AI-Driven Stock Market Forecasting: A Hybrid Approach of News Analytics and Time-Series Data

by

ASLI GİZEM ULUSOY, 121200107
UMUTCAN ADIGUZEL, 120200087
RAMİZ KADAYIFÇI, 120200027

Supervised by

Dr. Emel Küpçü

Submitted to the
Faculty of Engineering and Natural Sciences
in partial fulfillment of the requirements for the

Bachelor of Science

in the
Department of Computer Engineering

—June, 2025

Abstract

This study proposes a hybrid method of predicting stock market movement by combining sentiment analysis from news around the world with technical indicators. Conventional models that only forecast closing prices show high overall accuracy, but they are unable to offer valuable advice and frequently do no better than random guessing. A more advanced model based on target class prediction has been developed and evaluated in order to overcome this limitation. When tested without sentiment data, a number of baseline models, such as Random Forest and LSTM, achieved maximum accuracies of 54% and 57%, respectively. The proposed LSTM model outperformed related works that reported accuracy of 54.6% and 68.4%, achieving a significant improvement of 72% through the use of FinBERT for integrating sentiment features and the implementation of class weighting and threshold optimization techniques. The AI model was complemented with a smartphone application that combined real-time analytics with community-driven financial debate, allowing users to view predictions and share ideas in a social network environment. This two-purpose platform seeks to enhance financial literacy while encouraging making informed choices. Future research will focus on improving data robustness via scalable microservice architectures, integrating macroeconomic indicators, utilizing high-frequency trading data, and fine-tuning sentiment granularity.

Table of Contents

Abstract **ii**

Table of Contents **iii**

List of Figures **v**

List of Tables **vi**

List of Symbols **vii**

List of Abbreviations **viii**

1 Introduction **1**
- 1.1 Financial Literacy . . . 1
- 1.2 Reasons for the Lag in Financial Literacy . . . 1
- 1.3 Theories and Hypotheses . . . 2
- 1.4 Deep Learning and Language Models . . . 3
- 1.5 Use of Language Models and Deep Learning in Financial Forecasting . . . 3
- 1.6 Key Ideas . . . 4

2 Related Works **5**
- 2.1 Investor Sentiment and Stock Price Correlation . . . 5
- 2.2 News Sentiment and Stock Price Correlation . . . 5
- 2.3 News Sentiment Analysis . . . 6
- 2.4 Predicting Stock Prices with News Sentiment . . . 7
- 2.5 Predicting Stock Prices with LSTM . . . 8
  - 2.5.1 Depending on Solely Close Prices . . . 8
  - 2.5.2 Including Sentiment Analysis . . . 9
- 2.6 Movement Prediction with Sentiment Analysis . . . 10

3 Design **11**
- 3.1 Backend . . . 11
- 3.2 Frontend . . . 12
- 3.3 Project Management . . . 18

4 Methodology **19**
4.1 Numerical Analysis 19
4.1.1 The Need of Target Analysis 19
4.1.2 Experiment on LSTM 20
4.2 News Sentiment Analysis 21
4.3 Target Analysis 23

5 Experiments Results **25**
5.1 Numerical Analysis Results 25
5.1.1 LSTM Model with Train, Test 25
5.1.2 LSTM Model with Train, Validation, Test 26
5.1.3 LSTM Model with Train, Validation, Test, Isolated Scaling 27
5.2 Target Class Analysis Results 28
5.2.1 Baseline Models for Comparison, Without Sentiment Data 28
5.2.2 LSTM Model Baseline, Without Sentiment Data 30
5.2.3 Enhanced LSTM and BiLSTM Target Class Prediction Model with Sentiment Data 31

6 Discussion **32**

7 Conclusion **32**

8 Future Work **33**

References **35**

List of Figures

1 Comparison of CNN predicted value and real value [18] 8
2 Prediction of Close values (TCS) in Conv1DLSTM model on Testing Set [11] 9
3 Daily close price predicted by S_EMDAM_LSTM [12] 10
4 Finvisor App Model Diagram 11
5 System Design 12
6 Login Page 13
7 Register Page 13
8 Post 14
9 Post Comments 14
10 Edit Profile Information 15
11 Share Post 15
12 All Stocks 16
13 Historical Stock Data Page 16
14 AI Prediction Page 17
15 Methodology of Trend Prediction Model 19
16 Actual vs. Predicted Prices (2020-2025) 20
17 Actual vs. Predicted Prices Closer Look (2024-2025) 20
18 Actual vs. Predicted Prices (2020-2025) 21
19 Actual vs. Predicted Prices Closer Look (2024-2025) 21
20 Actual vs. Predicted Prices (2020-2025) 25
21 Actual vs. Predicted Prices Closer Look (2024-2025) 25
22 Actual vs. Predicted Prices (2020–2025) 26
23 Actual vs. Predicted Prices – Closer Look (2024–2025) 26
24 Actual vs. Predicted Prices (2020–2025) 27
25 Actual vs. Predicted Prices Scatter Plot 27
26 Comparison Confusion Matrices 29
27 F1 Score vs Threshold for Best Model, LSTM 32
28 F1 Score vs Threshold for Best Model, BiLSTM 32

List of Tables

1 Evaluation Metrics for the Model 25
2 Evaluation Metrics for the Model 26
3 Evaluation Metrics for the Model 27
4 Comparative Results of Numerical Analysis Models 28
5 Classification reports summary for different models 29
6 Classification reports summary for different model configurations 30
7 Classification Report for the Best Model, LSTM 31
8 Classification Report, BiLSTM 31

List of Symbols

a, b, c	Scalars
$\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$	Vectors
$\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}$	Matrices
$\forall x$	Universal quantifier: for all x
$\exists x$	Existential quantifier: there exists x
$\sum_{n=1}^{N} x_n$	Sum of the $x_n : x_1 + x_2 + ... + x_N$
$\prod_{n=1}^{N} x_n$	Product of the $x_n : x_1 \cdot x_2 \cdot ... \cdot x_N$

List of Abbreviations

i.e.	Id est (Latin: this means)
e.g.	Exempli gratia (Latin: for example)
LSTM	Long Short Term Memory
NLP	Natural Language Processing
AI	Artificial Intelligence
GPT	Generative Pre-trained Transformer
BERT	Bidirectional Encoder Representations from Transformers
NDX	NASDAQ 100 Index
MNIR	Multinomial Inverse Regression
SVM	Support Vector Machine
XAI	Explainable AI
DJIA	Dow Jones Industrial Average
CNN	Convolutional Neural Network
LLM	Large Language Model
RNN	Recurrent Neural Networks
MSE	Mean Squared Error
RMSE	Root Mean Squared Error
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error

1 Introduction

1.1 Financial Literacy

Financial literacy involves knowledge and skills that are crucial for individuals to understand, interpret, and make conscious decisions about financial matters. Although it is a necessary skill in the modern world, a large number of people today are not financially literate. There are several reasons for such a situation. [21]

1.2 Reasons for the Lag in Financial Literacy

This lag in financial literacy can be gauged on a wide range, starting from the deficiencies within the education system to societal taboos and the economic challenges faced by individuals in daily life. In educational systems, there is often a lack of emphasis on financial literacy. Theoretical subjects such as mathematics and economics are integrated into the syllabus, but they usually fail to develop connections with real-life applications and remain confined to abstract knowledge. This disconnect makes it hard for people to understand concepts of finance. It is, therefore, important that financial literacy education be complemented by practical and realistic knowledge.[1]

Other crucial barriers to financial literacy include societal taboos and cultural influences. Most societies view money matters as taboo topics for discussion, while others make accessing financial knowledge hard. To be precise, not discussing money matters at the family level hampers individuals' chances of building awareness about this issue. In addition, cultural values might pressurize people in their financial decision-making process. It is not the easiest topic to improve upon, also considering the complex nature of the financial world and decorative jargon usage. These factors contribute not just to the difficulty in accessing precise information but also increase the risk of misinterpretation of information obtained. This day and age, knowing how to interpret information accurately becomes just as critical as accessing it. [5]

The fast pace of the modern world and technology promotes a consumption-oriented lifestyle in individuals. This pace of fast consumption also makes it difficult for people to plan for the long term and to set conscious goals regarding the future. Although technology facilitates access to financial information, using this information effectively depends on the individual's level of financial awareness. [22] Apart from this, economic difficulties are among the factors impeding financial literacy development. The daily-life pressure of economic circumstances places individual's attention on the short-run challenges of survival, thereby excluding a potential interest in long-term financial planning. Individuals at a subsistence level, finding it hard to make both ends meet, are hardly expected to carve out the time and energy to gain and build knowledge on finance. [27]

1.3 Theories and Hypotheses

Understanding the dynamics of price movements in financial markets is essential to examine the mechanisms underlying investment decisions. The Random Walk Theory and the Efficient Market Hypothesis are two key approaches to understanding price movements.

- **Random Walk Theory:** It believes that in financial markets, prices change randomly and no pattern could be predicted for it. This theory states that Random Walk Theory does not depend upon the previous trend, performance, or movements, whereas future prices cannot affect a share's market. For this reason, there can't be above-average yields in the market via different means such as technical and fundamental analysis. It only denotes that markets are unstable to believe in, and all variations have a statistical resemblance to the Random Walk. [10]

- **Efficient Market Hypothesis:** This approach assumes that prices reflect all available information and that the markets can be efficient in three levels. The Efficient Market Hypothesis postulates that the markets rapidly adjust to new information and then change their prices accordingly. The hypothesis has been studied in three different levels of efficiency: [19]

  - **Weak Form Efficiency:** All past price information is included in the price; thus, technical analysis is useless.

  - **Semi-Strong Form Efficiency:** Both past price information and all publicly available information are reflected in prices; fundamental analysis is also ineffective.

  - **Strong Form Efficiency:** All public and private information is incorporated into prices, making it impossible to outperform the market except through luck or illegal means. [19]

These theories provide a critical framework for understanding price movements in financial markets and evaluating market efficiency. While the Random Walk Theory emphasizes the unpredictability of markets, the Efficient Market Hypothesis focuses on how markets process information. Both theories play a important role in shaping investor behavior and strategies.

1.4 Deep Learning and Language Models

Artificial Intelligence is a science that aspires to create systems capable of learning, problem-solving, and decision-making like human intelligence. Deep learning, which is the very foundation of this technology, uses artificial neural networks to process large volumes of data and find patterns within them. NLP, a sub-area of AI, is supposed to understand, analyze, and generate human language. Some of the most fundamental tools in NLP are language models, using deep learning techniques to grasp the meaning of language. [15] These language models learn the relations within texts and hence can understand, to some extent, contexts or what some word will mean. A model, such as Generative Pre-trained Transformer (GPT), or BERT, has gone through a dataset of some billions of words and finally gets their comprehension. Similarly, it generates text with key tasks such as the following:

1. **Text Generation:** Creating new and meaningful sentences.
2. **Language Understanding:** Analyzing the context between words, sentences, and paragraphs.
3. **Sentiment Analysis and Classification:** Detecting the emotional tone (positive, negative, neutral) in texts.
4. **Machine Translation and Summarization:** Translating one language into another or summarizing texts concisely.

