

# A Hype Detector based on LSTM, Stock Prices from Market Sentiment and Technical Indicators

Abhinav Dinesh Srivatsa<sup>1</sup>, Ronith Naguri Reddy<sup>2</sup>, Samiksha Racha<sup>3</sup> and Manikandan K<sup>4</sup>

<sup>1-4</sup>School of Computer Science and Engineering, Vellore Institute of Technology, Vellore, India

Email: abhinavdinesh.srivat2021@vitstudent.ac.in, {ronith.naguri2021, samiksha.racha2021}@vitstudent.ac.in, kmanikandan@vit.ac.in

**Abstract**—The chaotic nature of financial markets has historically posed challenges to traditional time series models like Long Short-Term Memory (LSTM), Autoregressive Integrated Moving Average (ARIMA), and Gated Recurrent Units (GRU), resulting in limited success in accurate stock price predictions. To address these limitations, this paper presents a neural network that integrates market sentiment with stock prices and technical indicators to enhance prediction accuracy. We use sentiment data from Indian financial news websites and historical stock data from Yahoo Finance. Our model combines LSTM for stock prices, dense layers for technical indicators, and an embedding layer with LSTM for sentiment analysis, providing a comprehensive approach to market forecasting. The model attains a scaled Root Mean Squared Error (RMSE) of 0.1010, which corresponds to an RMSE of 1.36 INR when unscaled, against an average stock price of 243.73 INR. These results demonstrate a significant improvement over baseline models that rely solely on stock price inputs and sentiment analysis.

**Index Terms**— Financial Markets, Stock Price Prediction, Neural Networks, Sentiment Analysis, Technical Indicators, Long Short-Term Memory (LSTM), Market Forecasting, Indian Financial News and Stock Market Sentiment.

## I. INTRODUCTION

Predicting stock prices has long been a central focus in financial markets, as accurate predictions can offer significant financial rewards. However, stock prices are notoriously difficult to predict due to the wide array of factors that influence market behaviour, including political events, economic changes, and investor sentiment. Traditional time series models like Autoregressive Integrated Moving Average (ARIMA), Long Short-Term Memory (LSTM), and Gated Recurrent Units (GRU), have been widely used for stock price forecasting. While these models perform well in capturing historical trends, they often need to account for external, nonquantitative factors like market sentiment.

New prediction techniques are being investigated as a result of the stock market's chaotic nature, which is influenced by both technical and sentimental variables. Recent developments in sentiment analysis and natural language processing (NLP) have demonstrated that outside influences, such as news stories and conversations on social media, can significantly affect stock values. A comprehensive picture of the market's future movements can be obtained by combining sentiment research with stock prediction algorithms. Despite the advancements in sentiment-based models, there remains a gap in combining market sentiment, technical indicators, and stock prices into a unified framework.

This paper proposes a neural network model that integrates financial news sentiment with historical stock prices and technical indicators. Using data from Indian news websites and stock prices from Yahoo Finance, our model leverages LSTM for stock prices, dense layers for technical indicators & sentiment scores and an embedding layer with LSTM for sentiment analysis to predict stock trends.

The contributions of this research include the integration of multiple data sources into a cohesive predictive model, yielding significant improvements in prediction accuracy. The proposed model attains a scaled Root Mean Squared Error (RMSE) of 1.54 INR, demonstrating its effectiveness in forecasting stock prices compared to baseline models that only use historical price data.

## II. LITERATURE REVIEW

The early work of Bharathi and Geetha [1] used extensive preprocessing with Natural Language Processing (NLP) to analyse RSS feeds. Their predictions showcased a Multi-Layer Perceptron (MLP) to predict a rising or falling trend with a precision of 78.75%.

Maqbool et al. [2] extended this NLP for Sentiment Analysis with an MLP and historical data to use pre-trained models for sentiment scores like VADER, TextBlob, and FLAIR. They were able to predict rising and falling trends with 80% accuracy and 100% precision for specific stocks.

Further advancements were made by Palomo [3], who utilised Twitter Sentiment Analysis to predict market prices. They used FinBERT and some fine-tuned versions of it for Sentiment Analysis to achieve trend prediction with 91.3% accuracy using a Random Forest classifier.

