The University of Hong Kong
COMP4801 Final Year Project

Empowering Financial Decisions with AI-Driven Forecasting and Personalized Investing

Interim Report

Lam Ngok Fung Herman (3035928328)
Wang Yujing (3035835595)
Sun Haoyu (3035856044)
Muhammad Ghassan Jawwad (3035834802)

Project Supervisor: Dr. Difan Zou

26th January, 2025

Abstract

In today's stock market, investors often struggle to navigate existing financial websites and extract meaningful insights. This report presents the development of an AI-driven financial management platform designed to enhance stock market decision-making for novice investors. The platform integrates a Natural Language Processing (NLP) model developed using Long Short-Term Memory (LSTM) networks and Bidirectional Encoder Representations from Transformers (BERT) architectures to analyse sentiment from textual data and predict stock prices based on historical trends. Additionally, a paper trading system will be implemented, allowing users to practice trading without financial risk while leveraging real-time market data from stock price APIs. By developing this integrated model, we anticipate significant improvements in the accuracy of stock price forecasting.

Preliminary results demonstrate the effectiveness of three distinct models in predicting stock price movements: LSTM for predicting price movement direction, LSTM for precise stock price forecasting, and the LSTM-BERT combined model for accurate price predictions, with LSTM-BERT yielding the best performance. Future work will focus on further optimizing the developed models and developing a functional website. This project aims to enhance user experience and improve investment strategies, ultimately contributing to better financial decision-making.

Acknowledgement

We would like to express our sincere gratitude to our project supervisor, Dr. Difan Zou, for his guidance and support during the development process of our final year project. The valuable feedback and encouragement have played a crucial role in helping us overcome the challenges we have faced throughout this project.

Table of Contents

List of Figures

List of Tables

Abbreviations

Abbreviation	**Definition**
ADABOOST	Adaptive Boosting
ADX	Average Directional Movement Index
AI	Artificial Intelligence
API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
ATR	Average True Range
BERT	Bidirectional Encoder Representations from Transformers
CCI	Commodity Channel Index
CNN	Convolutional Neural Network
DOM	Document Object Model
DRF	Django Rest Framework
EMA	Exponential Moving Average
EMH	Efficient Market Hypothesis
LSTM	Long Short-Term Memory
MACD	Moving Average Convergence Divergence
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MSE	Mean Squared Error
NLP	Natural Language Processing
REST	Representational State Transfer
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RSI	Relative Strength Index
XGBOOST	eXtreme Gradient Boosting

1 Project Background

This chapter provides a comprehensive overview of the project. It begins with an outline of the project's background and motivation, followed by an explanation of the project scope.

1.1 Background

The stock market plays a crucial role in the allocation of capital within the economy, facilitating companies in raising financial resources. Given the ability of investors to trade shares freely, stock prices reflect collective sentiments regarding each company's current and anticipated performance. The challenge of predicting stock prices has been a topic of extensive debate since the 20th century. The Efficient Market Hypothesis [1] posits that in a fully efficient market, it is impossible to outperform the market, as all available and relevant information about the underlying assets is already reflected in stock prices. Based on this theory, the Random Walk Theory [2] asserts that the stock market is efficient, making short-term price forecasting futile. In contrast, the Dow Theory [3] suggests that short-term stock movements can be predicted through technical analysis by identifying trends and patterns in historical data.

Building on these foundational theories, numerous efforts have been made to predict stock market behaviour in an attempt to achieve superior returns. With the advent of machine learning technologies in recent years, a variety of models have been developed using diverse algorithms. The efficacy of these prediction methods will be analysed in Section 1.2.

1.2 Literature Review

Recent advancements in machine learning and natural language processing have significantly influenced stock market prediction methodologies. Various studies have explored different architectures and techniques, each contributing to the evolving landscape of financial forecasting.

Selvin et al. [4] examined the effectiveness of several deep learning models, including the Convolutional Neural Network (CNN) with a sliding window approach, Recurrent Neural Network (RNN), and Long Short-Term Memory (LSTM) networks. Their findings indicated that the CNN architecture outperformed the other models in predicting stock prices based on real-time data. This superiority was attributed to its ability to effectively capture abrupt fluctuations in stock market data, showcasing the model's robustness in volatile conditions.

In a related study, Mohan et al. [5] improved the accuracy of sentiment analysis in deep learning algorithms by compiling an extensive dataset spanning over five years, which included more than 265,000 news articles. Their approach emphasized the importance of a comprehensive dataset in enhancing model performance, suggesting that the quality and quantity of input data are critical factors in achieving reliable sentiment analysis outcomes.

Further advancing the integration of sentiment analysis and stock prediction, Patel et al. [6] proposed an innovative model that combined natural language processing techniques with LSTM architecture. By leveraging sentiment analysis to assess the relationship between news sentiments and stock market movements, their model outperformed several state-of-the-art algorithms, including ADABOOST and XGBOOST. The model achieved remarkable predictive accuracy, reducing the mean squared error to 0.062, thereby highlighting the effectiveness of combining NLP with deep learning for financial forecasting.

Lastly, Qing et al. [7] explored the application of Transformer models for stock price prediction. They enhanced the conventional transformer architecture by improving locality and mitigating redundant heads in the self-attention mechanism. This adaptation allowed their model to excel in capturing long-term dependencies within financial time series data, thus demonstrating its superiority over LSTM networks in this context.

Together, these studies illustrate the significant strides made in utilizing advanced machine learning techniques for stock market prediction. The evolving methodologies not only enhance predictive accuracy but also underscore the interplay between sentiment analysis and financial forecasting, providing a strong foundation for the development of more sophisticated models in this domain.

1.3 Motivation

Both the Random Walk Theory and the Dow Theory, as discussed in Section 1.1, emphasize the critical role of information in stock price prediction. Consequently, the ability to assimilate relevant information in a timely manner has become increasingly vital for investors. Existing financial websites and trading platforms strive to provide users with extensive information. Leading financial data providers, such as Bloomberg [8], HKEX [9], and Yahoo! Finance [10], offer real-time market data and comprehensive financial news, empowering users to make informed portfolio decisions. Similarly, popular trading platforms in Hong Kong, including WeBull [11], Futu [12], and Longbridge [13], present multiple panels displaying stock market statistics, trading strategies, portfolio rankings, news, forums, and bulletins. While these functionalities aim to furnish users with comprehensive information, they can overwhelm novice investors, making it difficult to navigate the platforms and extract valuable insights. This challenge is further compounded by the high costs associated with premium financial services; for instance, the annual subscription fee for Bloomberg Terminal is approximately 27,660 USD [14].

In response to these challenges, our objective is to implement an intuitive platform that assists novice investors in forecasting stock market trends while enhancing the user experience. We aim to simplify the dashboard interface and streamline the information-gathering process by constructing a machine learning model that synthesizes data into a single, comprehensive evaluation of each stock's performance. To analyze textual information effectively, the model will employ natural language processing techniques to conduct sentiment analysis. Building on the performance of existing models discussed in **Section 1.2**, we will compare different model’s performance and try to combine LSTM networks and Transformer architectures to develop a robust predictive model.

Additionally, to enhance user experience, we intend to translate the evaluation results into personalized investment recommendations. By integrating a paper trading system into the website, beginners will have the opportunity to practice trading without incurring real financial losses. Concurrently, user behavior patterns will be recorded to analyze individual risk preferences, allowing the platform to provide tailored investment decisions that align with users' unique profiles.

1.4 Scope

As the final year project's scope is to develop a fully functional website integrated with the model within an eight-month timeframe, this interim report specifically covers the progress made during the first semester, with a primary focus on design and development of the NLP models, comparison between models, and the website prototype development.

1.5 Report Organization

This report is organized as follows: Chapter 2 outlines the project objective. Chapter 3 presents the general methodology for collecting data, building model, and constructing the overall AI-embedded web application. Chapter 4 introduces our preliminary results, including model construction and website development. Chapter 5 presents our future plan and schedule. Chapter 6 concludes the report.

2 Project Objective

This section outlines the purpose and goals of the AI-driven financial platform, emphasizing its focus on assisting novice investors. The intermediate objectives detail the key steps, including data collection, NLP model training, user interface design, and the development of a paper trading system. The ultimate goal highlights the deployment of a fully functional platform equipped with sentiment analysis, real-time market data integration, and personalized investment recommendations. Finally, the Platform Infrastructure section covers the development of frontend and backend systems, user trials, and iterative improvements to optimize the platform's functionality and user experience.

This project aims to develop an AI-driven, comprehensive financial management platform that integrates a paper trading system and a sophisticated Natural Language Processing (NLP) model for financial market forecasting. The primary objective is to assist novice investors by offering personalized investment recommendations and enhancing their decision-making capabilities.

The intermediate objectives of the project involve collecting and preprocessing data to train the NLP model effectively, extracting key features, and optimizing the model through rigorous training and testing to ensure high performance and reliability. Additionally, the project focuses on designing and prototyping a user-friendly interface that emphasizes usability and functionality. Another significant milestone is the development of a robust paper trading system equipped with risk control mechanisms to simulate real-world trading scenarios.

The ultimate goal is to deploy a fully functional, user-friendly financial website that incorporates a trained NLP model capable of performing sentiment analysis to forecast market trends and stock price movements. The platform will analyze real-time market data to provide users with timely notifications of trends and investment opportunities. A personalized recommendation system will utilize the paper trading system to collect user-specific data, enabling tailored forecasts and investment strategies based on individual risk profiles and trading preferences.