1.5 Use of Language Models and Deep Learning in Financial Forecasting

The financial sector is one of the areas where language models offer the greatest potential. This is because there is an abundance of textual data in the sector, such as news and report analysis, market sentiment forecasting, fraud detection, and algorithmic trading. The impact of these technologies on financial transactions creates significant opportunities for both investors and financial institutions. [17] Many financial models depend on numerical data, but sometimes numerical data may not be enough to predict market fluctuations.[6] This creates the necessity to analyze textual data in order to anticipate emotional market reactions, future developments, and possible crises. That is where AI and language models step in. These technologies have the capability to analyze not only numerical data but also news, reports, and announcements that can influence market movements. For example, an announcement about a company's bankruptcy or news of a global economic crisis can directly affect market movements. Capturing such events at an early stage allows investors to shape their strategies accurately. [14] Real-time developments mean everything in the financial markets. It is through periodic reports and announcements that companies, governments, and other institutions make their announcements known to the market. This news, when accurately analyzed, aids investors in formulating their strategies. Language models can scan through massive texts in no time and provide meaningful data. Examples include news on a company's earnings report, development of a new product, or any news related to a major agreement, all of which can stir stock prices. AI can analyze the positive or negative tones in such texts and guide investment decisions. The emotional tone within these texts may give several hints to investors about the company's future. Positive news drives up the stock price while negative events make the price fall. AI can run these analyses fast, saving precious time for investors and aiding them in making aware decisions. [3]

1.6 Key Ideas

AI is an effective tool in improving financial literacy and enabling individuals to make more aware decisions about their financial matters. The combination of time series analysis with NLP techniques allows for more accurate predictions in financial markets.

While the time series analysis foretell the future trend based on past movements in stock prices, NLP models analyze market sentiments after processing news and textual data. Such integration helps individuals and investors make strategic decisions.

The integration of AI into financial markets accelerates investment processes and enables more accurate and comprehensive analyses, driving important innovation in the sector.

2 Related Works

2.1 Investor Sentiment and Stock Price Correlation

According to the definition of **behavioral finance**, investor sentiment plays a role in predicting stock price values. In contrast, traditional finance theory argues that sentiment has no correlation with stock prices. McGurk et al. [20] examined whether investor mood on Twitter could predict unexpected stock returns. The authors developed sentiment indexes using a finance-specific dictionary-based methodology using Multinomial Inverse Regression (MNIR) on labeled data using 2.5 million tweets. The findings indicate that higher unusual returns are consistently predicted by good mood, particularly for mid- and large-cap stocks. While accuracy was enhanced by bigram models, negative sentiment was less constant. Out-of-sample estimates were slightly improved by sentiment data, indicating moderate but significant predictive potential.

2.2 News Sentiment and Stock Price Correlation

According to the **efficient market hypothesis**, news has an impact on changes in stock prices. In order to investigate this concept, Kalyani et al.[13] utilized news sentiment analysis to forecast stock trends for Apple Inc. A dictionary-based sentiment scoring system was used to label news articles, and Random Forest, SVM, and Naive Bayes models were used for classification. Random Forest was the most accurate of these, reaching up to 92% accuracy. The findings supported the use of news sentiment as a predictive indicator by clearly demonstrating a relationship between increasing stock prices and positive news and drops in stock prices.

In a different study, Bollen et al. [4] used almost 9.8 million mood-related tweets from 2008 to find out if public sentiment on Twitter could forecast movements in the stock market. The Calm mood dimension was Granger-causal to DJIA movements with a 2–6 day lead, according to their application of GPOMS, a multidimensional mood analysis tool. By integrating mood data to a neural network model, prediction accuracy increased significantly (up to 87% direction accuracy and reduced error rates), demonstrating that public sentiment (especially calmness) can greatly improve market trend prediction.

2.3 News Sentiment Analysis

Using neural networks and Long Short-Term Memory (LSTM) [24], Srinivas et al. [25] present a sentiment analysis model. Using data from Twitter, the study assesses three different approaches: Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Simple Neural Network. According to the data, LSTM performs better than the other methods, achieving the greatest accuracy of 87%.

In the study conducted by Lee et al. [16], it is stated that prior to the emergence of generative AI models, traditional methods were employed for financial text mining to analyze financial reports and predict stock prices. These methods relied primarily on the bag-of-words approach to generate word embeddings. However, financial texts typically lack a regular structure and the meaning of financial jargon can vary depending on the context. In contrast, pre-trained large language models (LLMs) using transformer architecture are capable of analyzing the complex relationships between words, allowing them to extract more meaningful insights from financial texts.

In the work of Bello et al.[2] BERT-based architectures and traditional models were tested for tweet sentiment classification. Using about 213,000 tweets, they discovered that BERT and BiLSTM performed significantly better than Word2Vec and traditional deep learning models, with an F1 score of 0.95 and an accuracy of a maximum of 93% The findings show how well context-aware embeddings capture sentiment details, indicating that transformer-based models perform better at financial sentiment analysis tasks.

Although Twitter is a popular data source for sentiment analysis, global news and other social media platforms can also be utilized. In order to classify financial news and Reddit postings, Deng et. al. [7] presented a semi-supervised sentiment analysis pipeline that uses Chain-of-Thought prompting and LLM-generated weak labels. Regression loss for soft sentiment scores was applied to train a lightweight student model (Charformer) using these labels. The LLM outperformed FinBERT versions on the FiQA News dataset, obtaining 97.3% accuracy, while the student model achieved 83.8%. The findings show that, even in the absence of sizable manually annotated datasets, LLM-labeled sentiment data can significantly improve sentiment analysis of financial news.

2.4 Predicting Stock Prices with News Sentiment

Srivastava, et al. [26] use two different datasets in their study. The first dataset consists of historical market prices. The second dataset consists of news items about important subjects from the past few years. Considering using solely numerical data for predictions was insufficient, they used natural language processing (NLP) for sentiment analysis. Long short-term memory (LSTM) architectures and recurrent neural networks (RNN) form the basis of the rest of their model. LSTM is used in the first module to analyze the data and generate predictions. The second module is designed to predict sudden stock movements by leveraging news articles and sentiment analyses which achieved 84.6% accuracy. At the conclusion of the study, the model obtained at most 0.102 RMSE score (for company "TTM"), which showed that it could accurately forecast steady changes that followed certain patterns, but it was less accurate at forecasting rapid moves in the market.

It has been observed that, in addition to financial data sources such as companies' annual reports, financial analysts often explore financial news and social media platforms to uncover hidden patterns in stock market prices. For instance, according to Dong et al.[8] news is a useful tool for forecasting changes in stock prices, but the subjectivity of stock analysts compromises the accuracy of this information. To address this, their study takes advantage of Twitter, which has an extensive user base and is a great source of breaking news. The developed BELT framework predicts stock values for the next day using an LSTM model with BERT-based sentiment analysis of financial tweets. BELT combines this data with past prices by selecting important tweets and extracts sentiment metrics. Models utilizing sentiment consistently outperformed price-only baselines across seven key stocks and the DOW; for example, AAPL's RMSE decreased from 0.110 to 0.027. Sentiment-based classifiers also outperformed the state-of-the-art baseline, StockNet, in direction prediction. These findings demonstrate that the accuracy of stock price predictions is greatly increased by news emotion.

Gite et al. [9] highlighted that it's still difficult to predict stock market fluctuations with great precision. They suggest a model that combines sentiment analysis from news articles with deep learning and machine learning methods, particularly Long Short-Term Memory (LSTM). Through the use of Explainable AI (XAI) tools like LIME, the model improves transparency by not just forecasting market movements but also offering explanations for its predictions. According to the findings, a more basic LSTM model based on stock price data obtained an accuracy of 88.73%, whereas the LSTM-CNN model obtained an accuracy of 74.76% on news headline data.

2.5 Predicting Stock Prices with LSTM

2.5.1 Depending on Solely Close Prices

Study conducted by Lu et al. [18] CNN-BiLSTM-Attention model was proposed in a study for predicting stock prices for 7,083 trading days using data from the Shanghai Composite Index. The model prioritizes critical steps by combining CNN for feature extraction, BiLSTM for temporal patterns, and Attention. With the lowest MAE (21.95), lowest RMSE (31.69), and highest R2 (0.9804), it outperformed eight baseline models. According to the results, combining several deep learning components improves forecast accuracy, which makes it a very useful tool for simulating complex market patterns.

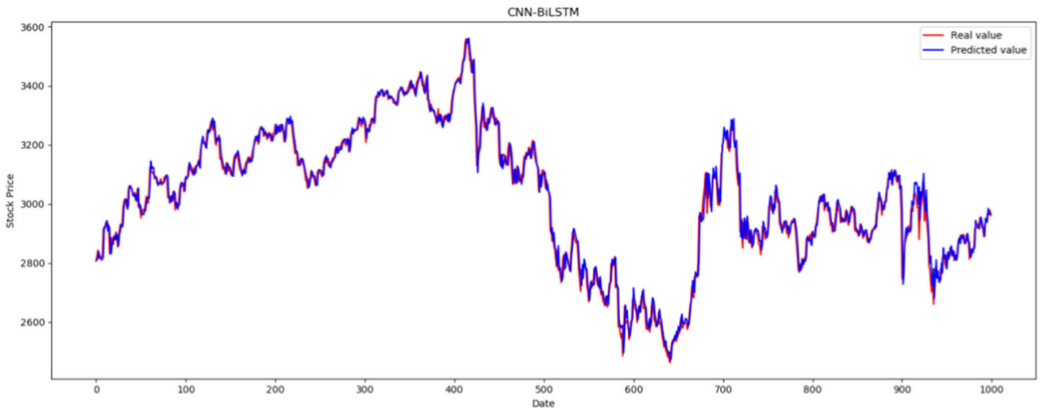

Figure 1: Comparison of CNN predicted value and real value [18]

Jain et al. [11] used historical OCHL data to predict the closing prices of Indian companies TCS and MRF for the next day via comparisons of CNN, LSTM, and a hybrid Conv1D-LSTM model. In this case, the Conv1D-LSTM model had the lowest errors (RMSE 0.0081, MAPE 1.98% for TCS, and RMSE 0.0179, MAPE 2.02% for MRF), consistently outperforming the others. The results indicate that short-term stock forecasting is more accurate when both temporal and spatial data are combined.