Q. Xiao and B. Ihnaini [4] combined news and Twitter feed sentiment coupled with a “holiday effect” for additional hype analysis.

They used models such as Random Forest, Naive Bayes classifiers, Logistic Regression classifiers, K-Nearest Neighbour classifiers, Support Vector Machine classifiers, and Decision Tree classifiers. The results showed a maximum accuracy of 62.4% on a Naive Bayes model and a maximum precision of 63.5% on a Logistic Regression model.

A predictive approach has incorporated additional metrics, including gold prices, fuel prices, bond yields, and the USD-INR exchange rate, alongside Sentiment Analysis [5].

Neural networks, including the Autoregressive Integrated Moving Average (ARIMA) model and Long Short-Term Memory (LSTM), were used in the study.

Without sentiment analysis, the regression model’s Mean Absolute Percentage Error (MAPE) was 1.6; with sentiment analysis, it increased to 1.25.

Yadav et al. [6] apply unsupervised sentiment analysis to financial news, tuning existing methods for the financial domain. They compare a Noun-Verb combination approach and a hybrid model, with the former yielding the best results. The study highlights the effectiveness of these techniques in capturing sentiment from financial texts and correlating it with stock market trends, showing further improvements and implementation in this paper.

Vijh et al. [7] used Random Forest (RF) and Artificial Neural Networks (ANN) to predict the closing values of stocks from five different industries. The models were trained using financial indicators, including high, low, and closing prices. ANN demonstrated superior accuracy compared to RF, highlighting its predictive effectiveness. The study reported a Mean Absolute Percentage Error (MAPE) of 0.70% for ANN and 0.75% for RF.

Kabir et al. [8] apply machine learning techniques like Linear Regression, Support Vector Machines (SVM), and Random Forest to predict stock prices. By analysing historical stock data and using a correlation matrix, they evaluate the model’s performance. The study concludes that machine learning has significant potential in predicting stock prices accurately.

X. Chen [9] reviews machine learning methods for stock price prediction with LSTM, GRU, and RNN for their superior performance over traditional models like SVM.

LSTM combined with sentiment analysis excels in capturing temporal patterns. While these models show promise, they note challenges in text mining and dimensionality reduction that need addressing for further improvement.

Bansal et al. [10] compare five machine learning algorithms: KNN, Linear Regression, Support Vector Regression (SVR), Decision Tree Regression, and LSTM for stock price prediction using data from 12 leading Indian companies. Their results show that LSTM outperforms all other models in terms of accuracy across performance metrics such as SMAPE, R-squared, and RMSE.

The LSTM model performed with an average SMAPE of 1.59%, the next best metric being 5.59% for SVR. Deep learning methods like LSTM are superior for time series prediction in volatile markets.

### III. EXISTING APPROACHES

#### A. Sentiment Analysis with NLP and Traditional Neural Networks

Bharathi and Geetha [1] applied extensive preprocessing with Natural Language Processing (NLP) on RSS feeds for stock prediction. Their model, based on a Multi-Layer Perceptron (MLP), was able to predict a rising or falling trend with a precision of 78.75%. This approach laid the groundwork for integrating sentiment analysis with stock price prediction.

#### B. Pretrained Sentiment Models with Machine Learning

Maqbool et al. [2] extended the use of NLP for sentiment analysis by utilising pre-trained models such as VADER, TextBlob, and FLAIR. Combining sentiment scores from these models with historical stock data, they achieved an 80% accuracy and 100% precision for predicting stock trends with an MLP model.

#### C. Social Media Sentiment with FinBERT

Palomo [3] took sentiment analysis further by utilising Twitter sentiment through the use of FinBERT, a sentiment analysis model fine-tuned for financial data. Their approach, combined with a Random Forest classifier, reached an accuracy of 91.3%, demonstrating the effectiveness of social media sentiment analysis in predicting stock prices.

#### D. Hybrid Models Combining Multiple Machine Learning Techniques

Q. Xiao and B. Ihnaini [4] integrated news and Twitter feed sentiment with additional features like the “holiday effect” for sentiment-based stock prediction. They used a variety of machine learning models, including Random Forest, Naive Bayes classifiers, Logistic Regression classifiers, K-Nearest Neighbour classifiers, Support Vector Machine classifiers, and Decision Tree classifiers. The maximum accuracy achieved was 62.4% using a Naive Bayes model, while the Logistic Regression model attained a maximum precision of 63.5%.