To ensure seamless functionality, the platform will feature a fully developed frontend and backend infrastructure. User trials will be conducted to gather feedback, which will guide iterative improvements to optimize the overall user experience.

3 Methodology

In this chapter, Section 3.1 introduces the different models to be developed in the project in the modelling process. Section 3.2 elaborates on the machine learning model architecture. Section 3.3 illustrates the design of the personalized investing recommendation system. Section 3.4 presents the implementation of the paper trading system. Section 3.5 introduces the methods for constructing the frontend and backend of the website.

3.1 Modelling Process

To address the diverse needs of users, a variety of predictive models have been designed and implemented, each tailored to specific investment objectives. The modelling process has been structured into three core components:

1. Predicting stock price direction using the LSTM model: this model addresses the binary classification problem of stock price, which determines whether a stock's price will rise or fall for a daily forecast horizon. This approach simplifies the decision-making process for investors by displaying only two potential outcomes.

2. Forecasting stock prices in numerical values using the LSTM model: this model provides users with detailed insights into market trends and the development of quantitative investment strategies.

3. Integration of multidimensional data through a combined LSTM-BERT model: this model incorporates market news and other external factors to enhance prediction accuracy and to better simulate the complexities of real-world financial environments.

Each of these approaches has been systematically developed to cater to different user requirements. For example, the classification model focuses on clear directional predictions, while the regression model emphasizes precise price forecasting for long-term strategies. The integrated regression model leverages additional data sources, such as market sentiment, to provide a holistic view of the financial landscape. By training and testing multiple LSTM models on the same dataset, we can compare their performance and refine our final model selection.

3.2 Machine Learning Models

This section presents the methodology utilized in the development of the three core machine learning models that are essential to the project. The first model emphasizes directional forecasting through LSTM networks, offering investors vital insights into expected stock price movements. The second model predicts the exact numerical future stock prices, providing a more comprehensive understanding of potential market dynamics. Lastly, the third model enhances the predictive capabilities of the second by integrating sentiment analysis derived from financial news, which enriches the input data for the LSTM framework. The methodologies for each model will be elaborated upon in the following subsections, detailing the specific techniques employed to ensure accurate predictions. The models for both predicting the price movement direction and precise stock price have been implemented and the preliminary results are explained in Section 4.

3.2.1 Stock Price Direction Forecasting Model

This section presents a thorough description of the dataset used, the data preprocessing techniques implemented, and the approach taken to assign labels to the data points for the stock price direction model. The first model addresses the binary classification problem of various stocks' price movement direction for the daily forecast horizon. This model, which focuses on directional forecasting utilizing LSTM, serves as a fundamental basis for the subsequent models. For novice investors, comprehending the anticipated direction of stock price movements—whether an increase or decrease—is indispensable for making investment decisions prior to receiving an exact price prediction. This initial insight enables users to undertake strategic actions in response to the projected market trends. Consequently, a clear understanding of directional movement enhances the significance of the precise price predictions offered by the second model, while the third model further refines this process by incorporating sentiment analysis, thereby enhancing investors' capabilities to make well-informed decisions in a dynamic market environment.

3.2.1.1 Data Collection and Train-Test Splitting

The stock price movement prediction model is utilized to forecast future price movements of the components within the Nasdaq stock market index. For the purpose of analysis, only companies listed as components of the Nasdaq market index with top market capitalization are considered.

The dataset is sourced from the publicly accessible Yahoo! Finance application programming interface (API) in Python with each data point representing a single trading day. These data points comprise daily open, close, high, and low prices along with the corresponding trading volume for that day. The closing price for each trading day was adjusted for various firm actions, for instance, the stock splits and the payment of dividends, to ensure the high performance accuracy of each stock.

The dataset is further divided into a training set and a testing set. The training set spans 6,288 trading days, covering the period from January 3, 2000, to December 30, 2024. The testing set includes 4,376 trading days. After splitting the dataset, around 70% of the data is used for training the classification model, and around 30% of the data is used for testing the robustness of the model. Therefore, the size ratio between the testing and training datasets is approximately 3:7.

3.2.1.2 Data Preprocessing: Heikin-Ashi Candlestick Transformation

Following the data collection process, the subsequent critical step involves transforming the stock price data into Heikin-Ashi candlesticks. This transformation serves several key purposes in forecasting future stock price movements. Heikin-Ashi candlesticks are specifically designed to smooth out price fluctuations, thereby facilitating the identification of trends by reducing market noise and enabling a clearer focus on the overall direction of the market. Additionally, this method contributes to improved trading signals by minimizing the occurrence of false signals often associated with traditional candlestick patterns.

To convert traditional stock price data into Heikin-Ashi candlesticks, specific calculations are applied for each trading day. The Heikin-Ashi closing price is computed as the average of the open, close, high, and low prices. The Heikin-Ashi opening price is determined by averaging the Heikin-Ashi Open and Close values from the previous period. The Heikin-Ashi High is calculated as the maximum value among the current high, Heikin-Ashi Open, and Heikin-Ashi Close, while the Heikin-Ashi Low is obtained as the minimum value among the current low, Heikin-Ashi Open, and Heikin-Ashi Close. By performing these calculations on price data, the dataset is transformed into Heikin-Ashi candlestick data, providing a more reliable foundation for subsequent analysis and modeling aimed at predicting future stock price movements.

3.2.1.3 Data Preprocessing: Feature Engineering

To improve the predictive performance of the directional forecasting model, technical indicators derived from the transformed Heikin-Ashi candlestick data have been incorporated as input features. The use of these indicators offers several benefits, as they condense complex and noisy information regarding price, volume, and the price momentum into metrics that are straightforward to interpret. By capturing key aspects of market behavior such as trends, volatility, momentum, and trading volume, technical indicators play a crucial role in identifying patterns and predicting future price movements. The inclusion of technical indicators allows the model to achieve a more comprehensive understanding of market dynamics.

A total of eight technical indicators are selected and categorized based on their specific roles. Among the trend-based indicators, the Exponential Moving Average (EMA) highlights the direction of a trend by smoothing price data; the Average Directional Movement Index (ADX) measures the strength of a trend; the Moving Average Convergence Divergence (MACD) detects shifts in momentum and trend direction; and the Commodity Channel Index (CCI) identifies overbought or oversold conditions by evaluating the deviation of price from its average.

For volatility measurement, the Average True Range (ATR) is used to quantify market volatility by analyzing price movement ranges. Within the momentum indicator category, the Relative Strength Index (RSI) assesses the speed and magnitude of price changes, while the William’s Percent Range determines overbought or oversold conditions of the stock concerned. Finally, the volume-based indicator, A/D Oscillator, examines the relationship between price and volume to identify accumulation or distribution phases in the market.

Incorporating these technical indicators as input features equips the model with the ability to capture a broad spectrum of market dynamics, thereby creating a more robust and reliable framework for forecasting future stock price movements.

3.2.1.4 Data Preprocessing: Feature Standardization

Feature standardization is performed to ensure that all input features have a zero mean and unit variance, which is essential for optimizing the model performance and stability of the predictive model. Standardization eliminates biases caused by differences in the scales of the input features. From the perspective of model training, it accelerates the convergence of optimization algorithms and prevents features with larger numerical ranges from dominating those with smaller ranges. As the LSTM machine learning models with the gradient-based optimization algorithms are sensitive to the scale of input data, standardization ensures that all features contribute equally to the learning process.

3.2.1.5 Data Labelling

For labeling the response variable, each valid data point corresponding to a single trading day is given a unique label based on the behavior of the adjusted closing price. A label of 'UP' is assigned to a data point when the closing stock price increases, while a label of 'DOWN' is assigned when the closing stock price decreases. The specific criteria for label assignment are as follows:

3.2.1.6 Model Architecture

The Long Short-Term Memory (LSTM) model is particularly beneficial for predicting the direction of stock price movements due to its ability to capture long-range dependencies within sequential time series data, for instance, the financial stock data. Unlike conventional recurrent neural networks (RNNs), LSTMs are specifically designed to retain information over extended periods, making them particularly suitable for time series analysis, where historical data plays a significant role in determining future possible outcomes. This capability enables LSTMs to identify complex and relatively noisy patterns and trends in stock prices over time. Given the nonlinear behaviors and fluctuations commonly observed in the financial markets driven by a wide range of factors, the architecture of LSTMs is adept at managing these complexities.

To further enhance the performance of the model and mitigate the risk of overfitting, a dropout layer is integrated into the architecture using Python Keras, which aids in improving generalization by compelling the model to learn robust features that are not dependent on any single input. This regularization technique is essential for preserving the integrity of the model when it encounters unseen data.

Additionally, the binary cross-entropy loss function is utilized to minimize loss during the training phase. This loss function is particularly effective for binary classification tasks, as it measures the discrepancy between the predicted probabilities and the actual labels. By optimizing this loss function, the model can effectively refine its predictions regarding whether stock prices will increase or decrease, thus enhancing its reliability and accuracy in forecasting stock price movements.

3.2.2 Numerical Stock Price Prediction Model

This section outlines the methodologies used for the stock price prediction model by using pure LSTM architecture.

3.2.2.1 Data Processing

The model employs two critical methods in feature engineering to optimize its performance. First, Feature Normalization was applied using MinMaxScaler to scale all features within the range of [0, 1]. This ensures faster convergence during training and prevents any single feature from dominating predictions due to differences in magnitude.

Second, the Sliding Window Method was utilized, wherein a 14-day sliding window was designed to capture temporal dependencies in the data. This approach uses the past 14 days of data to predict stock prices for the next 1, 7, and 15 days, effectively modeling the sequential nature of financial data.

