting data from multiple sources, including stock prices via an API, news feeds, and social media. This data is ingested in real time using Apache Kafka and then processed through various steps such as data cleaning, feature engineering, and sentiment analysis to extract relevant information. The processed data is fed into deep learning models, including CNN-LSTM and CNN-GRU, which are designed to capture complex patterns in stock movements. Additionally, an RNN model handles real-time preprocessing for sequential data predictions. The predictions from these models are combined using an ensemble approach to generate the final stock price predictions, which are then displayed on a prediction dashboard for user insights. This system effectively integrates diverse data sources and advanced machine learning techniques for accurate, real-time stock trend forecasting.

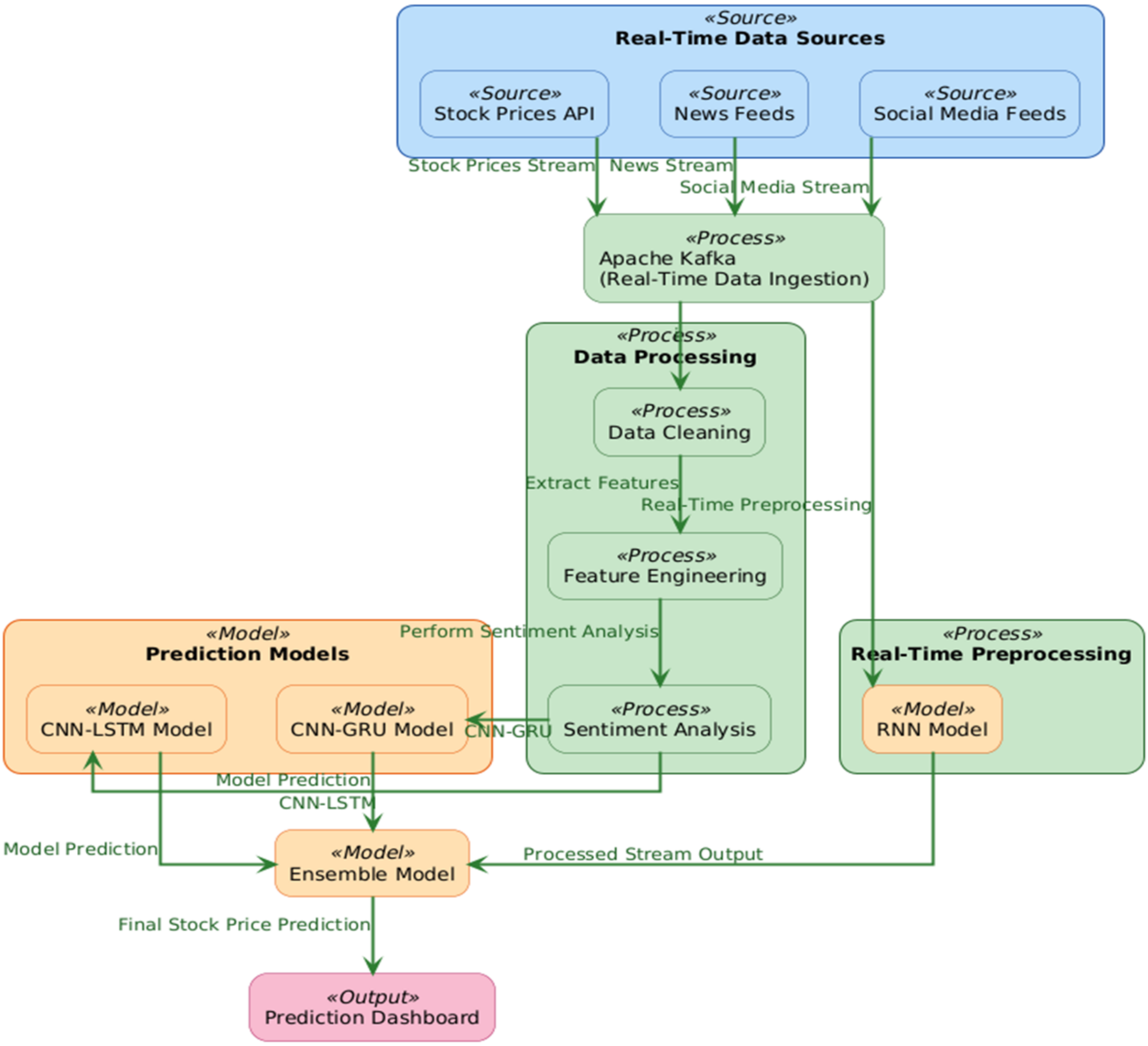


Figure 3.2: Data flow diagram of forecasting system

The web application for stock forecasting will utilize advanced machine learning techniques to provide accurate predictions. It will gather historical stock data from financial databases or APIs and obtain real-time news headlines through web scraping. The Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models will analyze the historical stock data to identify patterns and trends, while a Recurrent Neural Network (RNN) will process the scraped news headlines to generate sentiment scores that reflect market sentiment. The outputs from both the LSTM and GRU models, along with the sentiment scores from the RNN, will be integrated by an ensemble model to produce a comprehensive stock price forecast. Finally, this forecast will be displayed to the user through the web application interface, providing valuable insights into potential future stock movements.