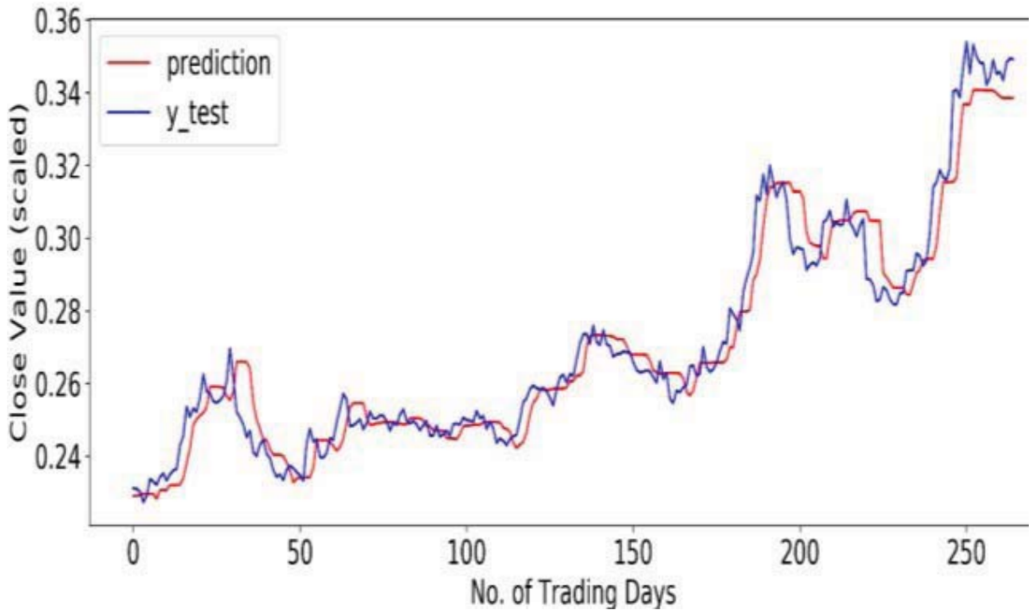

Figure 2: Prediction of Close values (TCS) in Conv1DLSTM model on Testing Set [11]

2.5.2 Including Sentiment Analysis

In order to predict stock closing prices, Jin et al. [12] developed a hybrid model (S_EMDAM_LSTM) that combines sentiment analysis, EMD, attention, and LSTM. The model outperformed the baseline LSTM by over 11 percent in accuracy, with a MAPE of 1.65%, RMSE of 3.20, and accuracy of 70.56% on AAPL data using sentiment from StockTwits and Yahoo Finance. The results suggest that using sentiment and deconstruction techniques considerably improves stock price prediction.

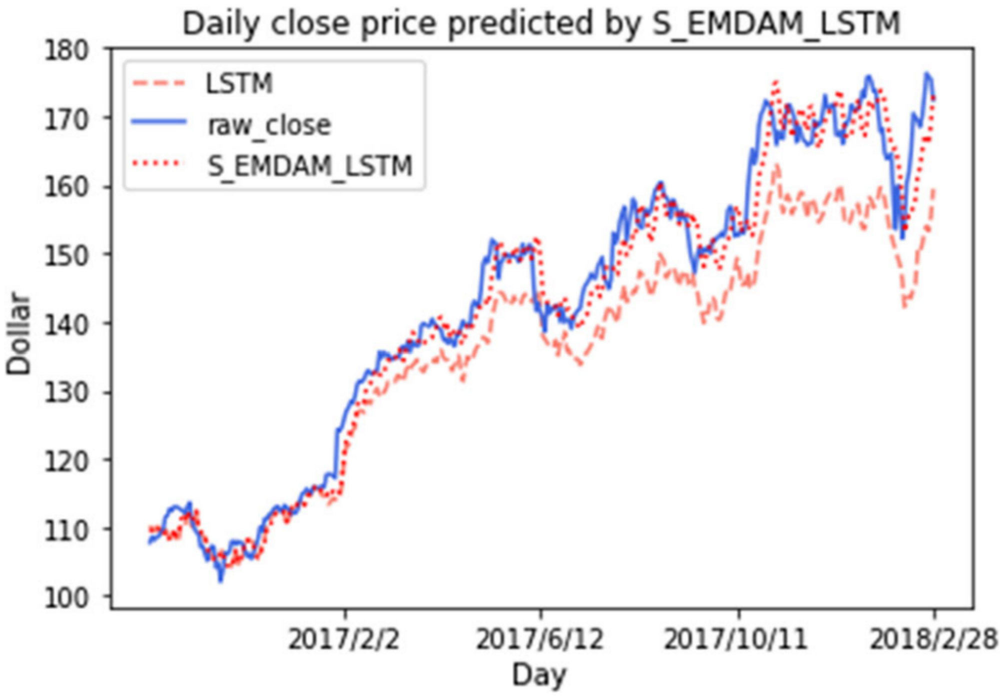

Figure 3: Daily close price predicted by S_EMDAM_LSTM [12]

2.6 Movement Prediction with Sentiment Analysis

Study conducted by Xu and Keselj [28], utilized LSTM and attention-based LSTM models to predict the direction of stocks over the next day through using technical indicators, historical stock prices, and sentiment from financial tweets (StockTwits). Implementing after-hours tweets, weighted by impact reach, produced the best outcomes. The attention-LSTM outperformed the regular LSTM (54.6% vs. 52.3%) and reached up to 63.78% accuracy (MSFT). The difficulty of short-term stock forecasting is shown by the results, which indicate that user impact and after-hours sentiment increase the accuracy of predictions yet enhancements are still modest.

In the studies mentioned, sentiment analysis was conducted with equal weights for all data points. However, by adding weighted content to the research, Qiu et al. [23] proposed a modified sentiment index that accounts for financial anomalies such as the day-of-the-week impact and weights investor feedback based on view count. Using data from Eastmoney.com and Baidu's SKEP model for sentiment classification, the authors proved how adding this refined sentiment to models like SVM and KNN greatly increased the accuracy of stock price prediction (e.g., SVM: 61.2% to 68.4%). Additionally, investment simulations revealed reduced risk and increased returns, proving that context-aware, weighted sentiment data enhances trading results and predictive performance.

3 Design

In design section, front-end, back-end and project management part of the project are discussed.

3.1 Backend

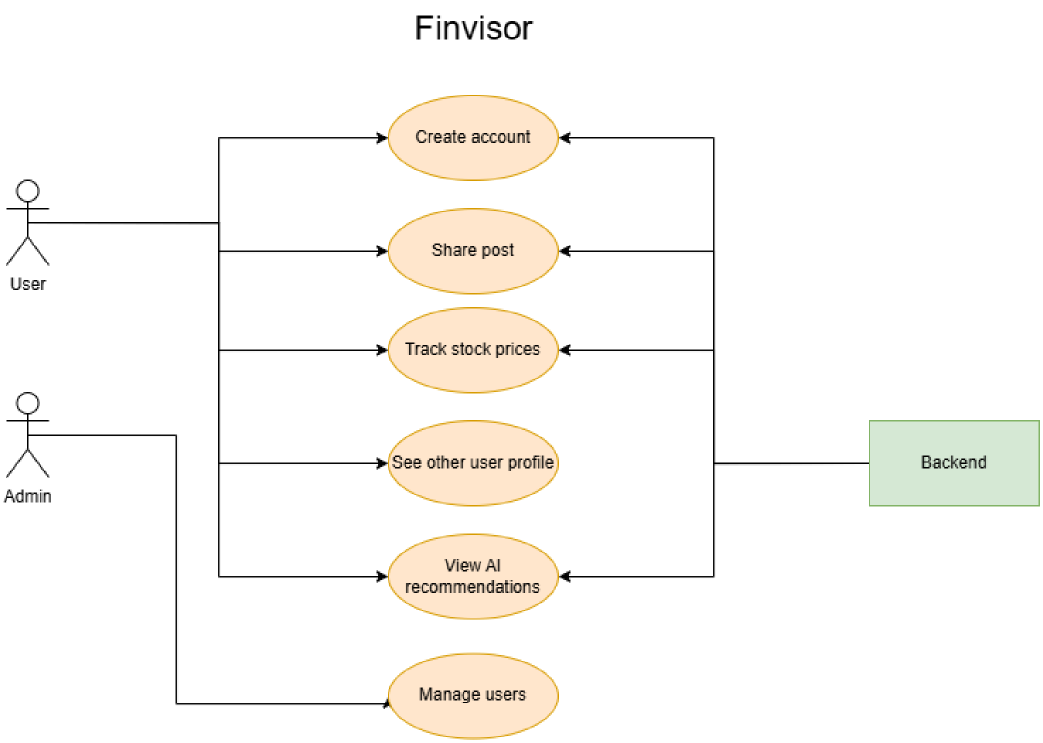

Figure 4: Finvisor App Model Diagram

In the backend Laravel framework of PHP used and classic MVS structure followed. for better organization special request and response classes which inherit original Laravel request and response classes are created. Request validation errors are customized which are explained better for user experience. For response necessary relations are loaded to avoid multiple requests for creating front-end content.

Relations and attributes that define relations between models are defined in model classes for ease of use. These attributes (`is_liked`, `is_post_owner`, `can_delete`, etc.) helps in middleware and check whether a post or comment is liked by the current user.

For better understanding and easier maintenance we add logging mechanisms for every possible backend failure. With log centralization tools we can track problems easily and fix them quickly

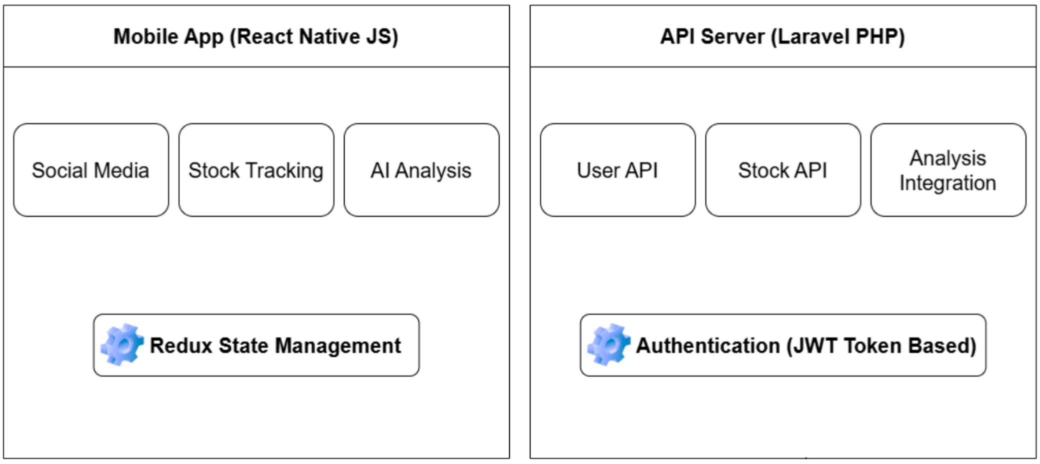

Figure 5: System Design

3.2 Frontend

In the frontend, react native is used which is a tool of javascript to make mobile applications. Native app connects backend with api calls. Redux state management tool is used for better user experience. Frontend builded with componential architecture to avoid every possible code replication. also Wagmi Charts is used to virtualize historical stock prices