3.2.2.2 Model Architecture

The architecture of the second model includes two main layers: a Dropout Layer, which mitigates overfitting by randomly disabling neurons during training, and a Dense Layer, which maps the processed data to the target stock price. Hyperparameters were carefully chosen, including a 14-day window, Mean Squared Error (MSE) as the loss function, and the Adam optimizer for efficient and reliable optimization.

During training, the dataset was split into training and testing sets in a 7:3 ratio. To ensure robustness, early stopping was implemented, automatically halting training when validation loss ceased to decrease. Additionally, dynamic learning rate adjustment was used to reduce the learning rate whenever validation loss plateaued, further stabilizing the training process. The model's performance was evaluated using three standard metrics: Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and R^2. Line and scatter plots were employed to compare predicted values with actual values, providing a clear visual representation of the model's effectiveness.

3.2.3 Integrated Stock Price Prediction Model with Natural Language Processing Techniques

To enhance the comprehensiveness of stock price predictions, a pre-trained distilbert-base-uncased model was introduced for sentiment analysis of financial news. Each financial article was assigned a sentiment score, where +1 indicated positive sentiment and -1 indicated negative sentiment. These scores were weighted by the model's confidence to produce a final sentiment score for each article. This sentiment score was then integrated as a new feature, offering additional context for the predictive model.

The integration process involved two key steps. First, time alignment ensured that sentiment scores were matched with corresponding stock price data using date and creation time fields. The problem of missing sentiment data for specific time points was addressed using linear interpolation. Second, feature combination expanded the feature set by merging sentiment scores with traditional stock features such as opening price, closing price, and trading volume. This augmentation enabled the model to simultaneously capture price trends and market sentiment fluctuations, enriching the input data for prediction.

3.2.3.1 Data Collection and Preprocessing

Our objective is to gather text data from diverse financial market sources, including financial news (CNBC), financial reports (Apple Inc.), social media information (X.com), and user comments (Yahoo et al.). This data will cover real-time stock market data, corporate financial data, news articles, financial reports, a small amount of macroeconomic data, and social media data, providing a comprehensive view of the financial landscape.

Our data collection strategy is versatile, employing different tools based on the nature of the data. For static pages, we'll utilize Scrapy and Beautiful Soup for web scraping, while dynamic data will be captured using Selenium. Additionally, we'll leverage APIs from financial platforms such as Yahoo Finance, Alpha Vantage, and CNBC to ensure we gather the most relevant and up-to-date data.

Regarding data volume, we plan to collect approximately 1.2 million valid data points to ensure that complex models can capture meaningful patterns. The data span should cover various market conditions, such as bull, bear, and volatile markets, and will span approximately 10 years to take different economic cycles into account.

The collected data will be formatted into JSON or CSV files for preprocessing. Further, the data will be cleaned by removing noise, stopping words, and normalizing the text (e.g., converting it to lowercase). TensorFlow will be used for tokenization, with particular attention paid to terms related to financial domains. Stemming and lemmatization will reduce word diversity, and low-frequency and high-frequency words will be filtered out to improve model efficiency. The text will be transformed into vector representations using Bag-of-Words, TF-IDF, or word embedding techniques. Finally, the dataset will split into training, validation, and test sets, and, if necessary, data augmentation techniques will be applied to generate additional training samples.

3.2.3.2 Natural Language Processing Model Architecture

To help users make informed investment decisions, we will employ NLP technology to analyse news articles and social media trends. This project will integrate multiple deep-learning models to process large amounts of textual data with time series as an integrated model. The integrated model (or cascade model) provides more accurate predictive outcomes in financial market sentiment analysis by connecting multiple deep learning models sequentially, with the output of one model serving as the input to the next, to take full advantage of the strengths of each model in different tasks [15].

In the integrated model, long short-term memory network (LSTM) will first be used to capture text-dependent dependencies. LSTM model can effectively deal with the dynamic characteristics of time series data, such as the temporal fluidity of financial news and the changing trend of social media emotions [16]. LSTM processes long-term and short-term information through its built-in "memory unit" and is particularly suited to capturing the short-term impact of financial events, such as breaking market news or policy announcements. The output of the LSTM is passed as input to the Transformer model, which captures global context information in the text through its self-attention mechanism and can identify long-term dependencies and complex semantic associations [17], such as the potential link between long-term stock fluctuations and market sentiment.

In practice, we will implement this cascade architecture using Python through training the LSTM and Transformer models in series and optimizing the weights of the entire model through backpropagation.

3.2.3.3 Enhanced LSTM-BERT Model

The enhanced LSTM-BERT model utilized the augmented feature set as input. The architecture comprised three layers:

1. **Input Layer:** This layer accepted time-series data generated by the sliding window method, combining historical price features with sentiment scores. This integration allowed the model to capture both temporal patterns and sentiment-driven influences on stock prices.

2. **LSTM Layer:** This layer focused on learning dependencies within the time-series data, identifying complex relationships between price trends and sentiment data.

3. **Output Layer:** The final layer generated stock price predictions for specified future intervals, offering actionable insights for investors.

3.3 Personalized Investment Recommendation System

After successfully training the aforementioned models, the following steps will be followed to integrate the model into the application to achieve personalized investment recommendations:

3.3.1 Data Input and User Preference Collection

In order to output user-specific recommendations, we will make the collection of user's investment style, risk appetite, financial goals, assets, and other relevant data. In the application's user interface design, a particular module will be set up for the user to fill in the aforementioned information, and it could also be analysed through the user's historical behaviour and transaction data.

3.3.2 Integration of the NLP Model

This stage embeds the trained NLP model into the back-end system. The model's primary function is to analyse market sentiment and related news to extract meaningful information that may affect market movements. This information will be used to generate investment recommendations.

The NLP model can communicate with the front end through an application programming interface (API) in the system architecture. When a user requests an investment advice, the front end passes user and real-time market data to the NLP model. After the calculation of overall sentiment score with the data and sentiment quantification by the NLP model, rules for investment suggestions are set according to the range of sentiment scores and the user's style data, for example:

$$sentiment > 0.6 : \text{Buy}$$

$$0.3 < mood\ score \leq 0.6 : \text{Hold}$$

$$sentiment\ score \leq 0.3 : \text{Sell}$$

3.3.3 Personalized Recommendation Algorithm

Use the model to generate portfolio recommendations appropriate for the user, combining the user's preferences and risk tolerance level. For example, if a user is a conservative investor, the model may prefer stocks or funds that are less volatile.

Potential market opportunities can also be added to recommendations based on market sentiment and news hotspots analysed by the NLP model. For example, when the model detects that sentiment in a particular sector or stock is more optimistic, it can recommend it to the user first.

3.3.4 Recommendation Result Display and User Interaction

In the application, we will design an intuitive user interface to display the recommended portfolio. As a result, the application can display the underlying reasons for the recommendation to users through charts and risk distributions to help them better understand the logic.

Moreover, users can choose whether to adopt the recommendations or fine-tune them, such as altering the proportion of certain assets to fit their risk tolerance level.

3.4 Paper Trading System

We will develop a paper trading system to enhance user experience and integrate a personalized investment recommendation system, allowing users to test investment advice and different trading strategies or positions in a simulated and virtual environment. This system provides virtual funds to execute simulated buy/sell operations based on recommendations generated by the developed models and real-time stock market data. All transaction data will be stored in a MySQL database, with real-time data ensuring market-based decisions. A trading engine will update users' current positions and account balances according to the users' stock portfolios. Additionally, users can assess the performance of recommendations through trade analysis and adjust the actual trading strategies based on the risk and return rates.

The system will incorporate two core risk control mechanisms: stop-loss/take-profit and position management. The stop-loss/take-profit mechanism automatically closes trades when preset thresholds are reached, calculated using Pandas and NumPy. Position management will allocate capital for each trade based on total account funds and user risk tolerance, using methods like the fixed percentage method or Kelly criterion. The system will dynamically monitor market changes by integrating Python trading APIs (e.g., Alpaca, Interactive Brokers), helping users practice risk management skills and optimize investment strategies.

3.5 Front-end and Back-end Construction

This section discusses the development of the application's user interface of the main web-based deliverable using React for frontend features, as well as the backend system built with Django and MySQL to manage data and facilitate communication between the client and server.

3.5.1 Frontend Development

The user interface of this application displays personalized investment recommendations, the user's transaction history, and real-time stock market data. Key stock trends, forecast values, and the latest financial news headlines are presented in chart form, highlighting the most influential positive or negative news from sentiment analysis. Apart from numerical forecast values to be displayed, predicted binary stock price direction from the machine-learning-based model is presented for investors to make prompt trading decisions. The interface will be developed using React, which, with its powerful component-based structure, provides a clean and smooth user experience.

3.5.2 Backend Development

The backend will be built using the Django framework, responsible for handling frontend requests and managing backend tasks. Django utilizes the Django Rest Framework (DRF) to implement RESTful APIs, facilitating communication with the frontend. The server is primarily responsible for retrieving data from the MySQL database, performing calculations, and returning personalized recommendation results. MySQL, as a relational database, is used to store key data, including user transaction records, market data, and NLP model inputs and outputs. With the tight integration of Django and MySQL, the system ensures data processing stability and performance.

4 Accomplished Part

In this chapter, Section 4.1 presents performance of stock price classification model. Section 4.2 elaborates on data collection and preprocessing. Following, section 4.3 highlights the integration of sentiment analysis using BERT. Section 4.4 presents web development of project.