The real-time stock price prediction system integrates multiple data sources, including stock prices via an API, news feeds, and social media feeds, to gather essential information for forecasting. Data ingestion is managed using Apache Kafka, which streams the collected data in real time. The data undergoes several processing steps: it is cleaned for accuracy, features relevant to stock prediction (such as sentiment, price trends, and financial indicators) are engineered, and sentiment analysis is conducted on news and social media data. A preprocessing stream prepares the data for real-time predictions using a Recurrent Neural Network (RNN) model. The system employs advanced deep learning models, including a CNN-LSTM hybrid model for feature extraction and sequential prediction, as well as a CNN-GRU hybrid model for trend prediction. The outputs from both models are integrated into an Ensemble Model to generate the final stock price prediction. Ultimately, the predicted stock prices are displayed on a Prediction Dashboard, providing users with insights into expected market movements.

3.3 Use Case Diagrams

A use case diagram for stock price forecasting illustrates the interactions between various users and the forecasting system. Key actors include the Investor, who seeks to make informed investment decisions; the Analyst, who analyzes market trends; and the Data Source, which provides historical stock prices and relevant market data. The primary use cases encompass several functionalities: users can input historical data into the system, which allows for analyzing market trends to inform predictions. The core function, generate forecast, employs algorithms to predict future stock prices based on historical data. Users can then visualize the forecast through graphs or charts, facilitating better understanding and interpretation. Additionally, there's an option to evaluate forecast accuracy by comparing predicted prices with actual outcomes, enabling users to refine their approaches. Finally, the Investor leverages these forecasts to make investment decisions regarding buying or selling stocks. This diagram effectively captures the essential workflows and relationships involved in the stock price forecasting process.