#### E. Incorporation of External Economic Indicators

Some methods have combined sentiment analysis with external economic factors, including gold prices, fuel prices, bond yields, and the USD-INR exchange rate, to predict stock prices [5]. Models such as Long Short-Term Memory (LSTM) and Autoregressive Integrated Moving Average (ARIMA) were utilised, achieving a Mean Absolute Percentage Error (MAPE) of 1.6 without sentiment analysis and 1.25 when sentiment analysis was included.

### IV. PROPOSED DESIGN

#### A. Data Collection and Storage

- **Web Scraping for News Articles:** To collect relevant financial news, we have developed web scrapers for three major Indian financial news websites: Live Mint, Economic Times, and Money Control. These scrapers extract real-time financial and market news articles, storing them for further analysis. The goal is to gather articles that may impact stock prices, such as those containing market sentiment, company news, or major economic events. All market-related articles are scraped.

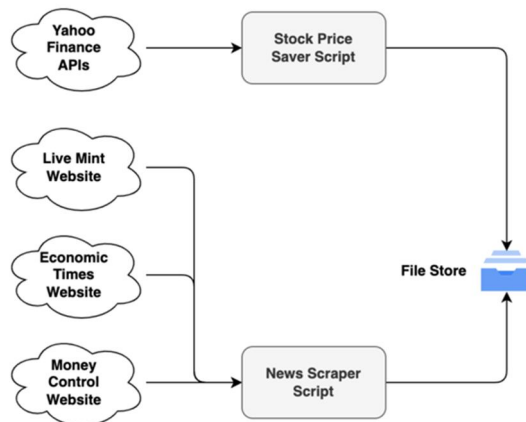


Figure 1. Flow of data collection and storage

- Minute-wise Stock Data Collection: Minute-wise stock price data is collected from Yahoo Finance. This data includes key stock metrics such as open, close, high, and low prices, along with trading volume. Minute-wise data allows for capturing market volatility and price fluctuations on a high-frequency basis, which is essential for precise prediction models to see the real-time impact of market sentiment from news.
- Centralised Data Storage: Both the financial news articles and stock price data are stored in a centralised GitHub repository. The repository ensures easy access, version control, and repeatability for future experiments. By storing both news and stock data in one location, we can maintain consistency during the preprocessing and model training stages.

## B. Data Preprocessing

In this section, we describe the preprocessing steps applied to both the financial news articles and the stock price data to prepare them for model training and analysis.

- Generating Sequences of Stock Data: To capture the temporal aspect of stock price movements, we first apply z-score normalisation to the minute-wise close price data. This normalisation ensures that the data has a mean of 0 and a standard deviation of 1, which helps stabilise model training and prevents certain features from dominating due to scale differences.

$$Z = \frac{X - \mu}{\sigma} \quad (1)$$

Where  $Z$  is the z-score (standardised value),  $X$  is the original value,  $\mu$  is the mean of the dataset and  $\sigma$  is the standard deviation of the dataset

We then generate sequences of close price data over a fixed time window. Each sequence contains normalised close price points for a specified number of minutes, allowing us to capture short-term trends and patterns. These sequences are then used as input for time-series models.

- Sentiment Analysis of News Articles: Sentiment analysis is performed on the news articles to quantify the market sentiment reflected in the text. We use the Valence Aware Dictionary and sEntiment Reasoner (VADER) sentiment analysis tool, which is specifically tuned for analysing sentiment in social media and financial contexts. VADER classifies each article as positive, negative, or neutral, assigning a sentiment score based on the intensity of the sentiment.

These sentiment scores are crucial because embedding layers in machine learning models, while effective at capturing semantic meaning, struggle to directly infer sentiment from text passages. By incorporating sentiment scores, we enhance the model's ability to understand the emotional tone of the news, which can significantly change how a stock price moves. The sentiment scores are then integrated with the stock price data to measure the real-time effect of news sentiment on stock fluctuations.