4.1 Performance of Stock Price Movement Classification Model

After successfully training the machine learning classification model to predict the direction of Apple stock price movements, an evaluation was conducted using a confusion matrix. This matrix provides insights into the model's performance by including the true positive, true negative, false positive, and false negative classifications. From this analysis, the accuracy rate was computed as the ratio of correctly predicted movements to the total number of testing data points. The accuracy rate was be calculated as follows,

$$Accuracy = \frac{TrueUp + TrueDown}{N_{test}} \approx 0.57.$$

As presented in **Figure 1**, the model achieved an accuracy rate of approximately 57%. While this figure may initially seem modest, it is essential to contextualize it within the complexities of financial markets. In environments characterized by significant noise and volatility, achieving higher accuracy can be particularly challenging. Market fluctuations are influenced by a multitude of unpredictable factors, including economic indicators, company earnings reports, and broader geopolitical events. Consequently, an accuracy of 57% can be considered moderate and indicative of the model's ability to discern patterns within a tumultuous landscape. This performance underscores the inherent difficulties in predicting stock price movements, highlighting the need for future continuous refinement of the model and exploration of additional features or alternative methodologies to improve predictive accuracy.

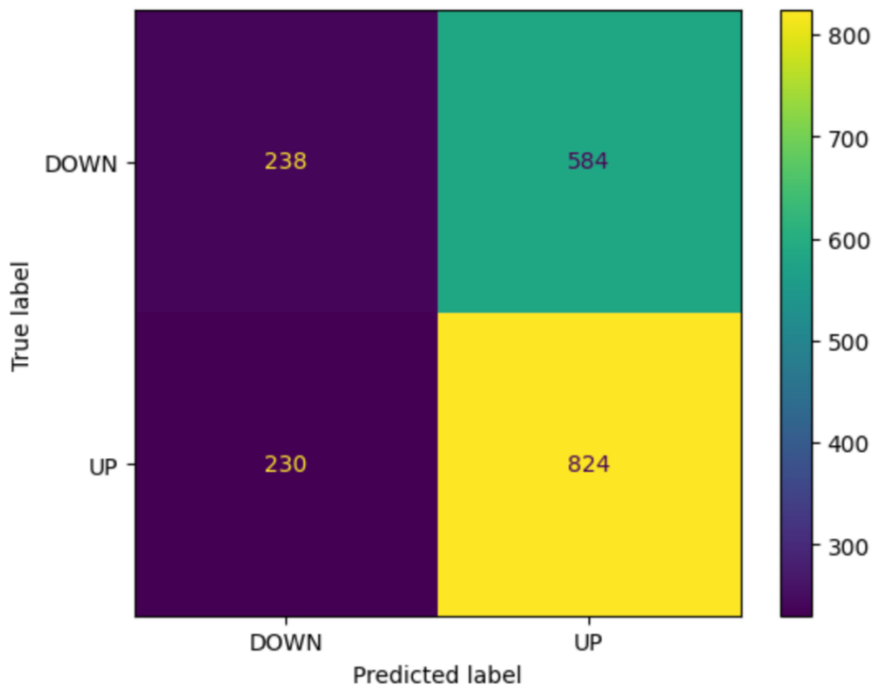

Figure 1. Confusion Matrix from the LSTM binary classification model for AAPL stock price movement prediction

4.2 Data Collection and Preprocessing

The data collection and preprocessing process has been a critical step in model development. Stock price data, serving as the primary input, was gathered using methodologies similar to earlier approaches. Complementing this, news data was collected from 2020 to 2024, including headlines and summaries of articles related to specific stocks. Text cleaning techniques, such as tokenization, were applied to extract key topics, and word clouds were generated to visualize the most frequently occurring keywords. For instance, keywords like "Microsoft" and "Nasdaq" prominently appeared, highlighting the core themes in the dataset. Filtering techniques were employed to ensure even distribution of news data across the timeline, maintaining consistent data quality.

The use of word clouds has been instrumental in visualizing major topics and enriching the dataset, forming a solid foundation for capturing market sentiment. These enriched datasets, combined with the advanced predictive capabilities of LSTM and BERT models, provide a comprehensive framework for accurate stock price forecasting and market analysis.

4.3 Integration of Sentiment Analysis Using BERT

The combined LSTM-BERT model demonstrated superior predictive capabilities by leveraging the strengths of both temporal modelling and sentiment analysis. These advancements have laid the groundwork for further optimization and testing, ensuring the models meet the demands of real-world financial forecasting.

The LSTM-BERT integrated model underwent the same rigorous testing and evaluation procedures as the standalone LSTM model. A comparative analysis of their preliminary results was conducted to assess their effectiveness in predicting stock prices over 7-day and 15-day timeframes.

Given the significant differences in trends and fluctuations among various Nasdaq stock prices, a stock-specific training approach was adopted. Instead of aggregating all stock data for training, individual models were tailored to specific stocks, resulting in varied performance outcomes across different datasets. This approach ensured that each model was optimized for the unique characteristics of individual stocks.

The 7-day prediction results highlighted a stark contrast between the two models. Scatter plots illustrated that the LSTM model’s predictions deviated significantly from actual values, particularly in higher value ranges (see Figure 2). The scatter points were noticeably distant from the ideal diagonal line, underscoring the model's limited fitting ability in capturing short-term price fluctuations. Conversely, the LSTM-BERT model demonstrated superior performance, with predictions closely aligning with the diagonal line (see Figure 3). Outliers exhibited smaller deviations, indicating that the integration of BERT enhanced the model’s capacity to capture time-series patterns and improve short-term prediction accuracy.

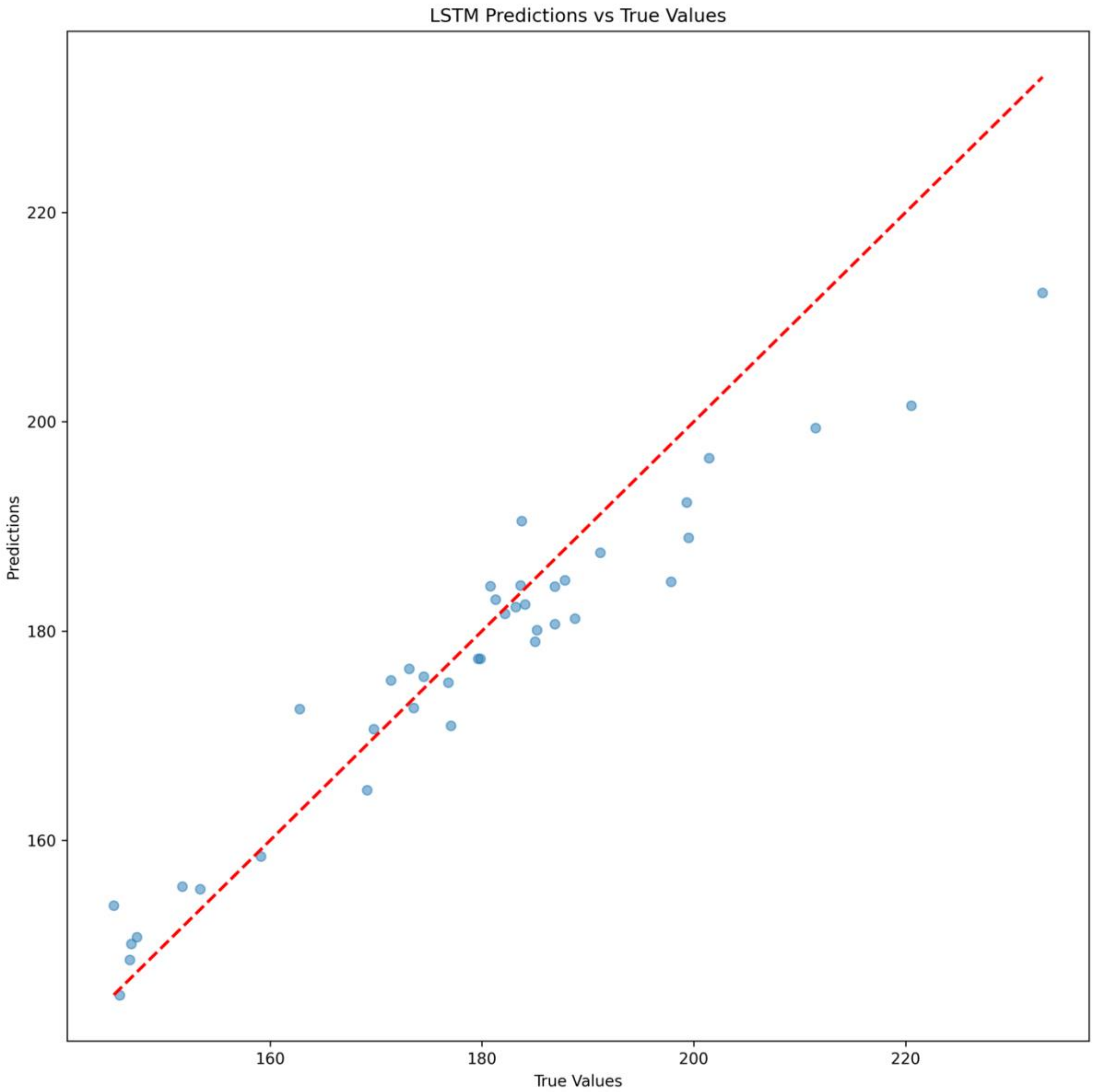

Figure 2. AMAZON LSTM Scatter Diagram (7 days)