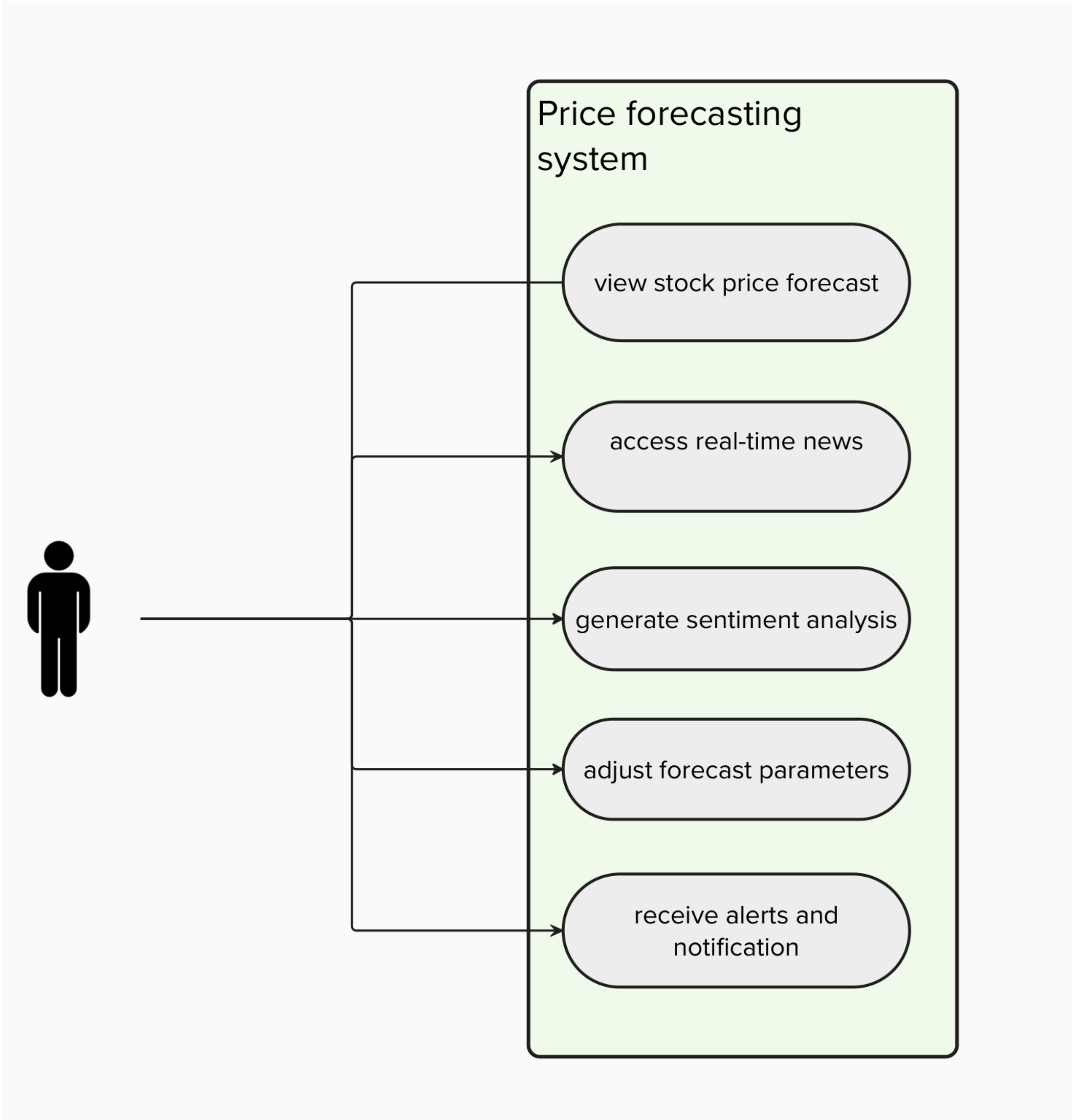


Figure 3.3: Use Case Diagram

The use case diagram for the stock forecasting system outlines key interactions between the user and the system, emphasizing five main use cases:

- **View Stock Price Forecast**: Users can access predicted stock prices to make informed investment decisions.
- **Access Real-Time News**: Users can read the latest news articles relevant to the stock market, helping them stay updated on events that may affect stock prices.
- **Generate Sentiment Analysis**: This feature allows users to analyze news sentiment, providing insights into market trends based on the tone of news articles.

- **Adjust Forecast Parameters**: Users can modify parameters such as forecasting models and timeframes, enabling customization to suit their investment strategies.
- **Receive Alerts and Notifications**: Users can set up alerts for significant price changes or sentiment shifts, ensuring they stay informed of critical developments.

Overall, the diagram illustrates the system's functionality, highlighting its capability to support informed investment decisions and adapt to user preferences, making it a valuable tool for investors and analysts.

Chapter 4

Project Implementation

The implementation of a stock forecasting system is a critical endeavor in today's fast-paced financial markets, where timely and accurate predictions can significantly influence investment decisions. This system leverages advanced machine learning techniques and real-time data from various sources, including historical stock prices, financial news, and social media sentiment, to forecast future stock price movements.

The code utilizes the yfinance library to extract stock price data for financial analysis and modeling. This library provides an easy-to-use interface to access historical market data from Yahoo Finance, enabling users to retrieve essential financial information for various stocks. The process begins by importing the yfinance library, which is a necessary step for fetching stock data and utilizing its built-in functions. Once the library is imported, a specific stock ticker is initialized using the 'yf.Ticker()' function. For example, a ticker such as "AAPL" (Apple Inc.) can be passed into the function to establish a connection to Yahoo Finance's database. This connection enables the user to access detailed information related to the chosen stock, such as historical prices, dividends, and company data, which can then be used for further financial analysis and modeling. This efficient method simplifies the process of accessing and working with stock market data, making it valuable for traders, analysts, and developers.

```
[ ]  START = "2015-01-01"
     # strf --> string format time
     TODAY = date.today().strftime("%Y-%m-%d")

[ ]  # function for loading data
     def load_data(ticker):
       data = yf.download(ticker, START, TODAY)
       data.reset_index(inplace=True)
       return data

[ ]  data = load_data('AAPL')
     df = data
     df.head()
```

Figure 4.1: Extraction Of Stock Data

The graph illustrates the movement of Apple's stock prices over time, focusing on the 100-day moving average. The X-axis represents over six years of trading days, while the Y-axis shows stock prices rising from below $50 to over 200$, indicating significant growth. The blue line reflects daily closing prices, showing volatility with peaks and dips, while the red line, representing the 100-day moving average, smooths these fluctuations to reveal a clearer upward trend. Key observations include overall growth, particularly after the 1000th day, and periods of volatility around the 1200th and 1600th days. The moving average helps highlight long-term trends despite short-term fluctuations.