- Generating Technical Indicator Features: To enhance the predictive power of our model, we compute a range of technical indicators from the stock data. These indicators provide insights into market trends, momentum, and volatility. Below, we detail each indicator used:

- Engulfing Candles: A candlestick pattern known as an engulfing candle occurs when the body of the previous candle is completely engulfed by a larger one. A little red candle followed by a larger green candle indicates upward momentum and a possible turnaround from a downtrend to an uptrend, which is known as a bullish engulfing pattern. A smaller green candle followed by a larger red candle, on the other hand, indicates downward pressure and the possibility of a trend reversal from an uptrend to a downtrend. This is known as a bearish engulfing pattern.

$$\text{Engulfing} = \begin{cases} \text{Bullish,} & \text{if } \text{Close}_i > \text{Open}_i \\ & \wedge \text{Close}_i > \text{Open}_{i-1} \\ & \wedge \text{Open}_i < \text{Close}_{i-1} \\ \text{Bearish,} & \text{if } \text{Close}_i < \text{Open}_i \\ & \wedge \text{Close}_i < \text{Open}_{i-1} \\ & \wedge \text{Open}_i > \text{Close}_{i-1} \\ \text{None,} & \text{otherwise} \end{cases} \quad (2)$$

- Marubozu Candles: When there is no upper or lower shadow on a Marubozu candle, it indicates that the stock opened and finished at its highest or lowest price for the session. Strong buying pressure is indicated by a green Marubozu, whilst strong selling pressure is shown by a red Marubozu.

$$\text{Marubozu} = \begin{cases} \text{Bullish,} & \text{if } \text{Close}_i > \text{Open}_i \\ & \wedge \text{High}_i = \text{Close}_i \\ & \wedge \text{Low}_i = \text{Open}_i \\ \text{Bearish,} & \text{if } \text{Close}_i < \text{Open}_i \\ & \wedge \text{High}_i = \text{Open}_i \\ & \wedge \text{Low}_i = \text{Close}_i \\ \text{None,} & \text{otherwise} \end{cases} \quad (3)$$

- Doji Candles: A Doji Candle occurs when the opening and closing prices are almost identical, reflecting market indecision. It frequently suggests a potential trend reversal, especially when it appears at the high or low points of price movements.

$$\text{Doji} = \begin{cases} \text{True,} & \text{if } |\text{Close}_i - \text{Open}_i| \leq \epsilon \\ & \text{where } \epsilon \text{ is a small threshold value} \\ \text{False,} & \text{otherwise} \end{cases} \quad (4)$$

- Hammer Candles: A Hammer Candle's small body and long lower shadow indicate a possible bullish reversal after a downward trend. The long lower shadow shows that although buyers were able to regain control and close the price close to its opening level, sellers drove the price down considerably. A potential bullish reversal is also indicated by an inverted hammer, which looks like an upside-down hammer. Buyers' attempts to drive the price upward are reflected in its extended upper shadow, while sellers dragged it back down. If buyers pick up steam, this pattern can come before a price hike.

$$\text{Hammer} = \begin{cases} \text{Bullish,} & \text{if } \text{Close}_i > \text{Open}_i \\ & \wedge \text{Low}_i < \text{Open}_i \\ & \wedge (\text{High}_i - \text{Close}_i) \\ & \leq k \cdot (\text{Open}_i - \text{Low}_i) \\ \text{Bearish,} & \text{if } \text{Close}_i < \text{Open}_i \\ & \wedge \text{High}_i > \text{Open}_i \\ & \wedge (\text{Close}_i - \text{Low}_i) \\ & \leq k \cdot (\text{High}_i - \text{Open}_i) \\ \text{None,} & \text{otherwise} \end{cases} \quad (5)$$

- Moving Average Convergence Divergence (MACD): The MACD is derived by subtracting the 26-period Exponential Moving Average (EMA) from the 12-period EMA. The resulting MACD line is then compared to a 9-period EMA of itself, known as the signal line. A crossover of the MACD above the signal line signals bullish momentum, while a crossover below indicates bearish momentum.

$$\text{MACD Line} = \text{EMA}_{12} - \text{EMA}_{26}$$

$$\text{Signal Line} = \text{EMA}_9(\text{MACD Line}) \quad (6)$$

- Relative Strength Index (RSI): One momentum indicator that measures the rate and size of price changes is the RSI. It helps determine whether a market is overbought or oversold and ranges from 0 to 100. The following formula is used to determine the RSI:

$$RS = \frac{\text{Average Gain}}{\text{Average Loss}}$$

$$RSI = 100 - \frac{100}{1 + RS} \quad (7)$$

An RSI above 70 suggests the stock is overbought, while below 30 indicates it is oversold.