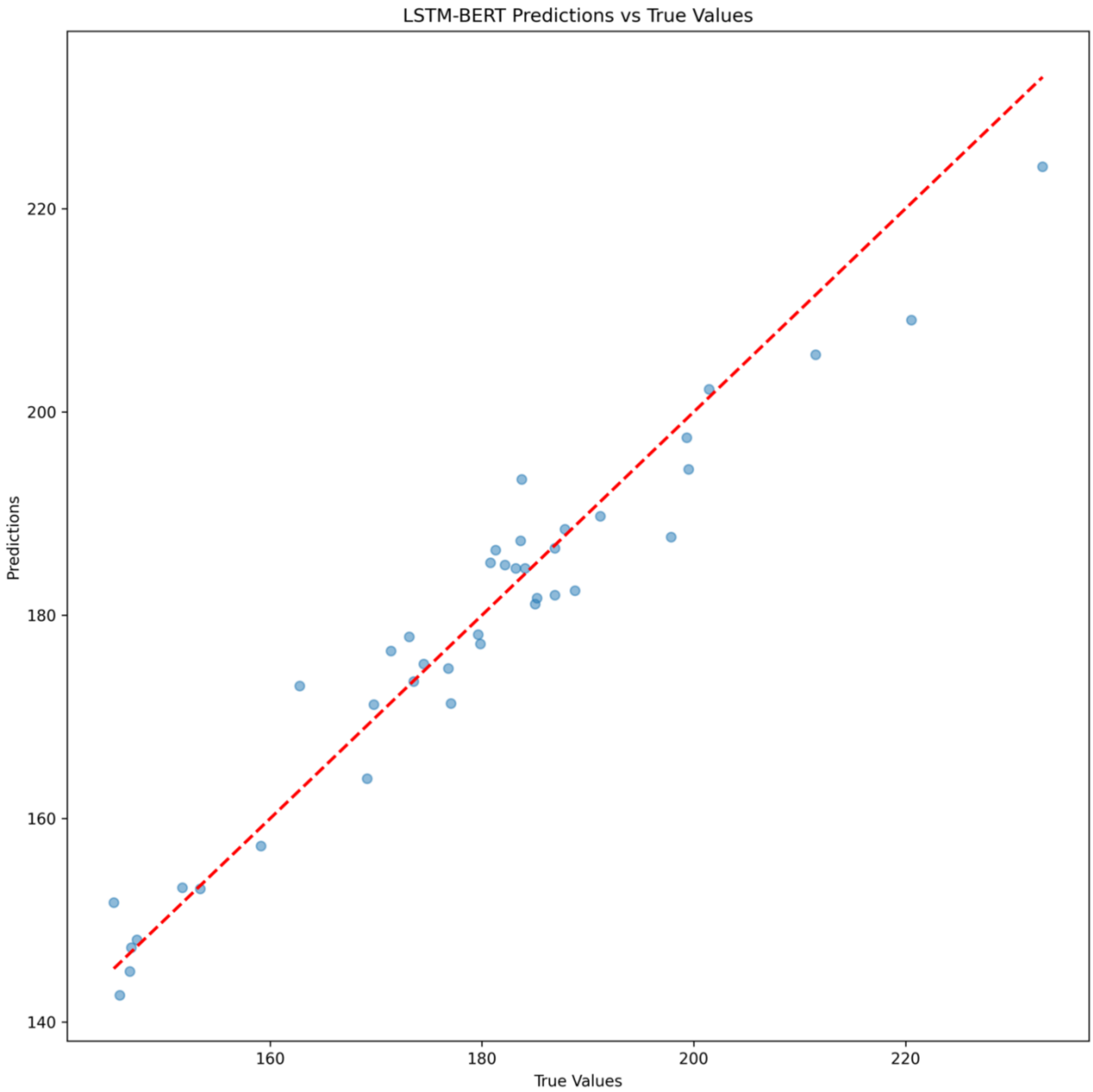

Figure 3. AMAZON LSTM-BERT Scatter Diagram (7 days)

For the 15-day prediction results, the performance gap between the two models became even more pronounced. The LSTM model's predictive accuracy declined significantly, with scatter points in lower value ranges deviating substantially from the diagonal line (see Figure 4). This revealed the LSTM model's challenges in handling long-term predictions. In contrast, the LSTM-BERT model exhibited greater robustness and stability. Although its performance was also affected over the extended prediction window, the scatter points remained relatively close to the diagonal line, with smaller deviations compared to the LSTM model (see Figure 5). This demonstrated the LSTM-BERT model's ability to maintain stability and accuracy even in long-term prediction tasks.

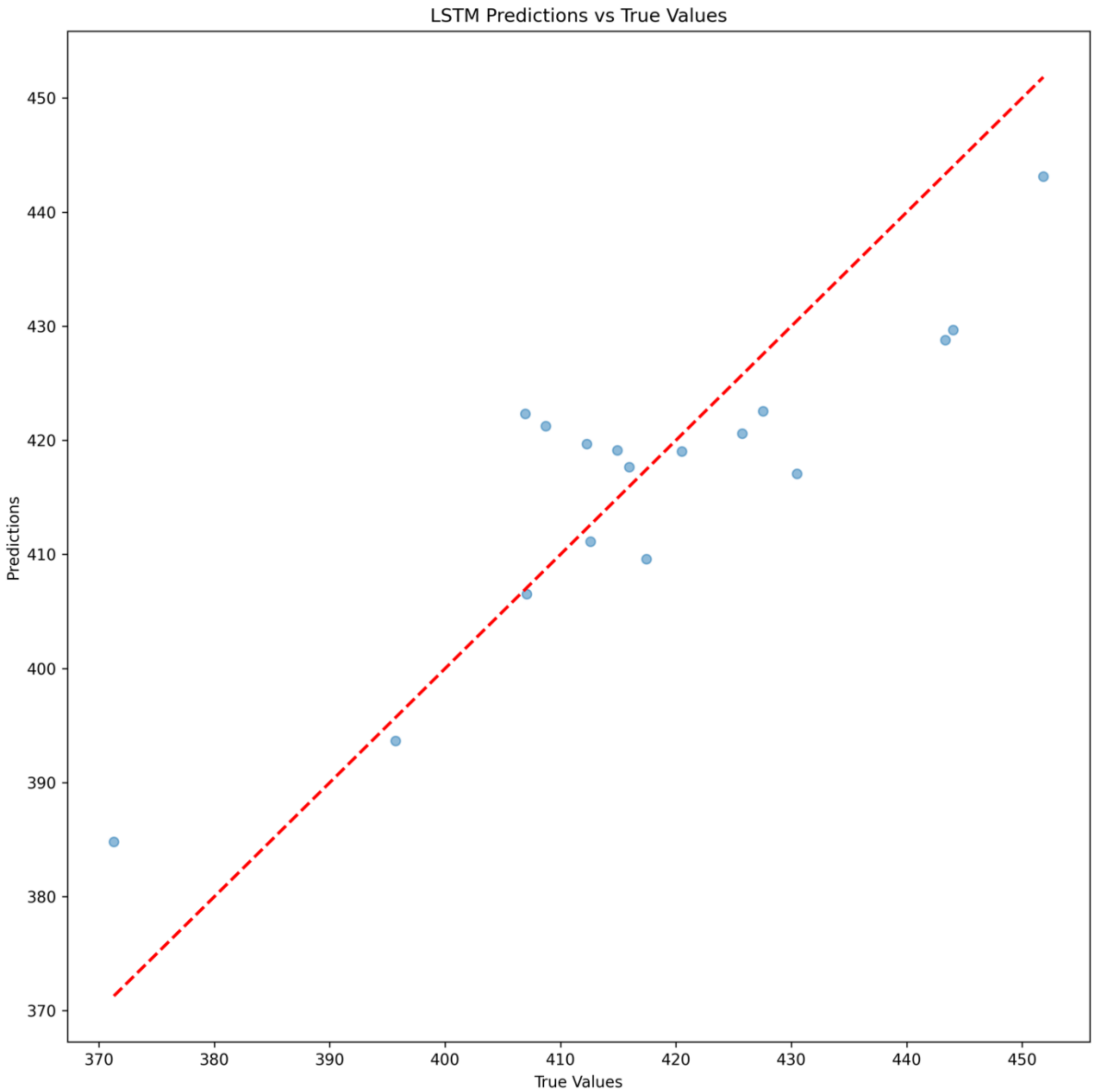

Figure 4. MICROSOFT LSTM Scatter Diagram (15 days)