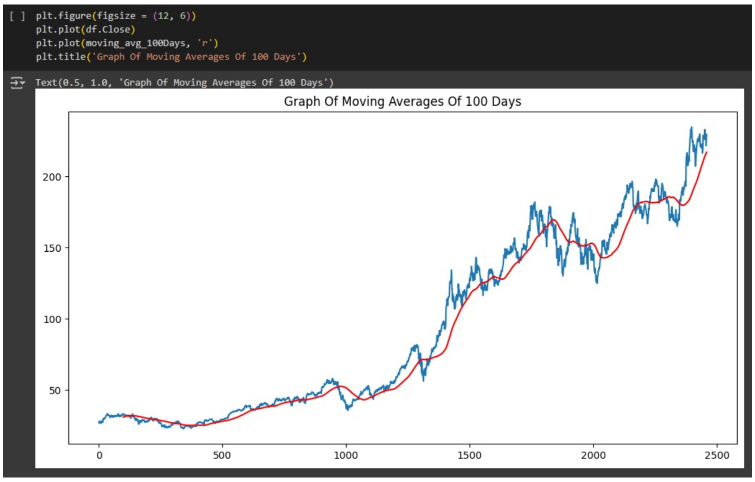


Figure 4.2: Visualization Of Data

The graph compares Apple's 100-day and 200-day moving averages to provide a broader view of stock trends. The X-axis represents several years of trading, while the Y-axis shows stock prices rising from under $50 to over 200$. The blue line depicts daily stock prices, with the red 100-day moving average responding more quickly to short-term fluctuations, and the green 200-day moving average providing a more stable, long-term trend. Key observations include crossovers between the two lines, with the "Golden Cross" signaling bullish momentum and the "Death Cross" indicating potential downturns.

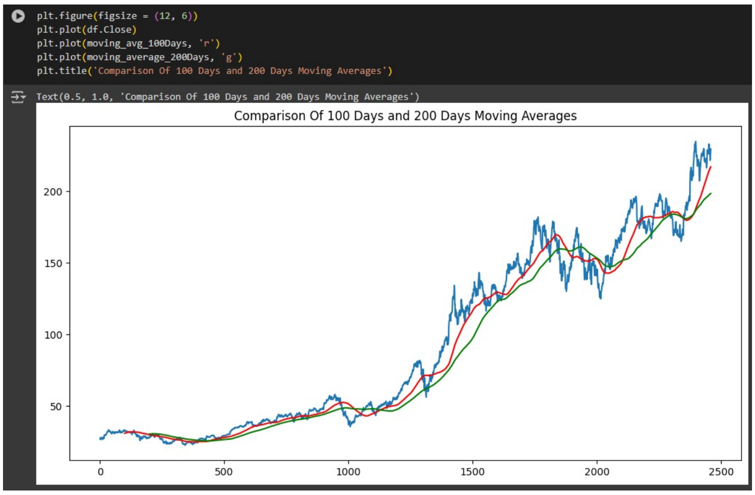


Figure 4.3: Comparison Between Stock Prices of 100 and 200 days data

The graph compares actual and predicted stock prices over time, showing that the model's predictions (red line) closely follow the real stock prices (blue line). Despite minor deviations during rapid price changes, the predictions generally align well with the actual data, capturing both upward and downward trends effectively. This demonstrates the model's strong accuracy in forecasting stock price movements over the observed period.