Application has 3 main key features. These are social media, tracking current and historical stock prices and AI stock direction prediction. For mobile application we can list actions as follow:

1. Auth (Login, Register, Logout, etc.)

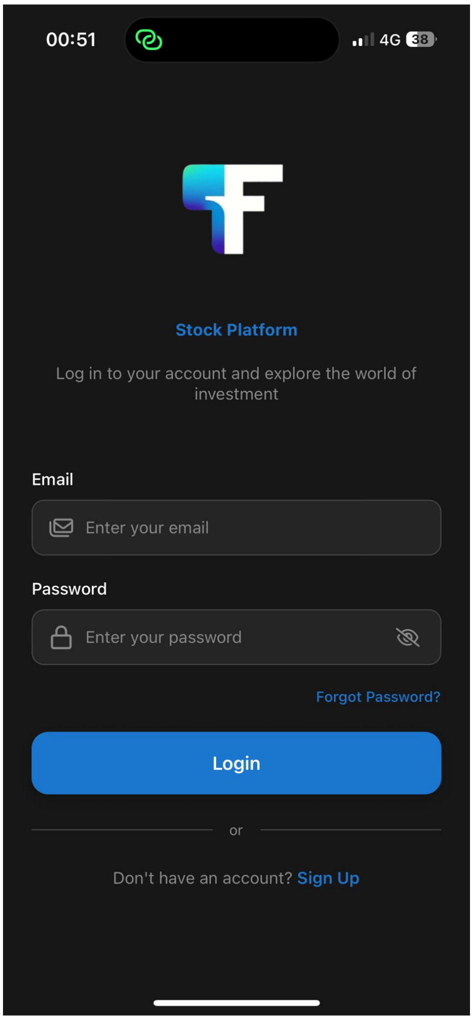

Figure 6: Login Page

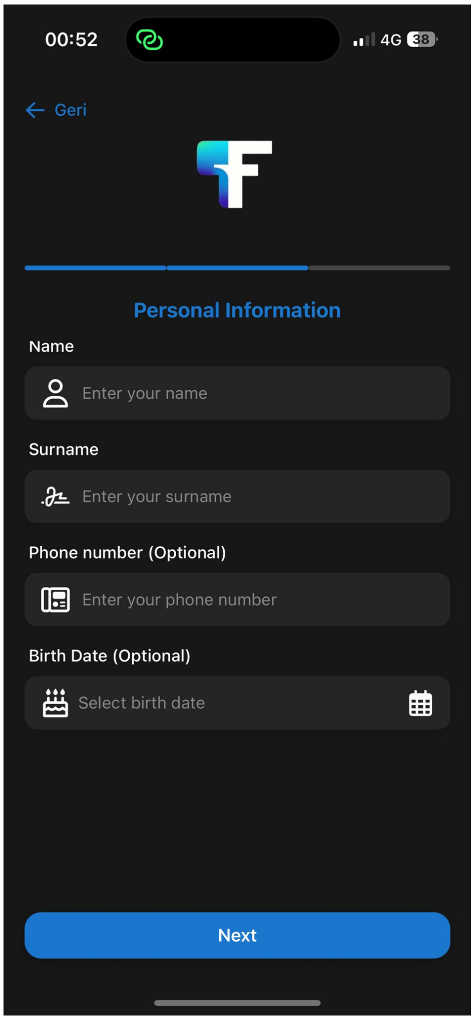

Figure 7: Register Page

Figures 6 and 7 shows the login and registration interfaces, respectively. To maintain a clean and user-friendly design, the registration process is divided into three separate steps. This step-by-step approach minimizes cognitive load and enhances the overall UX by guiding users through the process in a structured and manageable way.

2. Publish Post And Comment

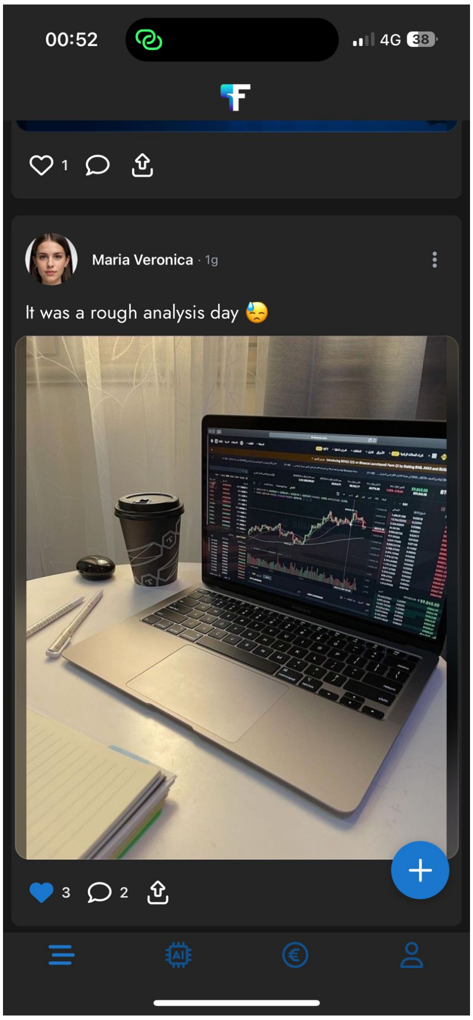

Figure 8: Post

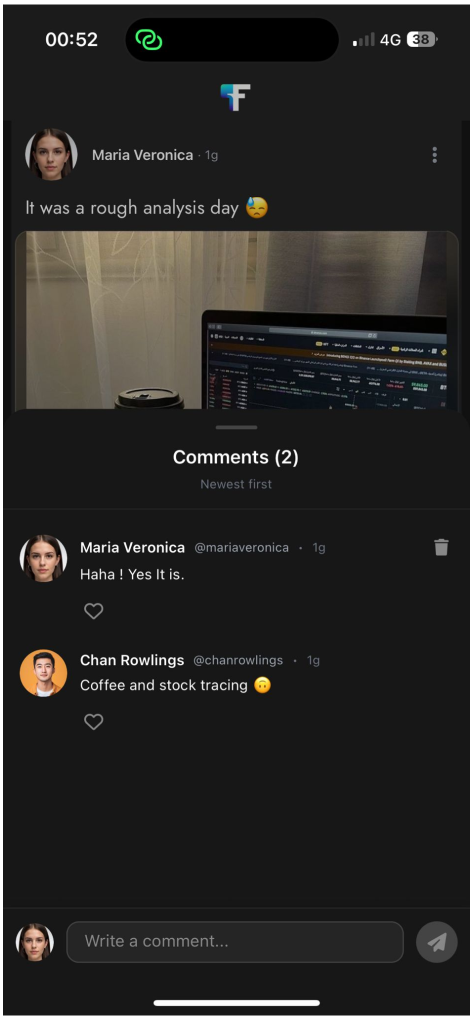

Figure 9: Post Comments

Figures 8 and 9 showcase the Post and Post Comment components, respectively. These components are designed with a focus on modularity and responsiveness, ensuring a seamless UX across different screen sizes. The Post component encapsulates all relevant metadata, media content, and interaction options (such as like, share, and comment).

3. Edit, Create Profile

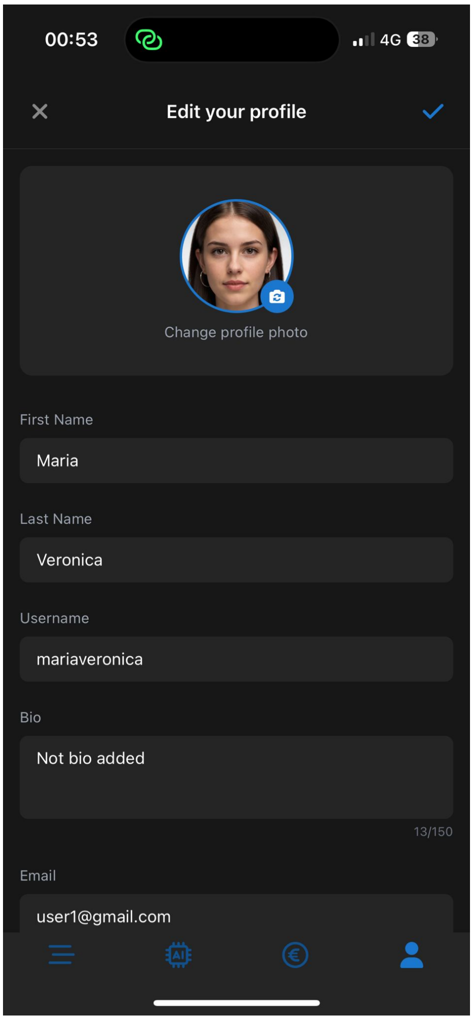

Figure 10: Edit Profile Information

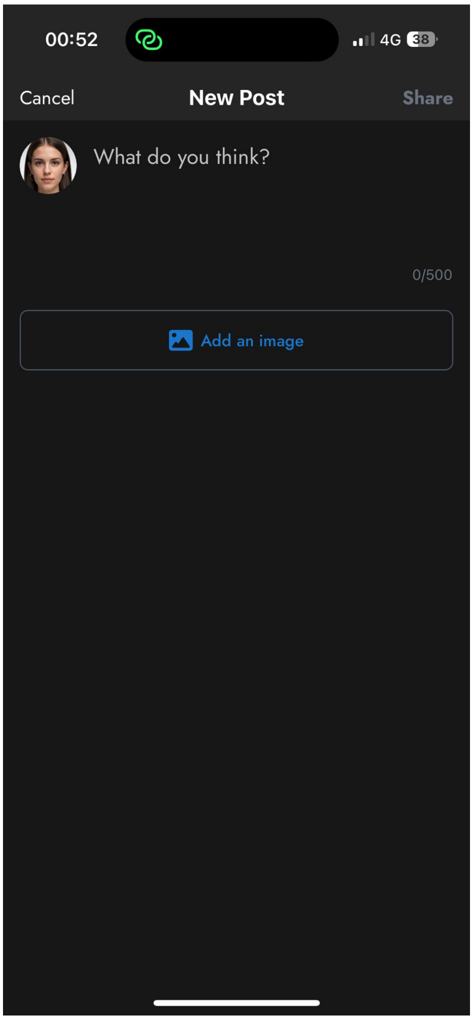

Figure 11: Share Post

Figure 11 presents the Post Form component, which enables users to create and submit new posts. The component is optimized for usability, incorporating features such as dynamic input validation, file upload support, and real-time previews. Its architecture promotes re-usability and scalability, allowing it to be integrated seamlessly across various parts of the application where content creation is required.