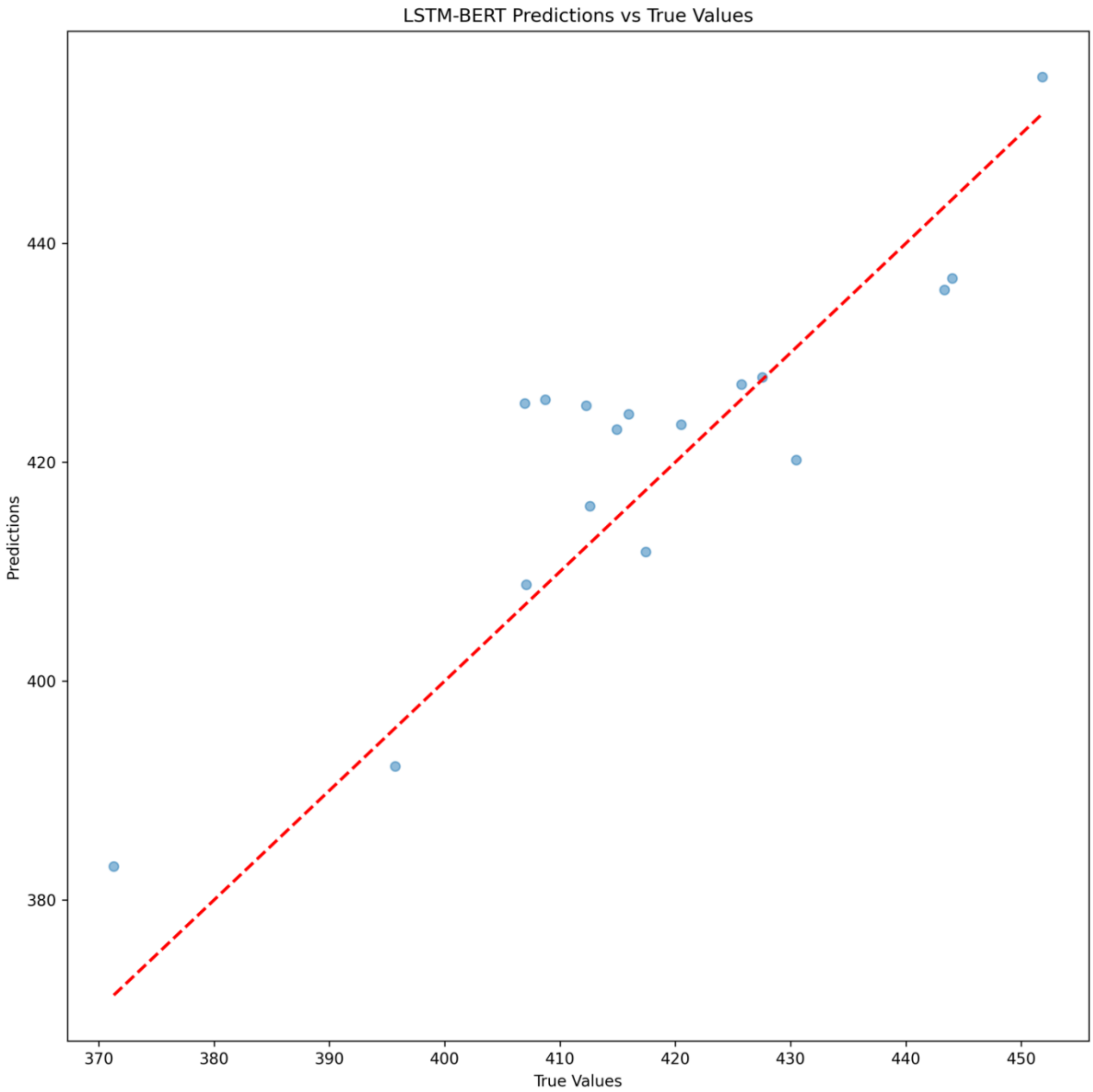

Figure 5. MICROSOFT LSTM-BERT Scatter Diagram (15 days)

To provide a more intuitive comparison, four performance metrics were analyzed and visualized in bar charts: the coefficient of determination (R^2), root mean square error (RMSE), mean absolute percentage error (MAPE), and mean absolute error (MAE). The LSTM model was represented in green, while the LSTM-BERT model was depicted in blue.

- R^2 (Coefficient of Determination): The LSTM-BERT model consistently scored higher, reflecting better fitting ability in predicting stock prices over 7 days.
- Error Metrics (RMSE, MAPE, MAE): The LSTM-BERT model showed significantly lower values, indicating reduced prediction errors and lower sensitivity to outliers compared to the LSTM model.

For the 15-day predictions, the superiority of the LSTM-BERT model was even more evident (see Figure 6). While both models experienced a decline in performance over the extended timeframe, the LSTM-BERT model remained relatively stable. Its R^2 value dropped by only 6%, whereas the LSTM model's R^2 plummeted to less than half of its original value. Additionally, the error metrics for the LSTM model increased by over 170%, further emphasizing its limitations in long-term forecasting.

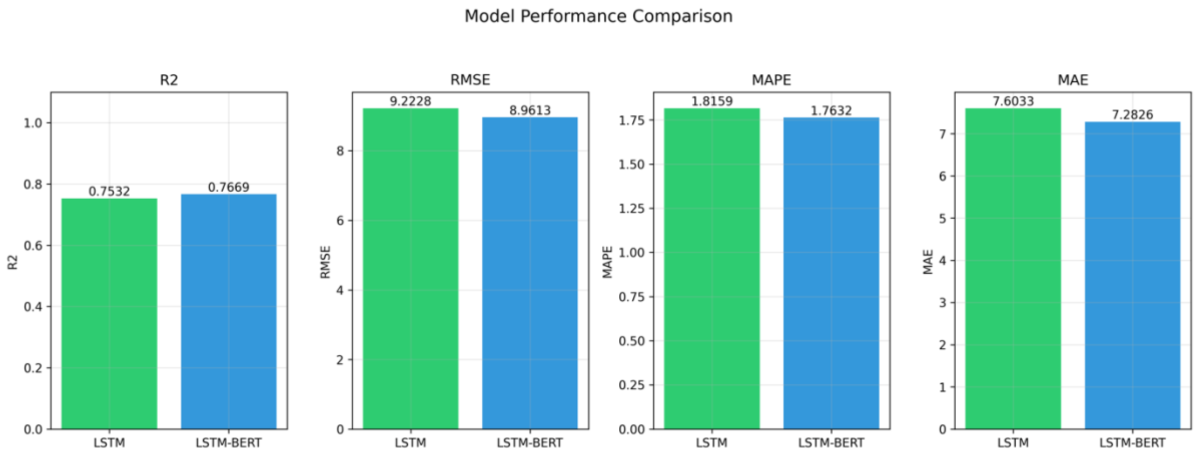

Figure 6. MICROSOFT Model Comparison (15 days)

4.4 Website Development

To construct a concise and user-friendly website, a clickable prototype has been developed using Figma. This prototype includes several key pages: the home page, stocks page, specific stock page, paper trading page, news page, and model page.

As illustrated in **Figure 7**, each page features a navigation bar at the top, enabling users to switch easily between sections. The home page provides a brief introduction to our website with a quick search box, facilitating users to find stock information quickly by directing to the searched stock page.

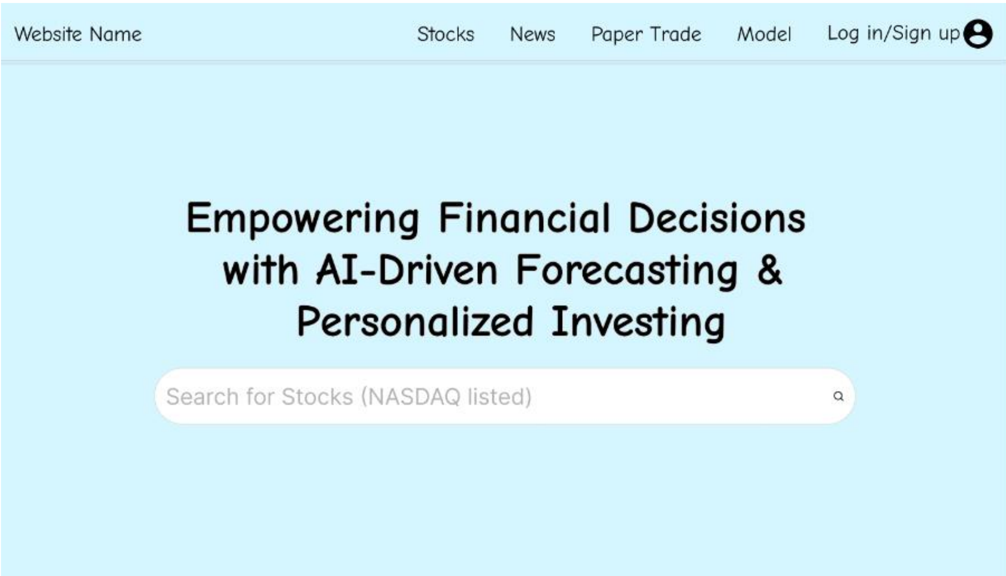

Figure 7. Home Page Prototype

The stocks page (see **Figure 8**) lists all stocks included in our model, currently focusing on NASDAQ Stocks. Users can filter stocks by price movement predictions and investment recommendations, and sort them alphabetically or by price. The chart displays essential stock data, such as price, change, and market capitalization, alongside predictions from our model to guide investors on possible movements. To maintain simplicity, only the predicted direction of price movement will be shown in different colours. The last column provides investment recommendations, tailored to individual risk preferences.