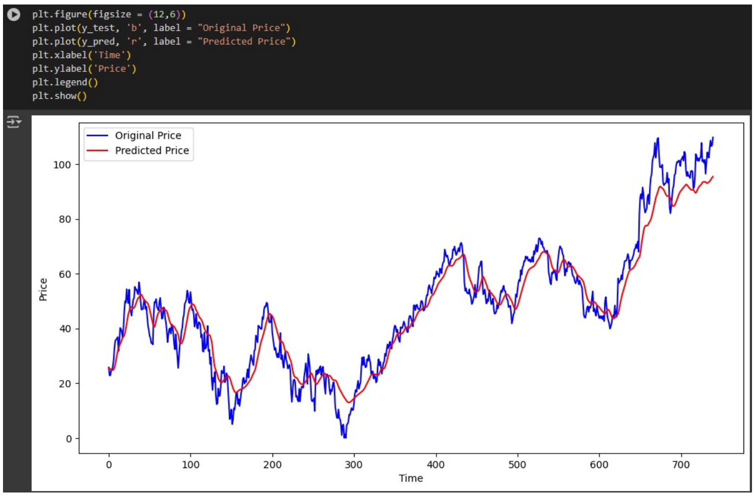


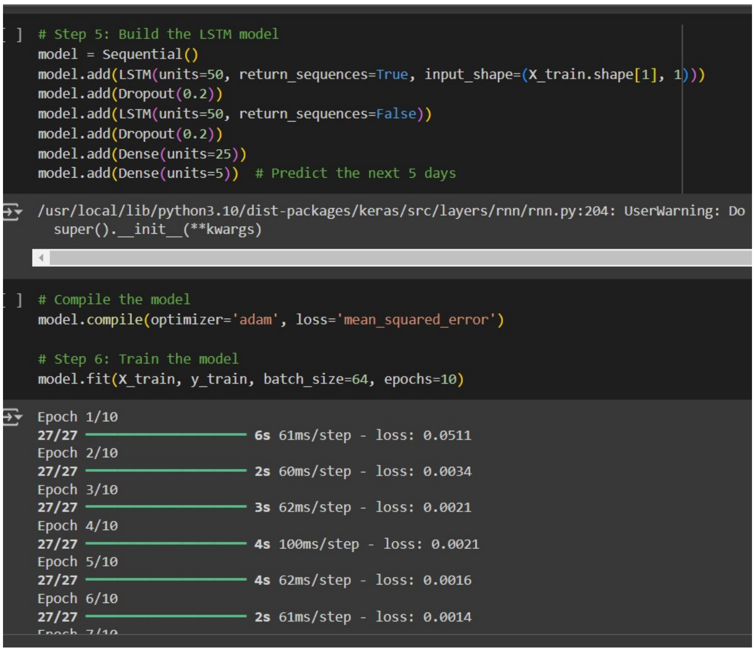
Figure 4.4: Difference between the actual and predicted price

The code fetches historical stock data for Apple (AAPL) from 2015 to 2023, extracts the closing prices, and normalizes them using 'MinMaxScaler'. It splits the data into 80% training and 20% testing sets. A function is then used to create time-step sequences for training an LSTM model, which helps in predicting stock prices for multiple future steps based on past data.

```
# Step 2: Fetch historical stock data
ticker = 'AAPL'  # Change this to any other stock symbol if you prefer
data = yf.download(ticker, start="2015-01-01", end="2023-10-01")

[*********************100%***********************]  1 of 1 completed

# Step 3: Preprocess the data
# Use the 'Close' price for prediction
close_prices = data['Close'].values
close_prices = close_prices.reshape(-1, 1)

# Scale the data
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(close_prices)

# Split the data into training and testing sets (80% training, 20% testing)
training_size = int(np.ceil(len(scaled_data) * 0.8))
train_data = scaled_data[0:training_size]
test_data = scaled_data[training_size:]

# Step 4: Prepare the data for the LSTM model
def create_multistep_dataset(data, time_step=60, prediction_steps=5):
    X, y = [], []
    for i in range(len(data) - time_step - prediction_steps):
        X.append(data[i:(i + time_step), 0])
        y.append(data[i + time_step:(i + time_step + prediction_steps), 0])
    return np.array(X), np.array(y)
```

Figure 4.5: preparing data for lstm model

The code builds, compiles, and trains an LSTM neural network model using Keras. It features two LSTM layers with 50 units each, a 0.2 dropout rate to prevent overfitting, and Dense layers predicting the next 5 values. The model is compiled with the Adam optimizer and mean squared error loss function, then trained on ‘Xtrain‘ and ‘ytrain‘ for 10 epochs with a batch size of 64, showing decreasing loss values throughout training.



```
# Step 5: Build the LSTM model
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(X_train.shape[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=False))
model.add(Dropout(0.2))
model.add(Dense(units=25))
model.add(Dense(units=5))  # Predict the next 5 days

/usr/local/lib/python3.10/dist-packages/keras/src/layers/rnn/rnn.py:204: UserWarning: Do
  super().__init__(**kwargs)

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')

# Step 6: Train the model
model.fit(X_train, y_train, batch_size=64, epochs=10)

Epoch 1/10
27/27 ━━━━━━━━━━━━━━━━━━━━ 6s 61ms/step - loss: 0.0511
Epoch 2/10
27/27 ━━━━━━━━━━━━━━━━━━━━ 2s 60ms/step - loss: 0.0034
Epoch 3/10
27/27 ━━━━━━━━━━━━━━━━━━━━ 3s 62ms/step - loss: 0.0021
Epoch 4/10
27/27 ━━━━━━━━━━━━━━━━━━━━ 4s 100ms/step - loss: 0.0021
Epoch 5/10
27/27 ━━━━━━━━━━━━━━━━━━━━ 4s 62ms/step - loss: 0.0016
Epoch 6/10
27/27 ━━━━━━━━━━━━━━━━━━━━ 2s 61ms/step - loss: 0.0014
```