Figure 10 displays the Profile Edit component, which allows users to update their personal information and preferences. It is built with a strong emphasis on user experience and data integrity, offering intuitive input controls, instant feedback mechanisms, and secure form submission. The component supports responsive design and client-side validation to ensure consistent performance across devices.

4. Historical Stock data & All stocks page

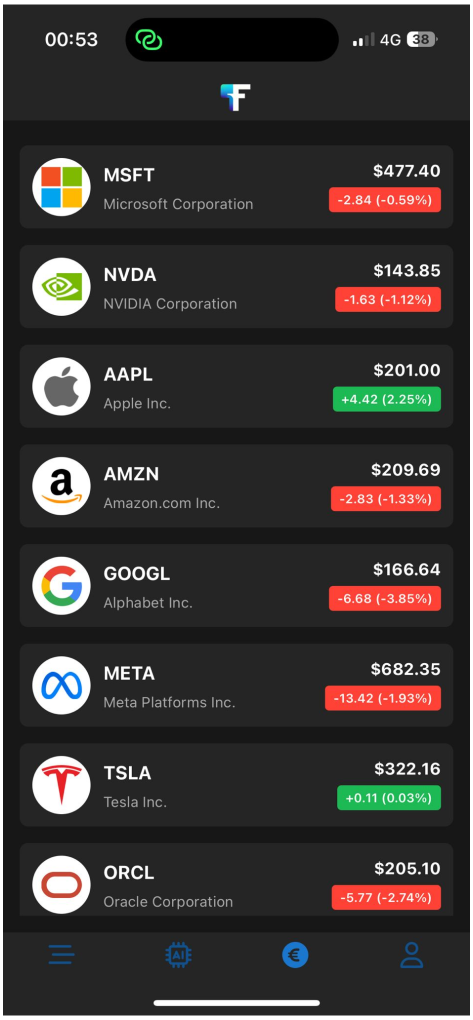

Figure 12: All Stocks

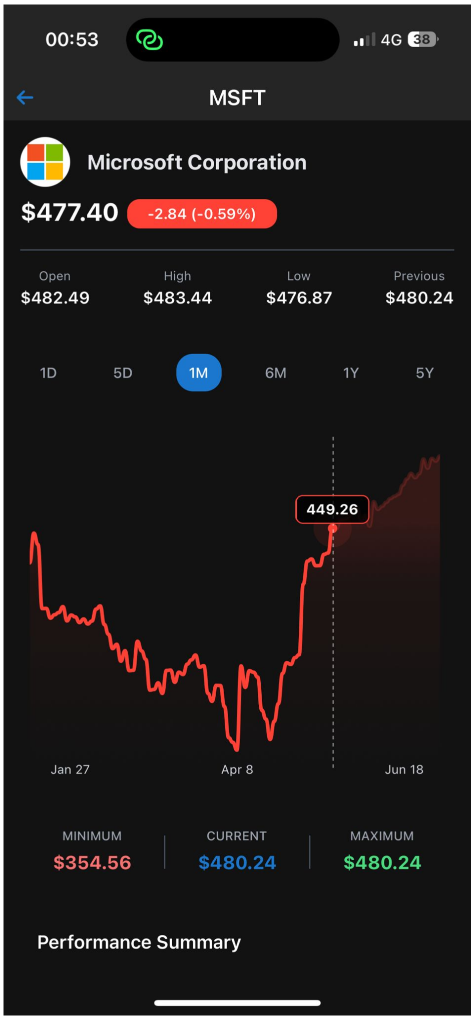

Figure 13: Historical Stock Data Page

Figure 13 illustrates the Single Historical Stock Data page, which provides users with detailed insights into the historical performance of a selected stock. The page features interactive charts. Designed for performance and scalability, the data is retrieved asynchronously and cached to minimize load times and ensure a smooth user experience during data exploration and analysis.

Figure 12 shows the All Stocks Current page, which aggregates and displays real-time data for all tracked stocks in a tabular format. It includes key metrics such as current price, daily change, volume, and market status, with sorting and search functionalities to enhance usability. The page is built using an efficient data-fetching mechanism with periodic updates, enabling users to monitor market movements with minimal latency.

5. AI Prediction Results Page

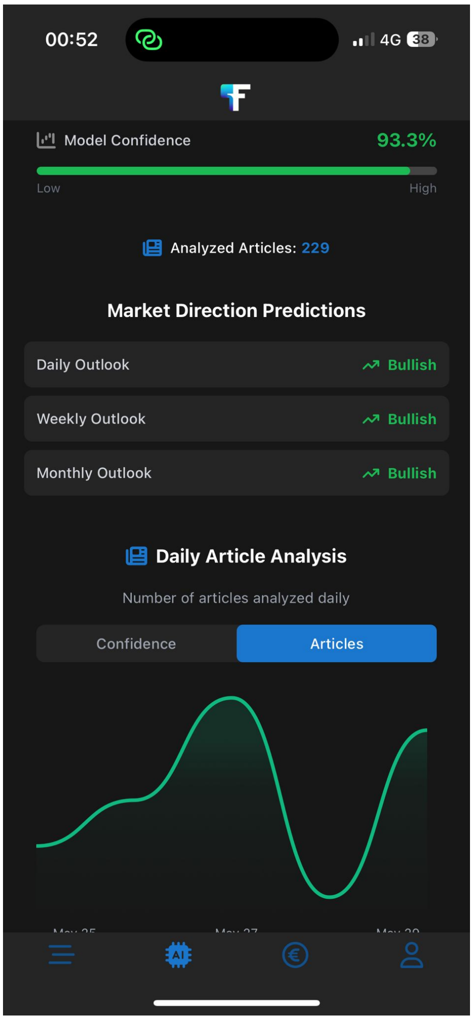

Figure 14: AI Prediction Page

Figure 14 illustrates the AI Prediction page, where an AI model forecasts the directional movement of the NASDAQ index on a daily, weekly, and monthly basis. In addition to the predicted market direction, the page presents supporting indicators such as the number of news articles processed per day, aggregate sentiment analysis results, and the model's confidence level for each forecast. The layout is designed to help users quickly assess the AI's outlook while maintaining transparency in how predictions are derived.

3.3 Project Management

For project management the Scrum method is used and 2 weeks decided for each sprint. The project divided into 3 parts (AI model, Frontend, and Backend). Everybody was managing one of the 3 parts. At the beginning of each sprint, a planning meeting were made where the goals for the next two weeks discussed and assigned tasks accordingly. Throughout the sprint, daily stand-up meetings continued to track progress, identify blockers, and ensure smooth coordination between the three parts. Each team member was responsible not only for completing their own tasks but also for regularly communicating with others to align on integration points. At the end of each sprint, a sprint review made where we reviewed the progress made, followed by a retrospective to discuss what went well, what could be improved, and how we could enhance our collaboration in the next sprint. This structured approach enabled us to maintain clear organization, adhere to deadlines, and ensure that the AI model, frontend, and backend components were developed cohesively and in parallel.

4 Methodology

In this section methods and techniques used and will be used will discussed below.

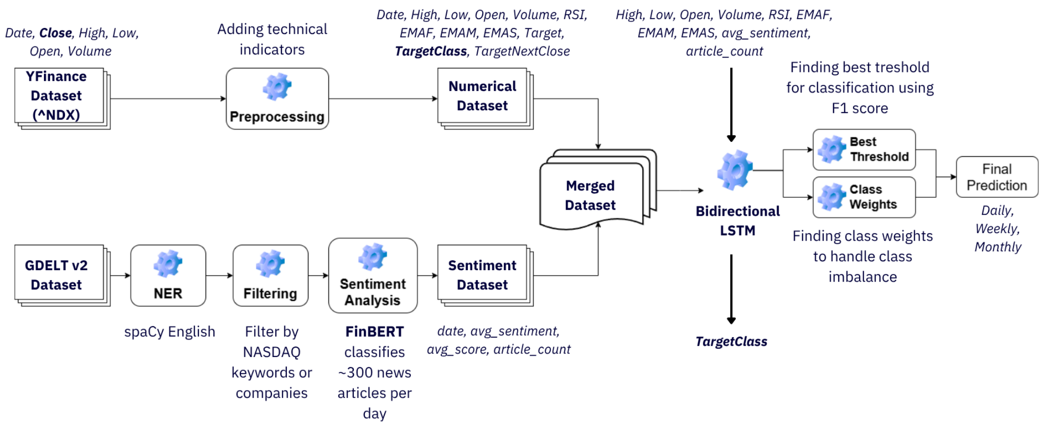

Figure 15: Methodology of Trend Prediction Model

4.1 Numerical Analysis

4.1.1 The Need of Target Analysis

In this section, the statement for the need of target analysis is discussed.

Initial model built gathered the financial data from **Yahoo Finance**, including "Close" prices of the stock `^NDX` for the dates between 1985-10-01 to 2025-06-17 (10006 rows of data) with interval of 1 day.

For improved computational efficiency, the dataset was scaled to the [0, 1] interval using the **MinMaxScaler** normalization method. Then, divided into train, validation and test datasets. After preprocessing, the dataset was partitioned into training (70%), validation (15%), and test (15%) subsets to ensure robust model evaluation. The dataset was transformed into sequences of 60-day sliding windows using a lookback function to serve as input for the LSTM model.

The LSTM model was trained using a sequential architecture consisting of two stacked LSTM layers. The first LSTM layer had 128 units and returned sequences to allow further temporal processing by the subsequent LSTM layer, which also contained 128 units but returned a final output vector. To prevent overfitting, a dropout rate of 0.3 was applied after each LSTM layer. The output from the LSTM layers was passed through a dense layer with 64 units and ReLU activation, followed by a final dense layer with a single neuron to produce the prediction.

The model was compiled using the Adam optimizer and the Huber loss function, which is robust to outliers and suitable for regression tasks.

To prevent overfitting and reduce unnecessary training time, early stopping was employed. The training process was monitored on the validation loss, and stopped if no improvement was observed for 15 consecutive epochs. Furthermore, the model automatically reverted to the weights from the epoch with the best validation performance.