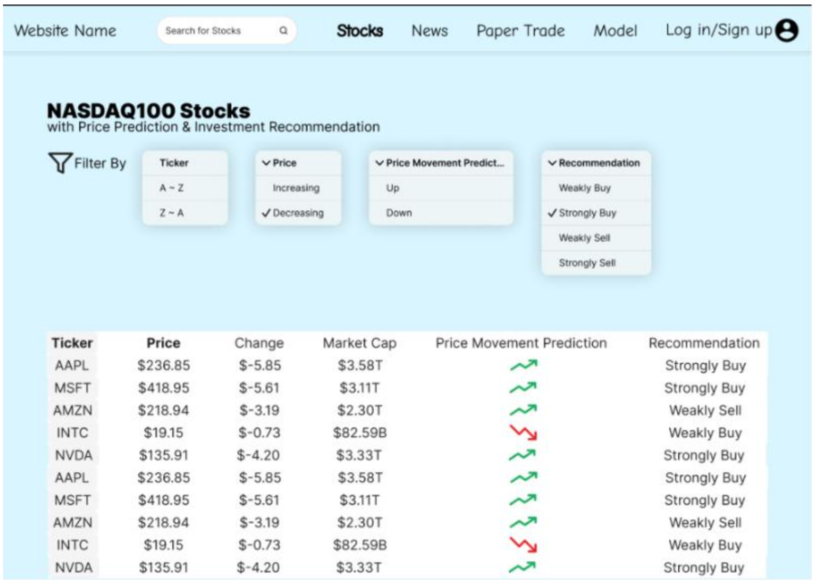

Figure 4.8. Stocks Page Prototype

By clicking on a ticker symbol, users will be directed to the specific stock page (see **Figure 9**). This page features a title displaying the ticker and company name, along with a stock chart showing price movements across various time frames. Below the chart, relevant stock data unique to our website will be displayed, including a news sentiment score, price movement prediction, and our recommendation. In the bottom right corner, users can choose to buy or sell the stock through our paper trading function.

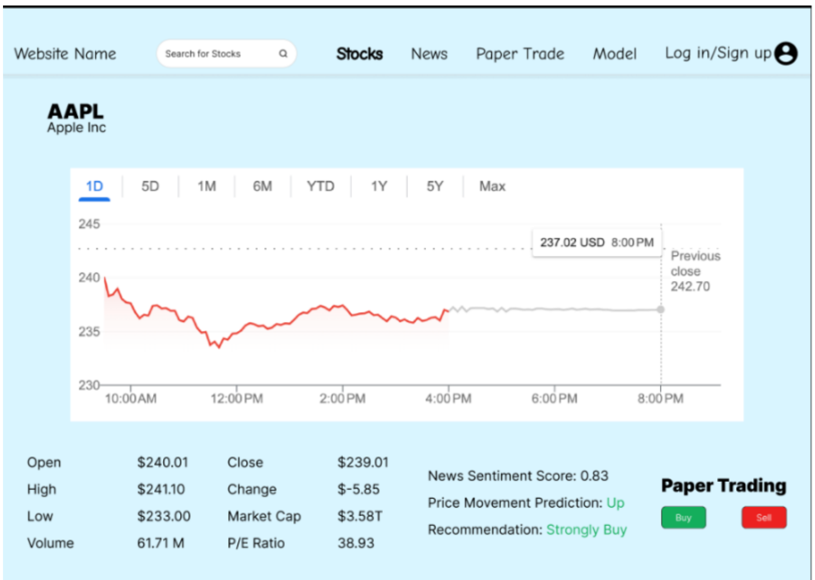

Figure 59. Specific Stock Page Prototype

On the paper trading page (see **Figure 10**), users can view their net assets and buying power, represented in virtual currency. Our system analyses each user's risk preferences based on their past trading behaviour, or users can manually set their risk preference. Below, users will find their current positions and transaction history, displaying all necessary information.

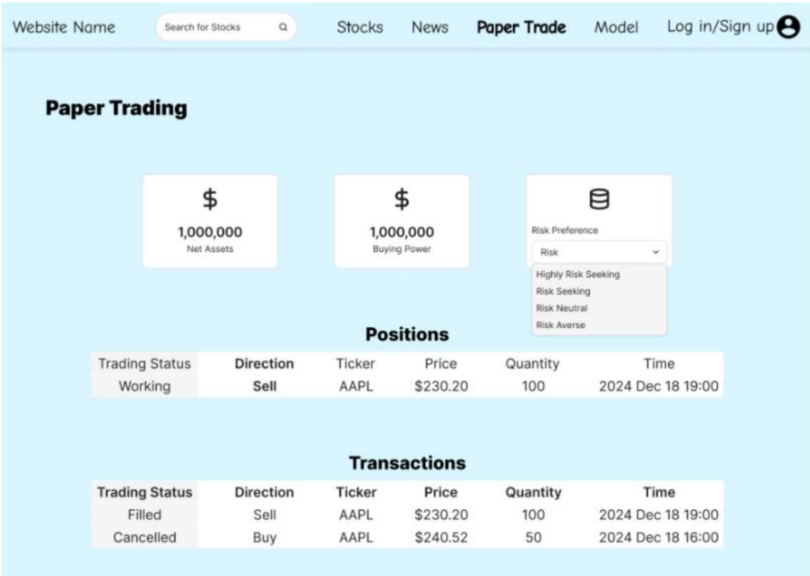

Figure 610. Paper Trading Page Prototype

Finally, the news page (see **Figure 11**) will feature selected news relevant to our model, providing timely information for each chosen stock. The model page (see **Figure 12**) will explain the principles behind our model, helping users better understand how to utilize our website for their personal investments.

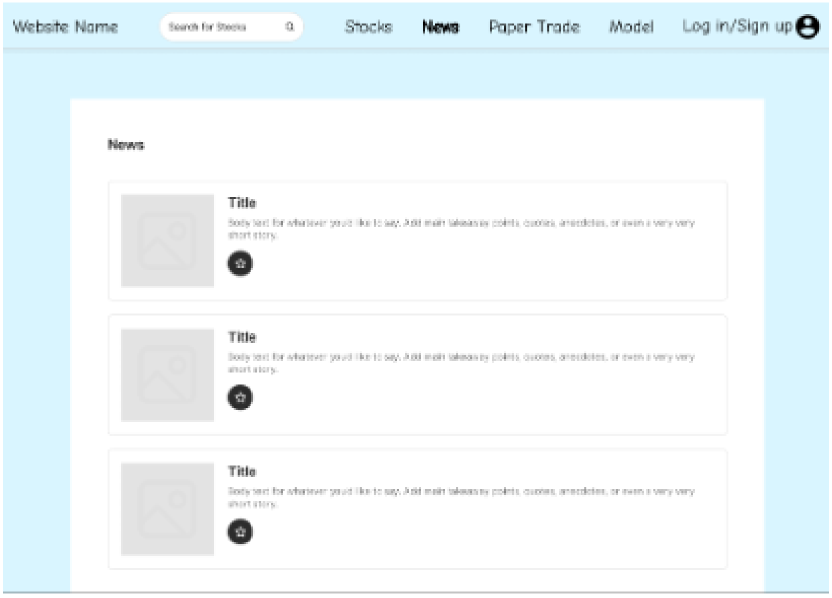

Figure 11. News Page Ptototype

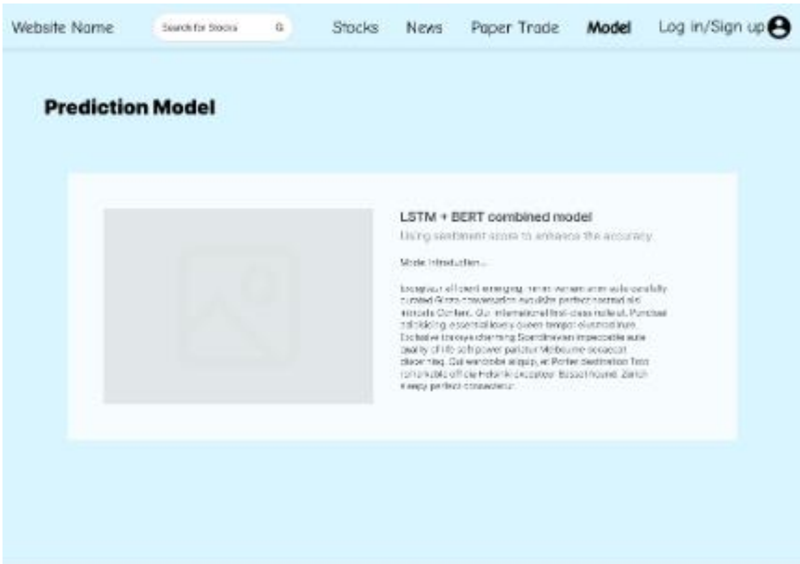

Figure 12. Model Page Prototype

5 Future Work

In the second semester of this academic year, the project will focus on advancing the development and optimization of predictive models, integrating additional statistical models for comparative analysis, and refining the application's functionality to enhance its real-world applicability.

5.1 Model Tuning and Optimization

The primary focus will be on the fine-tuning and optimization of the existing LSTM and LSTM-BERT models. Hyperparameter optimization will be carried out by using various common and useful techniques, for instance, the grid search and Bayesian optimization techniques, to improve the models' predictive accuracy and efficiency. Advanced performance metrics, such as the Sharpe Ratio and the Pearson's product-moment correlation coefficient between the predicted and training values, will be incorporated to evaluate model performance in terms of risk-adjusted returns, ensuring that the models are not only accurate but also aligned with the practical needs of investors.

Additionally, the training pipeline will be scaled to accommodate additional NASDAQ stocks, leveraging distributed computing techniques to enhance model performance and reduce training time. This will allow the models to generalize better across a broader set of stocks and provide more comprehensive market insights.

5.2 Comparison of Statistical Models

In response to the feedback from the professors, the study will expand to include the comparison of traditional statistical models for a valid comparison with the machine learning-based models. Models such as Linear Regression, Autoregressive Integrated Moving Average (ARIMA), and Random Forests will be compared and tested for their predictive capabilities on stock price movements. These models will serve as baselines for assessing the effectiveness of the deep learning approaches and provide insights into the strengths and weaknesses of both traditional and modern methods in financial forecasting.

5.3 Frontend and Simulation System Development

The second phase of the project will focus on the development of the frontend and simulation system. A responsive user interface will be created using React.js to ensure seamless interaction with the application. The interface will feature dynamic visualizations of market trends and predictions, providing users with an intuitive platform for analyzing stock data and model outputs.