Figure 4.6: building and training lstm model

The graph illustrates the prediction of Apple (AAPL) stock prices for the next five days based on an LSTM model. The blue line represents the actual stock prices over the past 60 days, showcasing fluctuations and a general downward trend as it nears late September 2023. The orange line with dots indicates the model’s predicted stock prices for the next five days, starting from October 1, 2023. The prediction suggests an upward trend, with stock prices expected to rise slightly after a period of decline. This chart visually compares the historical price movement with the model’s future price predictions to assess potential trends.

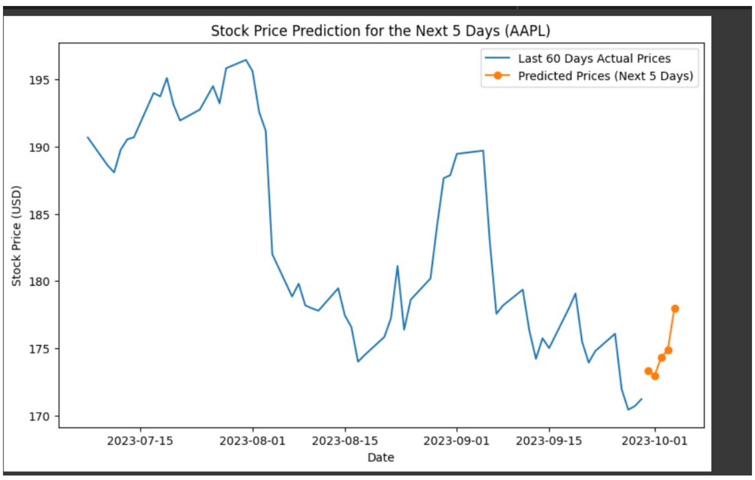


Figure 4.7: visualizing the lstm model result

4.1 Timeline Sem VII

Project scheduling is the process of organizing tasks, resources, and timelines to ensure a project is completed efficiently and on time. It involves defining all tasks, identifying dependencies between them, allocating resources, and setting deadlines. This helps project managers coordinate activities, monitor progress, and manage the overall flow of work. A Gantt chart is one of the most widely used tools for project scheduling. It provides a visual representation of the project timeline, where each task is shown as a horizontal bar across a calendar. The length of the bar reflects the task's duration, while dependencies between tasks are illustrated with connecting lines, showing which tasks must be completed before others can start. Gantt charts also include milestones, marking significant deadlines or events. This tool is highly effective for tracking progress, managing resources, and providing a clear overview of the project timeline, making it easier to communicate the plan with stakeholders and team members.

The project timeline for the Stock Price Forecasting System begins with Problem Identification and Need Assessment, where our team conducted discussions to pinpoint key challenges and establish the necessity for developing a forecasting system. Following this, we moved to Title Finalization, where we agreed on a clear and concise project title. Next, in the Abstract and Objectives Development phase, we collaborated to draft a comprehensive abstract that outlines the goals and purpose of the project. We then proceeded to the Research and Technology Study, reviewing various IEEE research papers to explore the essential technologies needed for our system's development, during which we also identified potential areas for improvement and innovation within existing methodologies. This led to our Scope and Technology Stack Discussion, where we discussed the project's overall scope and determined the appropriate technology stack for implementation. In the Applications and Advantages Analysis phase, we explored the various applications and advantages that our proposed forecasting system could offer.

Finally, we entered the Implementation Phase, initiating the process by developing and implementing two models: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), specifically designed for stock price forecasting. This structured approach ensured that we covered all necessary aspects of the project comprehensively and effectively.