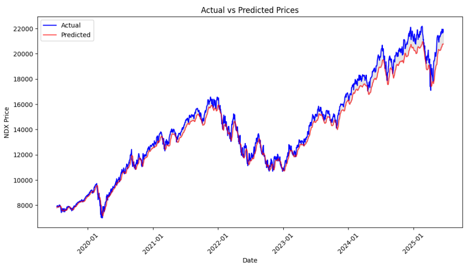

Figure 16: Actual vs. Predicted Prices (2020-2025)

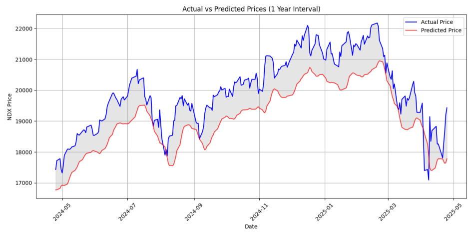

Figure 17: Actual vs. Predicted Prices Closer Look (2024-2025)

A major issue with this approach is the low directional accuracy, which remained at only 51.47%, making the model's predictions nearly equivalent to random guessing. Notably, there was no evidence of overfitting in the model. Although the model appeared to yield good results in terms of loss metrics, its inability to make reliable directional forecasts prompted a further investigation. To understand the cause of this behavior, an experiment discussed in Section 4.1.2 was conducted.

4.1.2 Experiment on LSTM

In the previous section, the dataset was divided into three parts: training, validation, and testing. This allowed the model to adjust based on the unseen data in the validation set before being evaluated on the test set. To further investigate the issue, in this section, the model is trained solely on the training dataset and directly evaluated on the test dataset.

Under the same configuration settings, the data was collected, scaled, partitioned, and input into the identical model. Datasets divided as: training (80%), test (20%).

Although the model initially appears highly successful with an accuracy of 98.29%, this outcome actually reflects a form of data leakage. Since the model uses the close price data during training, it essentially learns to predict values very close to the next day's close price. Consequently, the model's

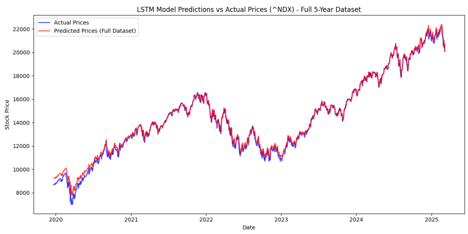

Figure 18: Actual vs. Predicted Prices (2020-2025)

Figure 19: Actual vs. Predicted Prices Closer Look (2024-2025)

predictions for the following day closely resemble the price on the day it makes the prediction, rather than genuinely forecasting future movements.

In conclusion, although close price prediction is a commonly used method in the literature (and is presented with performance figures; 1, 2, 3), this study goes beyond by **focusing not only on close price prediction but also on directional accuracy**. Therefore, it offers a more advanced and comprehensive modeling approach discussed in the Target Analysis (Section 4.3).

4.2 News Sentiment Analysis

As discussed in Sections 2.2 and 2.5.2 it is proven that news sentiment affects stock market, on the contrary of Efficient Market Hypothesis 1.3.

In this section, the initial goal is collecting global news that likely to influence `^NDX` stocks and perform sentiment analysis on gathered news articles. To accomplish that GDELT v2 database is used.

GDELT v2: The GDELT v2 dataset (Global Database of Events, Language, and Tone) is a large-scale, open-source database that monitors global news media in over 100 languages, extracting structured information about geopolitical events, actors, locations, and sentiment. Using the CAMEO coding system, it captures who did what to whom, when, and where; along with tone, impact scores, and emotional content. Continuously updated and publicly available, GDELT v2 is widely used in areas such as event forecasting, sentiment analysis, and financial market prediction.

Data was first fetched from the GDELT database and preprocessed. Since the data was provided as zip files divided in 15-minute intervals, the files were unzipped, grouped by date, and concatenated. It is observed that each day had approximately 76,000 news articles.

The Initial Methodology

- **Entity Extraction**: Named Entity Recognition (NER) was performed using spaCy's English model to identify and extract organization names from the text.
- **Financial Domain Filtering**: To ensure data reliability, only articles from trusted financial news sources were retained by filtering based on domain names.
- **Keyword Matching in Title and Content**: A comprehensive list of NASDAQ-100 companies, including both their full names and ticker symbols, was used to identify relevant articles. Only articles mentioning these companies in the title or full text were selected.
- **Financial Sentiment Analysis**: Sentiment analysis was conducted using FinBERT, a BERT-based language model fine-tuned for financial texts, to classify the filtered news articles as positive, negative, or neutral.
- **Daily Sentiment Aggregation**: The sentiment scores of all relevant news articles were averaged on a daily basis to generate a single sentiment label (positive or negative) for each day.

Due to the computational cost of these tasks, a manual filtering process was applied. The full content of each news article (up to a maximum length of 200 tokens) was used, and only articles containing relevant keywords from the predefined list were retained. These filtered articles were then analyzed using FinBERT for sentiment classification.

FinBERT Pipeline

- **Model Selection:** The FinBERT model (`yiyanghkust/finbert-tone`), a BERT-based model fine-tuned for financial sentiment analysis, is selected.
- **Tokenizer Loading:** A corresponding BERT tokenizer is loaded using `BertTokenizer.from_pretrained()` to convert input text into token IDs suitable for the model.
- **Model Loading:** The pretrained FinBERT model is loaded using `BertForSequenceClassification.from_pretrained()` to enable sentiment classification.

Following the sentiment analysis process, six months of data—consisting of approximately 200–300 news articles per day—were analyzed. This resulted in a structured daily output with the following fields:

- `date`: The specific day for which the aggregated sentiment was calculated.
- `avg_sentiment`: The average sentiment score for the day, normalized to a range of $[-1, 1]$ where -1 indicates strong negativity, 0 is neutral, and 1 is highly positive.
- `avg_score`: The average confidence score of the model's sentiment predictions for that day, ranging from $[0, 1]$
- `article_count`: The number of news articles successfully analyzed for that particular day.

4.3 Target Analysis

As in the discussed in Section 4.1.1, rather than solely depending on stock prices, it is necessary to focus on directional accuracy too.

To prepare the dataset for stock price prediction with integrated sentiment analysis, the following preprocessing steps were applied:

- **Historical Data Collection**: Historical price data for the NASDAQ-100 index (`^NDX`) was retrieved using the `yfinance` library for the period between 2024-09-30 and 2025-06-02.
- **Technical Indicator Calculation**: Several technical indicators were computed using the `ta` (technical analysis) library to enrich the dataset:
  - **RSI (Relative Strength Index)** with a window of 15 days,
  - **EMAF (Exponential Moving Average - Fast)** with a 20-day period,
  - **EMAM (Exponential Moving Average - Medium)** with a 100-day period,
  - **EMAS (Exponential Moving Average - Slow)** with a 150-day period.

  Missing indicator values were backfilled using the `bfill` method to maintain continuity.
- **Target Variable Construction**:
  - `Target`: The difference between the next day's adjusted close price and the current day's opening price.

  - `TargetClass`: A binary label indicating whether the next day's price increased (1) or not (0).
  - `TargetNextClose`: The adjusted close price of the following day, used for regression-based predictions.

- **Sentiment Data Integration**:
  - Sentiment scores generated from news articles were loaded.
  - A left join was performed on the `Date` column to merge sentiment features: `avg_sentiment`, `avg_score`, and `article_count`.

This final dataset, enriched with both technical indicators and sentiment metrics, was used to train models for predicting next-day stock price direction and magnitude.

The minority class accounted for 43% of the data, indicating a class imbalance in the target variable. Techniques for handling class imbalance were used to address this problem. `'High', 'Low', 'Open', 'Volume', 'RSI', 'EMAF', 'EMAM', 'EMAS', and 'avg_sentiment'` are the feature columns that are used in the model. Notably, the input features purposefully exclude the closing price (`Close`).

Using a lookback value of 15, a sliding window technique was used. The dataset was normalized using the StandardScaler after being split into training and testing sets with corresponding ratios of 0.8 and 0.2.

In order to identify the configuration that produces the greatest F1 score, the F1 model was then trained using a variety of configurations. Based on the model scores, a threshold value was then chosen for classifying what is expected as either class 0 or 1. The Experiments & Discussing 5 section presents the outcomes of these investigations.

5 Experiments Results

5.1 Numerical Analysis Results

In this section, the results of the numerical analysis which is also mentioned in Section 4.1.1 is evaluated.

Following experiments conducted with same configurations. Data is gathered from Yahoo Finance with 1 day interval, between 1985-10-01 to 2025-06-17 for the ticker `^NDX`. The LSTM model used a sequential architecture with two stacked LSTM layers (128 units each), where the first returned sequences and the second returned the final output. Dropout (0.3) was applied after each LSTM layer to reduce overfitting. The output was passed through a dense layer with 64 ReLU units and a final dense layer for prediction. The model was compiled with the Adam optimizer and Huber loss, suitable for robust regression.

5.1.1 LSTM Model with Train, Test

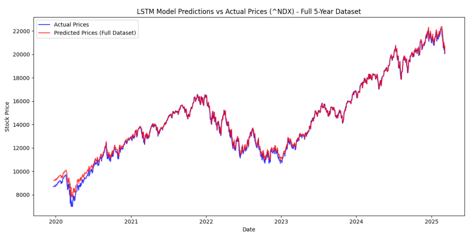

Figure 20: Actual vs. Predicted Prices (2020-2025)

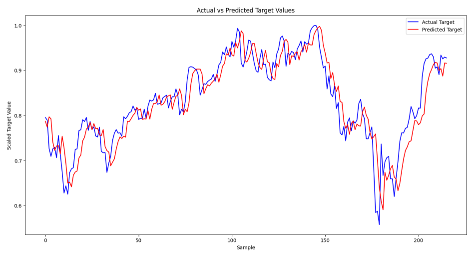

Figure 21: Actual vs. Predicted Prices Closer Look (2024-2025)

Metric	Value
Mean Squared Error (MSE)	79585.6523
Root Mean Squared Error (RMSE)	282.109
Mean Absolute Error (MAE)	209.7027
R-squared Score (R^2)	0.9932
Explained Variance Score	0.9880
Mean Absolute Percentage Error (MAPE)	1.71%
Directional Accuracy	**47.22%**
MAPE Percentage	**98.29%**

Table 1: Evaluation Metrics for the Model

In this configuration, the data was scaled before splitting into training and testing sets. However, this approach introduces data leakage, as the model is exposed to information about the overall minimum and maximum values of the dataset—including those from the test set during the scaling process. While this setup yields the lowest MAPE (Mean Absolute Percentage Error), it also results in the poorest directional accuracy, indicating that the model is fitting the scale of the data rather than genuinely learning to predict market direction.