### C. Neural Network Model

In this section, we outline the architecture and components of the neural network model utilised for predicting stock price movements based on the preprocessed data. The model is designed to capture complex patterns and relationships in the stock price data, sentiment scores derived from news articles, and keywords from the articles.

- **Architecture:** The proposed neural network consists of multiple components to handle the different input types and features. The input layers are text input, stock input and sentiment and technical indicators input. The text input processes news articles using a text vectorisation layer, followed by an embedding layer that converts the text data into numerical vectors. The stock input receives minute-wise stock price sequences through a separate input, followed by an LSTM layer to capture temporal dependencies. The sentiment and technical indicators input takes in sentiment scores and technical indicators (RSI, MACD, and candlestick patterns), which are processed together through a dense layer. The hidden layers are A combination of LSTM layers (for stock and text inputs) and dense layers for processing the combined sentiment scores and technical indicators. The layers utilise activation functions such as Rectified Linear Unit (ReLU) to introduce non-linearity, enabling the model to capture complex relationships in the data. All inputs (news content, stock prices, sentiment scores, and technical indicators) are concatenated and passed through additional fully connected dense layers to combine the features effectively. The final layer is a dense layer that produces a single continuous value for the regression task of predicting the stock's value a certain amount of minutes in the future.
- **Loss Function and Optimization:** To train the model, we employ a loss function and optimisation algorithm tailored for the regression task. We utilised the Root Mean Squared Error (RMSE) as the loss function, a widely used metric for regression tasks. It helps assess the model's prediction accuracy by giving more weight to larger errors, making them more impactful in the evaluation. The ADaptive Moment estimation (ADAM) optimiser is used for adjusting the weights of the neural network during training.