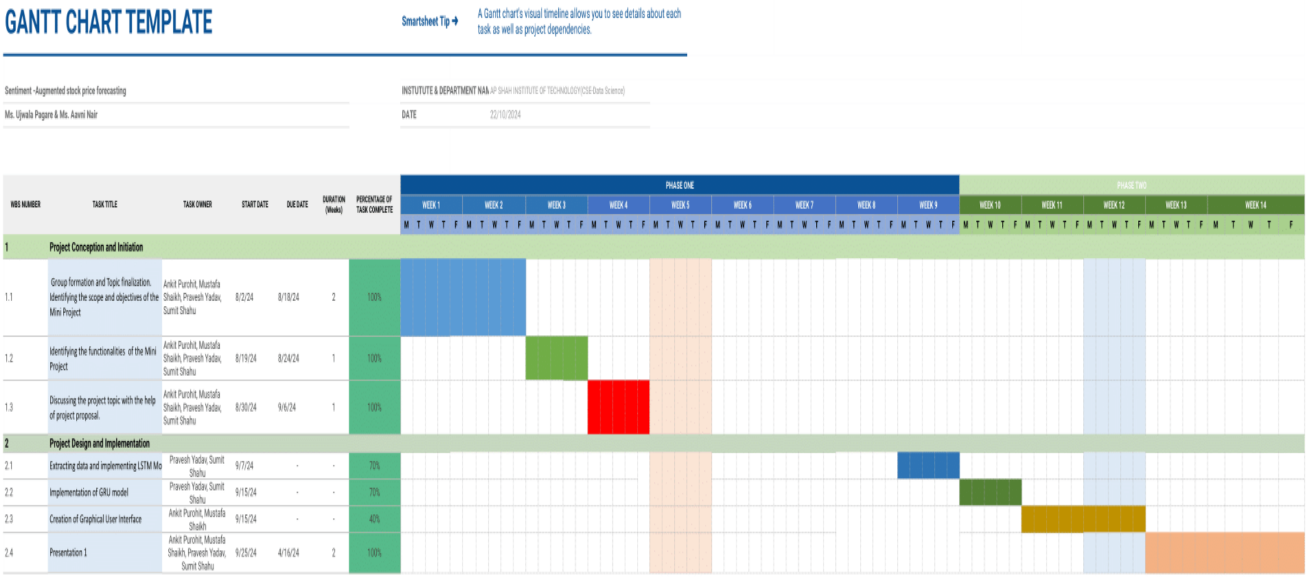


Figure 4.8: Gantt Chart

Chapter 5

Conclusion

The stock forecasting web application stands at the forefront of technological innovation, utilizing advanced machine learning methodologies to deliver precise stock price predictions. By integrating extensive historical stock data sourced from various financial APIs with real-time news headlines acquired through sophisticated web scraping techniques, the application constructs a dynamic framework for analysis. It leverages the strengths of complex models such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) to meticulously analyze historical trends and patterns that serve as a foundation for forecasting future price movements. Simultaneously, the implementation of a Recurrent Neural Network (RNN) enables the system to process real-time news articles effectively, generating sentiment scores that provide a quantifiable measure of market sentiment, derived directly from current news events.

Looking ahead, the future scope of this stock forecasting system is promising. There are numerous avenues for enhancement, such as incorporating additional data sources, including social media sentiment analysis and macroeconomic indicators, which could further refine the predictive capabilities of the models. Additionally, integrating more advanced machine learning techniques, such as reinforcement learning, could allow the system to adapt dynamically to changing market conditions. Furthermore, the implementation of user-customizable features, such as personalized alerts and tailored forecasting options, would enhance user engagement and satisfaction. As the application evolves, it has the potential to become an indispensable tool for both individual investors and financial analysts, providing deeper insights and fostering more strategic investment approaches in an increasingly volatile financial landscape.

Bibliography

[1] J. Liu, Z. Wang, and B. Zheng, "A Deep Reinforcement Learning-Based Decision Support System for Automated Stock Market Trading," IEEE Access, Volume 8, Pages 143676-143686, 2020.

[2] Sheikh Mohammad Idrees, M. Afshar Alam, Parul Agarwal, "A Prediction Approach for Stock Market Volatility Based on Time Series Data," IEEE Access, Volume 7, Pages 17287-17295, 2019.

[3] Jinyang Du, John S. Kimball, Justin Sheffield, Ming Pan, Colby K. Fisher, Hylke E. Beck, Eric F. Wood, "Satellite Flood Inundation Assessment and Forecast Using SMAP and Landsat," IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, Volume 14, Pages 6707-6715, 2021.

[4] Xingqi Wang, Kai Yang, and Tailian Liu, "Stock Price Prediction Based on Morphological Similarity Clustering and Hierarchical Temporal Memory," IEEE Access, Volume9, Pages 67241-67248, 2021.