5.1.2 LSTM Model with Train, Validation, Test

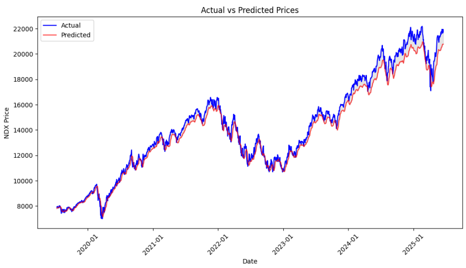

Figure 22: Actual vs. Predicted Prices (2020–2025)

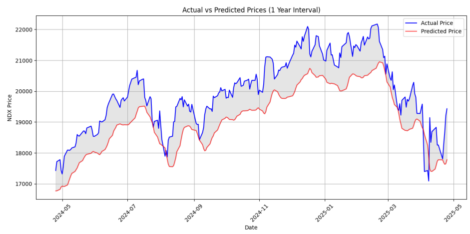

Figure 23: Actual vs. Predicted Prices – Closer Look (2024–2025)

Metric	**Value**
Mean Squared Error (MSE)	341,492.8248
Root Mean Squared Error (RMSE)	584.3739
Mean Absolute Error (MAE)	459.6304
R-squared Score (R^2)	0.9771
Explained Variance Score	0.9880
Mean Absolute Percentage Error (MAPE)	3.0041%
Directional Accuracy	**51.47%**
MAPE Percentage	**97.00%**

Table 2: Evaluation Metrics for the Model

The same problem of data leakage is introduced in this scenario since the data was pre-scaled before splitting. Despite training on a validation set, the scaling process gave the model indirect access to the worldwide distribution of data. The influence of the dataset's overall median during normalization resulted in the model learning the proper pattern but repeatedly predicting values that were lower on the y-axis.

5.1.3 LSTM Model with Train, Validation, Test, Isolated Scaling

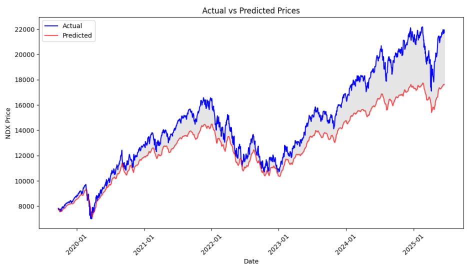

Figure 24: Actual vs. Predicted Prices (2020–2025)

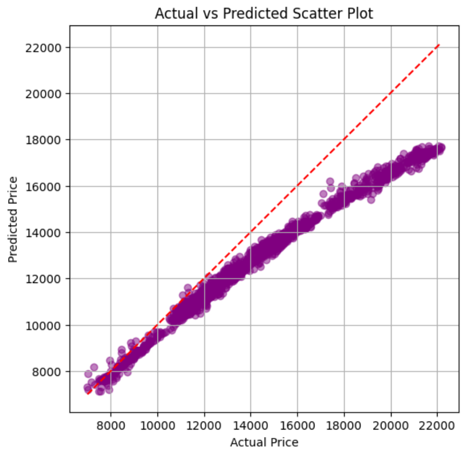

Figure 25: Actual vs. Predicted Prices Scatter Plot

Metric	**Value**
Mean Squared Error (MSE)	3699373.7409
Root Mean Squared Error (RMSE)	1923.3756
Mean Absolute Error (MAE)	1548.1073
R-squared Score (R^2)	0.7353
Explained Variance Score	0.9048
Mean Absolute Percentage Error (MAPE)	9.5421%
Directional Accuracy	**50.66%**
MAPE Percentage	**90.46%**

Table 3: Evaluation Metrics for the Model

In this section, unlike the previous configurations, the data was not pre-scaled. Instead, the dataset was first split into training, validation, and test sets. The scaler was then fitted exclusively on the training data, ensuring no information from the validation or test sets leaked into the scaling process. This approach yielded the lowest MAPE score, indicating improved generalization. However, the predicted values were significantly shifted downward along the y-axis, likely influenced by the lower median of the training subset used for scaling.

Model	$R^2 score$	MAPE Accuracy	Directional Accuracy
5.1.1	0.9932	**98.29**%	47.22%
5.1.2	0.9771	97.00%	**51.47**%
5.1.3	0.7353	90.46%	50.66%

Table 4: Comparative Results of Numerical Analysis Models

5.2 Target Class Analysis Results

5.2.1 Baseline Models for Comparison, Without Sentiment Data

In this section, several baseline models are trained and evaluated to provide a performance comparison against the proposed model. Historical data for the NASDAQ-100 index (`^NDX`) was obtained from Yahoo Finance, covering the period from June 1, 2015, to June 1, 2025, comprising a total of 2,516 daily records. To enrich the dataset, a comprehensive set of technical indicators was computed and appended, including:

- Relative Strength Index (RSI)
- Moving Average Convergence Divergence (MACD)
- MACD Signal Line
- Stochastic Oscillator (Stoch)
- Williams
- Commodity Channel Index (CCI)
- Average Directional Index (ADX)
- Rate of Change (ROC)
- On-Balance Volume (OBV)
- Exponential Moving Average (EMA, 10-day)
- Simple Moving Average (SMA, 20-day)
- Bollinger Bands Middle (BB_BBM), Upper (BB_BBH), and Lower (BB_BBL) Bands

Model	Class	Precision	Recall	F1-score	Support	Accuracy
Logistic Regression	0	0.35	0.09	0.15	150	0.53
	1	0.56	0.87	0.68	196	
	Macro Avg	0.45	0.48	0.41	346	
	Weighted Avg	0.47	0.53	0.45	346	
Random Forest	0	0.44	0.23	0.30	150	**0.54**
	1	0.57	0.78	0.66	196	
	Macro Avg	0.50	0.50	0.48	346	
	Weighted Avg	0.51	0.54	0.50	346	
XGBoost	0	0.43	0.24	0.31	150	0.47
	1	0.54	0.73	0.62	196	
	Macro Avg	0.48	0.48	0.47	346	
	Weighted Avg	0.49	0.47	0.47	346	
MLP Neural Net	0	0.44	0.51	0.47	150	0.51
	1	0.58	0.52	0.54	196	
	Macro Avg	0.51	0.51	0.51	346	
	Weighted Avg	0.52	0.51	0.51	346	

Table 5: Classification reports summary for different models

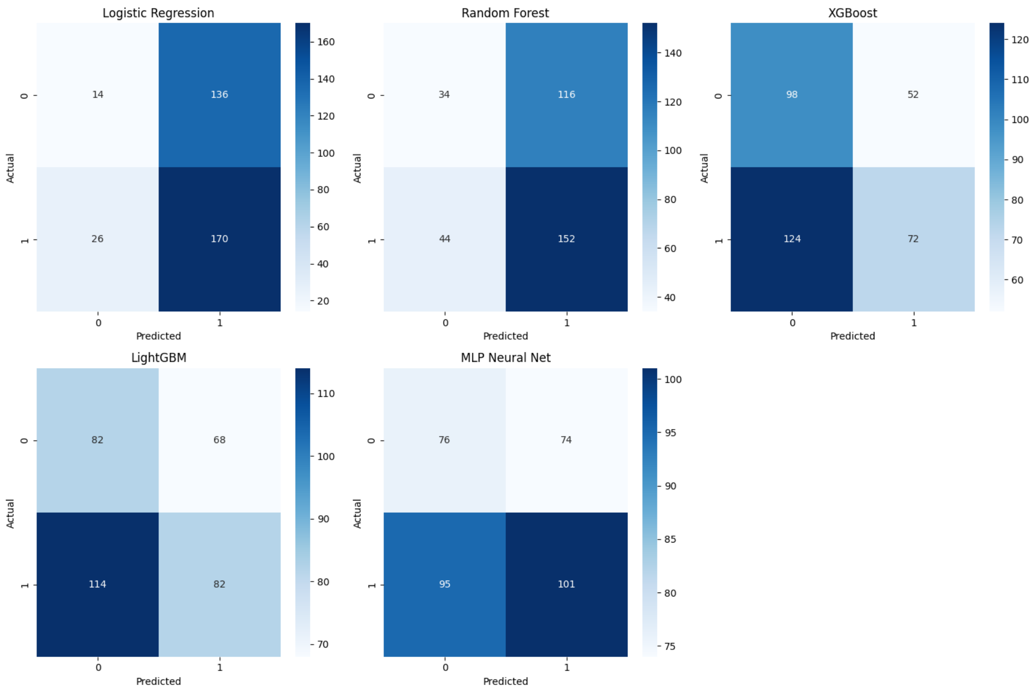

Figure 26: Comparison Confusion Matrices

5.2.2 LSTM Model Baseline, Without Sentiment Data

An LSTM model was trained using various configurations, including class weighting to address class imbalance and early stopping to prevent overfitting. The model architecture consists of two stacked LSTM layers, the first with 128 units returning sequences to capture temporal dependencies, followed by a 64-unit LSTM layer that outputs the final sequence representation. Dropout layers with a rate of 0.3 are applied after each LSTM layer to mitigate overfitting. The LSTM outputs are then passed through a dense layer with 32 units and a configurable activation function (defaulting to ReLU), followed by a final dense layer with a single neuron and sigmoid activation for binary classification. The model is compiled with the Adam optimizer (default learning rate 0.001) and binary cross-entropy loss, with accuracy as the evaluation metric.

Model	Class	Precision	Recall	F1-score	Support	Accuracy
32/20/0.001	0.0	0.42	0.11	0.18	152	0.55
	1.0	0.57	0.89	0.69	201	
	Macro Avg	0.50	0.50	0.43	353	
	Weighted Avg	0.51	0.55	0.47	353	
64/30/0.0005	0.0	0.44	0.51	0.47	152	0.51
	1.0	0.58	0.51	0.54	201	
	Macro Avg	0.51	0.51	0.51	353	
	Weighted Avg	0.52	0.51	0.51	353	
128/40/0.0001	0.0	0.40	0.01	0.03	152	**0.57**
	1.0	0.57	0.99	0.72	201	
	Macro Avg	0.48	0.50	0.37	353	
	Weighted Avg	0.50	0.57	0.42	353	

Table 6: Classification reports summary for different model configurations

As shown in Table 6, the models perform only marginally better than random guessing when sentiment analysis is not incorporated. Among the baseline models, the Random Forest classifier achieves the best performance. For the LSTM model, the optimal configuration was found with a batch size of 128, 40 epochs, and a learning rate of 0.0001, yielding the best results within the tested parameter space.

Based on these results, we proceed into more complex models, focusing again LSTM networks due to its performance on sequential data.

5.2.3 Enhanced LSTM and BiLSTM Target Class Prediction Model with Sentiment Data

In this section, the results of the model described in the methodology are evaluated and discussed. The model was trained and tested under various configurations, with the best performance achieved using the following parameters:

`{'lstm1': 128, 'lstm2': 64, 'dense1': 128, 'dense2': 64, 'drop1': 0.4, 'drop2': 0.3, 'drop3': 0.3, 'epochs': 30, 'batch_size': 64}`

This configuration yielded the highest F1 score of 0.75, demonstrating the effectiveness of the selected architecture and hyperparameters.

Class	Precision	Recall	F1-score	Support
0	0.75	0.38	0.50	8
1	0.64	0.90	0.75	10
Accuracy		**0.67**		
Macro avg	0.70	0.64	0.62	18
Weighted avg	0.69	0.67	0.64	18

Table 7: Classification Report for the Best Model, LSTM

Following the same methodology, a Bidirectional LSTM model was implemented and evaluated to assess potential performance improvements.