Additionally, a backend trading engine will be developed to simulate real-time market scenarios using Flask and WebSocket APIs. This simulation system will allow users to test their investment strategies under various market conditions and assess the performance of different models. The integration of trading rules and user-defined constraints will further enhance the system's applicability to real-world trading environments. The set of user-defined trading preferences will be developed during the construction phase of the project.

5.4 Model Deployment and System Integration

In the construction phase of the project, the trained models will be embedded into the backend using TensorFlow Serving and FastAPI to ensure efficient and scalable deployment. RESTful APIs will be developed to connect the predictive models with the frontend, allowing users to interact with the models in real-time. Data pipelines will be secured with OAuth2 authentication to ensure the integrity and confidentiality of user data.

5.5 Application Testing and User Feedback

The application will undergo rigorous testing, including unit, integration, and stress tests, to ensure robustness and reliability. Edge cases in trading scenarios will be simulated to validate the application's performance under various conditions. Beta testing will be conducted with a controlled group of users to collect feedback and refine the user experience, ensuring that the application meets the needs of investors.

5.6 Final Presentation and Deployment

In the final phase, the project will culminate in the preparation of a detailed final report documenting the methodologies, results, and analysis. The application will be deployed on cloud platforms to ensure scalability and accessibility, providing users with a robust and user-friendly tool for stock price prediction and investment decision-making.

By the end of the second semester, the project aims to deliver a comprehensive and optimized financial forecasting system that integrates both machine learning and statistical models, offering a versatile solution for investors seeking to make informed decisions in the stock market.

5.7 Project Milestones and Future Schedule

To ensure a smooth development process and balanced workload, the project has been broken down into several distinct tasks. The objectives, along with their respective completion dates and task statuses, are detailed in **Table 1** (Project Schedule).

Table 1. Project Schedule

<table>
<tr><th colspan="2">Schedule</th><th colspan="2">Milestones
(predicted number of learning hours)</th><th>Status</th></tr>
<tr><td colspan="5">Phase 1: Project Inception</td></tr>
<tr><td>2024</td><td>Sep</td><td>• Preparation of Project Plan (10)
• Literature Review (10)
• Setup of Project Website (10)</td><td><u>Deliverables:</u>
1. Detailed Project Plan
2. Project Website Setup</td><td><u>DONE</u></td></tr>
<tr><td colspan="5">Phase 2: Project Elaboration</td></tr>
<tr><td rowspan="3">2024</td><td>Oct</td><td colspan="2">• Model Selection and Determination (10)
• Model Data Collection including numerical and textual data (10)
• Data Pre-processing and Cleaning (25)</td><td><u>DONE</u></td></tr>
<tr><td>Nov</td><td colspan="2">• Feature Extraction and Determination (15)
• Model Training (30)</td><td><u>DONE</u></td></tr>
<tr><td>Dec</td><td colspan="2">• LSTM-BERT Model Integration (10)</td><td><u>DONE</u></td></tr>
</table>

<table>
<tr><td></td><td></td><td colspan="2">Model Performance Evaluation and Reporting （10）
Front-end Basic Webpage Design and Style Selection （15）</td><td></td></tr>
<tr><td>2025</td><td>Jan</td><td>Interim Report Drafting (25)
Model Performance Evaluation（10）
Front-end Basic Webpage Design and Style Selection （10）</td><td><u>Deliverables:</u>
Preliminary Trained Model Prototype
Interim Report
First Presentation</td><td><u>DONE</u></td></tr>
<tr><td colspan="4">Phase 3: Project Construction</td><td></td></tr>
<tr><td rowspan="3">2025</td><td>Feb</td><td colspan="2">Frontend and Backend Construction (20)
Construction and Determination of the User-definable Trading Preference Rules (10)
Personalized Recommendation System (20)
User Data collection
Recommendations generation
Optimization and Updating
Trading system performance evaluation and optimization （20）</td><td><u>PENDING</u></td></tr>
<tr><td>Mar</td><td colspan="2">Final Report Drafting (25)
Frontend and Backend Construction (20)
Software Review and Testing (15)
Possible User trial and Collection of Feedback (15)</td><td><u>PENDING</u></td></tr>
<tr><td>Apr</td><td>Final Report Drafting (20)
Project Video Construction (15)
Preparation of Project Poster (10)</td><td><u>Deliverables:</u>
Project Exhibition – 3-min Video
Final Report
Final Application
Final Presentation</td><td><u>PENDING</u></td></tr>
</table>

6 Conclusion

In summary, the vast amount of information on existing financial websites and trading platforms often overwhelms investors, making it difficult for them to extract actionable insights. To tackle this challenge, our project aims to develop a user-friendly, one-stop financial management website that offers personalized investing advice through an advanced NLP model and a paper trading system. The website will leverage React for the frontend, the Django framework for the backend, and MySQL for database management. The integration of the paper trading system will utilize real-time market data sourced from stock price APIs, enhancing the overall functionality of the platform.

Preliminary results indicate that our integrated models, the LSTM-BERT combination, significantly outperform traditional single models in stock price forecasting. Moving forward, we will continue to refine these models and explore their applicability in real-world scenarios, ultimately developing a functional website that facilitates user interaction with the platform.

Throughout the project's progress, we have recognized potential challenges, including data collection and model training complexities, and have identified corresponding mitigations, such as utilizing multiple data sources. Additionally, several limitations should be acknowledged. The reliance on textual data from news and forum discussions may not fully capture the market attitude toward specific stocks, which could affect the model's predictive capability. Moreover, while sentiment analysis can provide valuable insights, it may not account for all factors influencing stock prices, such as macroeconomic indicators or geopolitical developments.

This project not only seeks to enhance investment strategies but also aspires to empower investors with the tools necessary for informed financial decision-making, ultimately fostering greater engagement in the stock market. Through ongoing enhancements and user feedback, we are committed to creating a robust platform that meets the evolving needs of investors in a dynamic financial landscape.

References

[1] E. F. Fama, “Efficient Capital Markets: A review of theory and empirical work,” The Journal of Finance, vol. 25, no. 2, p. 383, May 1970. doi:10.2307/2325486

[2] B. G. Malkiel, A Random Walk down Wall Street. New York: Norton, 1973.

[3] S. J. Brown, W. N. Goetzmann, and A. Kumar, “The dow theory: William Peter Hamilton’s track record reconsidered,” The Journal of Finance, vol. 53, no. 4, pp. 1311–1333, Aug. 1998. doi:10.1111/0022-1082.00054

[4] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, and K. P. Soman, “Stock price prediction using LSTM, RNN and CNN-sliding window model,” 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Sep. 2017. doi:10.1109/icacci.2017.8126078

[5] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia, and D. C. Anastasiu, “Stock price prediction using news sentiment analysis,” 2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService), Apr. 2019. doi:10.1109/bigdataservice.2019.00035

[6] R. Patel, V. Choudhary, D. Saxena, and A. K. Singh, “LSTM and NLP based forecasting model for stock market analysis,” 2021 First International Conference on Advances in Computing and Future Communication Technologies (ICACFCT), Dec. 2021. doi:10.1109/icacfct53978.2021.9837384

[7] Q. Ding, S. Wu, H. Sun, J. Guo, and J. Guo, “Hierarchical multi-scale gaussian transformer for stock movement prediction,” Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, pp. 4640–4646, Jul. 2020. doi:10.24963/ijcai.2020/640

[8] Bloomberg, “Bloomberg,” Bloomberg.com, 2023. https://www.bloomberg.com/ (accessed Nov. 10, 2024).

[9] “HKEX,” www.hkex.com.hk. https://www.hkex.com.hk/ (accessed Nov. 12, 2024).

[10] Yahoo Finance, “Yahoo Finance - Business Finance, Stock Market, Quotes, News,” Yahoo Finance, 2024. https://finance.yahoo.com/ (accessed Nov. 11, 2024).

[11] “Webull - Investing in Stocks, Trading, Online Broker and Research the Market,” Webull. https://www.webull.com/ (accessed Nov. 11, 2024).

[12] “FUTUBULL - Hong Kong,US and China Connect Stocks,” Futunn.com, 2024. https://www.futunn.com/en (accessed Nov. 11, 2024).

[13] “Longbridge HK - Open U.S. Stock Account, Open Hong Kong Stock Account, Hong Kong Stock Market Data, U.S. Stock Market Data - Longbridge - Longbridge Securities (Hong Kong),” Longbridge.com, 2023. https://longbridge.com/hk/ (accessed Nov. 11, 2024).

[14] “Bloomberg vs. capital IQ (capiq) vs. Factset vs. Refinitiv,” Wall Street Prep, https://www.wallstreetprep.com/knowledge/bloomberg-vs-capital-iq-vs-factset-vs-thomson-reuters-eikon/ (accessed Sep. 30, 2024).

[15] K. Cao, T. Zhang, and J. Huang, “Advanced hybrid LSTM-transformer architecture for real-time multi-task prediction in engineering systems,” Scientific Reports, vol. 14, no. 1, Feb. 2024, doi: 10.1038/s41598-024-55483-x.

[16] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, “LSTM: a Search space Odyssey,” IEEE Transactions on Neural Networks and Learning Systems, vol. 28, no. 10, pp. 2222–2232, Jul. 2016, doi: 10.1109/tnnls.2016.2582924.

[17] S. M. Jain, Introduction to Transformers for NLP. 2022. doi: 10.1007/978-1-4842-8844-3.