[5] Shwetha Salimath, Triparna Chatterjee, Titty Mathai, Pooja Kamble, Megha Kolhekar, "Prediction of Stock Price for Indian Stock Market: A Comparative Study using LSTM and GRU," Proceedings of the International Conference on Advances in Computing and Data Sciences (ICACDS), Pages 123-130, 2020 .

[6] Salah Bouktif, Ali Fiaz, and Mamoun Awad, "Augmented Textual Features-Based Stock Market Prediction," IEEE Access, Volume 8, Pages 40269-40279, 2020.

[7] Avraam Tsantekidis, Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, Alexandros Iosifidis, "Forecasting Stock Prices from the Limit Order Book using Convolutional Neural Networks," 2017 5th International Conference on Cloud Computing and Intelligence Systems (CCIS), Pages 1-7, 2017.

[8]] Mohammad Obaidur Rahman, Md. Sabir Hossain, Ta-Seen Junaid, Md. Shafiul Alam Forhad, Muhammad Kamal Hossen, "Predicting Prices of Stock Market using Gated Recurrent Units (GRUs) Neural Networks," IJCSNS International Journal of Computer Science and Network Security, Volume 19, Issue 1, Pages 213-222, January 2019.

[9] Md. Ebtidaul Karim, Sabrina Ahmed, "A Deep Learning-Based Approach for Stock Price Prediction Using Bidirectional Gated Recurrent Unit and Bidirectional Long Short Term Memory Model," IEEE, Volume 2021, Pages 1-6, 2021.

[10] Rubell Marion Lincy G, Nevin Selby, Aditya Taparia, "An Efficient Stock Price Prediction Mechanism Using Multivariate Sequential LSTM Autoencoder," IEEE, Volume 2023, Pages 1-15, 2023.

[11] Tej Bahadur Shahi, Jitendra Pandey, Junaid Ahmad, ”Stock Price Forecasting with Deep Learning: A Comparative Study,” Mathematics, Volume 8, Pages 1441, 2020.

[12]]Md. Ebtidaul Karim, Mohammed Foysal, ”Stock Price Prediction Using Bi-LSTM and GRU-Based Hybrid Deep Learning Approach,” IEEE, Volume 2021, Pages 1-10, 2021.

[13] Ying Xu, Saiyan Wang, Junhui Li, Zhe Liu, ”Stacked Deep Learning Structure with Bidirectional Long-Short Term Memory for Stock Market Prediction,” Neural Computing, Springer, Volume 2020, Pages 89-106, 2020.

[14] Karim Md. Ebtidaul, Foysal Mohammed, Sabrina Ahmed, ”Bitcoin Candlestick Prediction with Deep Neural Networks,” IEEE, Volume 2021, Pages 17-25, 2021.

Appendices

Detailed information, lengthy derivations, raw experimental observations etc. are to be presented in the separate appendices, which shall be numbered in Roman Capitals (e.g. "Appendix I"). Since reference can be drawn to published/unpublished literature in the appendices these should precede the "Literature Cited" section.

Appendix-A: NS2 Download and Installation

1. Download ns-allinone-2.35.tar.gz from http://sourceforge.net/projects/nsnam/

2. Place ns-allinone-2.35.tar in your desired directory; like /home/vishal.

3. Go to terminal and do as following commands
sudo apt-get update
sudo apt-get install automake autoconf libxmu-dev build-essential

4. Extract ns-allinone-2.35 and after extracting go to folder ns-allinone-2.35 from Terminal as
$cd ns-allinone-2.35
$./install

5. Path Setting
$ gedit .bashrc

This command will open an existing file in editor. Just put the following path which is given bellow. [Remember that our ns-allinone path is /home/vishal. we will change this path according to our ns-allinone folder's path]

export PATH=$PATH:/home/vishal/ns-allinone-2.35/bin:/home/vishal/ns-allinone-2.35/tcl8.5.10/unix/home/vishal/ns-allinone-2.35/tk8.5.10/unix

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/home/vishal/ns-allinone- 2.35/otcl-1.14:/home/vishal/ns-allinone-2.35/lib

export TCL_LIRARY_PATH=$TCL_LIBRARY_PATH:/home/vishal/ns-allinone-2.35/tcl8.5.10/library

After this save and exit.

6. Now type in terminal to check that, is all command we entered in .bashrc is correct or not? And To take the effect immediately
$source .bashrc

7. Then perform the validation test using this command.
$ ./validate

8. Run ns2 using this command
$ns

We will get % prompt in our terminal. Now ns2 has been installed.