Class	Precision	Recall	F1-score	Support
0	0.80	0.50	0.62	8
1	0.69	0.90	0.78	10
Accuracy		**0.72**		
Macro avg	0.75	0.70	0.70	18
Weighted avg	0.74	0.72	0.71	18

Table 8: Classification Report, BiLSTM

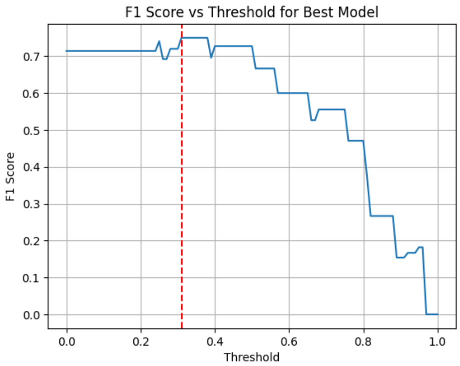

Figure 27: F1 Score vs Threshold for Best Model, LSTM

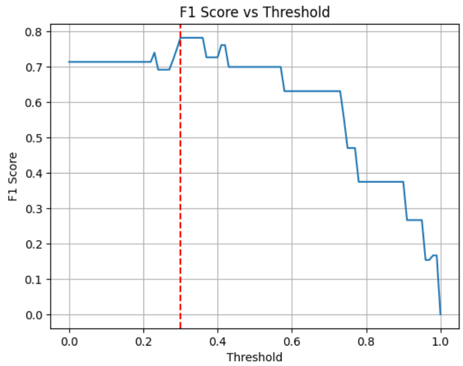

Figure 28: F1 Score vs Threshold for Best Model, BiLSTM

6 Discussion

As observed from the numerical analysis results in Section 4.1, predicting the close price alone proved insufficient (see 4.1.1), reinforcing the necessity of conducting directional analysis. While overall accuracy ranged from a high of 98.29% to a low of 90.46% (Table 4), the directional accuracy hovered around 50%, indicating performance barely better than random guessing.

Subsequently, in the target class prediction task—prior to incorporating sentiment data—the highest accuracy among baseline models was achieved by the Random Forest classifier with 54% accuracy (Table 5). Similarly, LSTM models trained without sentiment data reached a maximum accuracy of 57% (Table 6).

With model improvements and the inclusion of sentiment data, accuracy increased significantly to **72%** (Table 8), outperforming studies mentioned in Section 2.6. Key enhancements identified include the use of class weights, threshold optimization focused on maximizing F1 score, and the integration of sentiment features.

Thus, as discussed in Section 2.2, it was confirmed that incorporating sentiment data positively impacts model performance.

7 Conclusion

In conclusion this study aims to develop a hybrid model for stock market movement prediction. Moreover, this paper serves as a comprehensive literature survey, presents detailed experimental results, and introduces a practical application of the proposed model. Relying solely on close price prediction is insufficient, according to previous research and experimental findings in the field of financial forecasting. As a result, the target class prediction approach was used to create a more sophisticated model. By fusing sentiment research from worldwide news data with numerical analysis based on stock prices and technical indicators, this model takes a hybrid approach. **The suggested model outperforms a number of related works that focused on movement prediction, which reported accuracies of 54.6% and 68.4%, respectively, with an achieved accuracy of 72%.**

In addition to the development of the AI model, a mobile application has been developed where users can share their financial insights, experiences and predictions. The AI model is embedded within this application, enabling users to both interact in a social network environment and monitor the model's predictions in real time. This dual-function platform serves as a space for community driven financial discussion while simultaneously acting as an intelligent decision support system. By combining social interaction with AI-powered analytics, the application offers a unique and accessible tool to enhance financial awareness, encourage data-informed thinking, and foster engagement around market trends.

8 Future Work

Building on the results of this study, a number of possibilities for improving model performance and applicability can be researched:

Finer-Grained Sentiment Classification: Future models may use elaborate classifications like strong positive, neutral, and weak negative in place of the binary sentiment labels (positive/negative). This could increase prediction accuracy and enable more accurate sentiment representation.

Impact of News on Time: News might influence stock movements over long periods of time in along with instantly. To represent the long-term effects of news events over days or even months, future models should include temporal decay or attention mechanisms.

Robust Data Sources:

Integration of several news and financial data sources is necessary to guarantee continuous data availability. When a primary source is unavailable, a second source would enable backup plans.

Scalable Data Retrieval Architecture: To improve scalability and data depth, future implementations can make use of microservices that extract extensive market data via platforms such as YFinance, rather than depending on limited or rate-capped stock APIs.

Granular News Filtering for Single Stocks: To guarantee relevance and cut off noise, single-stock predictions (like AAPL) need more specific news filtering, even while broad keyword filtering can be acceptable for index-level analysis (like the NASDAQ-100).

Use of High-Frequency Data: Considering the extreme volatility of financial markets, adding tick-level (high-frequency) trading data may enhance prediction models and offer more comprehensive insights, particularly for intraday forecasting.

Macroeconomic Indicator Integration: By adding global economic indicators, such as employment data, interest rates, and inflation rates, the model may be better able to adjust to macro-level developments while enhancing forecasting accuracy.

References

[1] Edith Ebele Agu et al. "Proposing strategic models for integrating financial literacy into national public education systems". In: *International Journal of Frontier Research in Science* 3.2 (2024), pp. 010–019.

[2] Abayomi Bello, Sin-Chun Ng, and Man-Fai Leung. "A BERT framework to sentiment analysis of tweets". In: *Sensors* 23.1 (2023), p. 506.

[3] Shri Bharathi and Angelina Geetha. "Sentiment analysis for effective stock market prediction". In: *International Journal of Intelligent Engineering and Systems* 10.3 (2017), pp. 146–154.

[4] Johan Bollen, Huina Mao, and Xiaojun Zeng. "Twitter mood predicts the stock market". In: *Journal of computational science* 2.1 (2011), pp. 1–8.

[5] Emin Huseyin Cetenak, Ayse Cingoz, and Elif Acar. "The effect of national culture on corporate financial decisions". In: *Risk Management, Strategic Thinking and Leadership in the Financial Services Industry: A Proactive Approach to Strategic Thinking* (2017), pp. 355–368.

[6] Weisi Chen et al. "A CEP-driven framework for real-time news impact prediction on financial markets". In: *Service Oriented Computing and Applications* 17.2 (2023), pp. 129–144.

[7] Xiang Deng et al. "Llms to the moon? reddit market sentiment analysis with large language models". In: *Companion Proceedings of the ACM Web Conference 2023*. 2023, pp. 1014–1019.

[8] Yingzhe Dong et al. "Belt: A pipeline for stock price prediction using news". In: *2020 IEEE International Conference on Big Data (Big Data)*. IEEE. 2020, pp. 1137–1146.

[9] Shilpa Gite et al. "Explainable stock prices prediction from financial news articles using sentiment analysis". In: *PeerJ Computer Science* 7 (2021), e340.

[10] Michael D Godfrey, Clive WJ Granger, and Oskar Morgenstern. "The Random-Walk Hypothesis Of Stock Market Behavior a". In: *Kyklos* 17.1 (1964), pp. 1–30.

[11] Sneh Jain, Roopam Gupta, and Asmita A Moghe. "Stock price prediction on daily stock data using deep neural networks". In: *2018 International conference on advanced computation and telecommunication (ICACAT)*. IEEE. 2018, pp. 1–13.

[12] Zhigang Jin, Yang Yang, and Yuhong Liu. “Stock closing price prediction based on sentiment analysis and LSTM”. In: *Neural Computing and Applications* 32 (2020), pp. 9713–9729.

[13] Joshi Kalyani, Prof Bharathi, Prof Jyothi, et al. “Stock trend prediction using news sentiment analysis”. In: *arXiv preprint arXiv:1607.01958* (2016).

[14] Alex Kim, Maximilian Muhn, and Valeri Nikolaev. “Financial statement analysis with large language models”. In: *arXiv preprint arXiv:2407.17866* (2024).

[15] Ivano Lauriola, Alberto Lavelli, and Fabio Aiolli. “An introduction to deep learning in natural language processing: Models, techniques, and tools”. In: *Neurocomputing* 470 (2022), pp. 443–456.

[16] David Kuo Chuen Lee et al. “A Comprehensive Review of Generative AI in Finance”. In: *FinTech* 3.3 (2024), pp. 460–478.

[17] Yinheng Li et al. “Large language models in finance: A survey”. In: *Proceedings of the fourth ACM international conference on AI in finance*. 2023, pp. 374–382.

[18] Wenjie Lu et al. “A CNN-BiLSTM-AM method for stock price prediction”. In: *Neural Computing and Applications* 33.10 (2021), pp. 4741–4753.

[19] Burton G Malkiel. “Efficient market hypothesis”. In: *Finance*. Springer, 1989, pp. 127–134.

[20] Zachary McGurk, Adam Nowak, and Joshua C Hall. “Stock returns and investor sentiment: textual analysis and social media”. In: *Journal of Economics and Finance* 44 (2020), pp. 458–485.

[21] Bohuslava Mihalčová, Adriana Csikosova, and Mária Antošová. “Financial literacy–the urgent need today”. In: *Procedia-Social and Behavioral Sciences* 109 (2014), pp. 317–321.

[22] Vaishali V Nikalje. “A Study of consumption saving and investment patterns of the millennial generation in pune city”. PhD thesis. Tilak Maharashtra Vidyapeeth, 2022.

[23] Yue Qiu, Zhewei Song, and Zhensong Chen. “Short-term stock trends prediction based on sentiment analysis and machine learning”. In: *Soft Computing* 26.5 (2022), pp. 2209–2224.

[24] M. Schuster and K.K. Paliwal. “Bidirectional recurrent neural networks”. In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681. DOI: 10.1109/78.650093.

[25] Akana Chandra Mouli Venkata Srinivas et al. "Sentiment analysis using neural network and LSTM". In: *IOP conference series: materials science and engineering*. Vol. 1074. 1. IOP Publishing. 2021, p. 012007.

[26] Swati Srivastava et al. "Stock price prediction using LSTM and news sentiment analysis". In: *2022 6th international conference on trends in electronics and informatics (ICOEI)*. IEEE. 2022, pp. 1660–1663.

[27] Marzieh Kalantarie Taft et al. "The relation between financial literacy, financial wellbeing and financial concerns". In: *International journal of business and management* 8.11 (2013), p. 63.

[28] Yichuan Xu and Vlado Keselj. "Stock prediction using deep learning and sentiment analysis". In: *2019 IEEE international conference on big data (big data)*. IEEE. 2019, pp. 5573–5580.