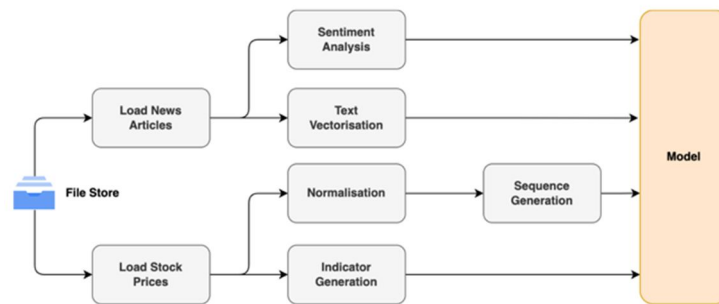


Figure 2. Data preprocessing steps

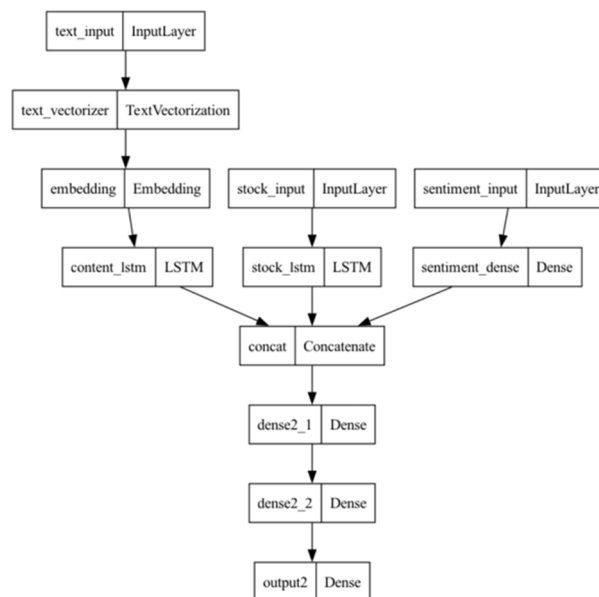


Figure 3. Neural network model architecture

TABLE I. NEURAL NETWORK MODEL PARAMETERS

|    | Type                | Output Shape      | Param #          |
|----|---------------------|-------------------|------------------|
| 1  | InputLayer          | (None, 1)         | 0                |
| 2  | TextVectorization   | (None, 13783)     | 0                |
| 3  | InputLayer          | (None, 10, 1)     | 0                |
| 4  | InputLayer          | (None, 17)        | 0                |
| 5  | Embedding           | (None,13783, 128) | 2,449,280        |
| 6  | LSTM                | (None, 128)       | 66,560           |
| 7  | Dense               | (None, 64)        | 1152             |
| 8  | LSTM                | (None, 64)        | 49,408           |
| 9  | Concatenate         | (None, 256)       | 0                |
| 10 | Dense               | (None, 128)       | 32896            |
| 11 | Dense               | (None, 32)        | 4128             |
| 12 | Dense               | (None, 1)         | 33               |
|    | <b>Total Params</b> |                   | <b>2,603,457</b> |

- **Training Process:** The training of the neural network involves several key steps. The main dataset is split into 80% for training and 20% for testing. During training, 10% of the training data is further used for validation to monitor the model's performance. The model is trained for 5 epochs. The data is processed in batches to improve training efficiency and convergence. The model is trained on three input datasets: stock price sequences, sentiment & technical indicators, and news content. The target labels used for training are continuous stock price forecasts a certain amount of minutes in the future.
- **Evaluation Metrics:** We used the following evaluation metrics. The primary metric used is the square root of the average squared differences between predicted and actual values. RMSE is especially effective in highlighting large errors. After computing the RMSE, we apply inverse scaling to convert the error back into rupee values, allowing us to understand the largeness of the error relative to the original prices.

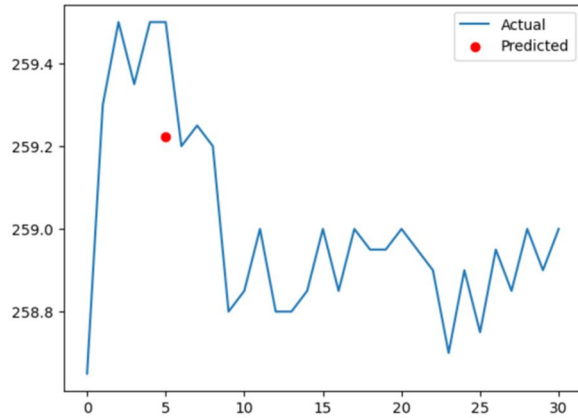


Figure 4. Good prediction – non-neutral sentiment

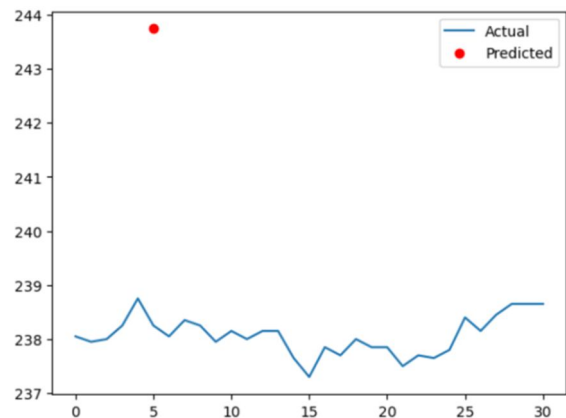


Figure 5. Bad prediction – neutral sentiment

## V. RESULTS ANALYSIS

### A. Summary of Findings

The model was evaluated using Root Mean Squared Error (RMSE), a value of 0.084 for train data and 0.101 for test data.

For the training stock having a mean value of 243.73 INR and standard deviation of 13.49 INR, this comes to an error of 1.13 INR for train data and 1.36 INR for test data. Prediction prices from the model, 2 minutes into the future, have a percentage error of 0.56%. More on these metrics can be seen in Table II.

TABLE II. EVALUATION METRICS FOR DIFFERENT DATASETS

| Dataset    | RMSE (Scaled) | RMSE (Unscaled) (Rupees) | % Error |
|------------|---------------|--------------------------|---------|
| Train      | 0.084         | 1.13                     | 0.463   |
| Validation | 0.081         | 1.09                     | 0.447   |
| Test       | 0.101         | 1.36                     | 0.558   |

### B. Error Analysis

Several cases highlighted the model's strengths and weaknesses. The model struggles predicting the market outcome when the news indicates a neutral sentiment. This can be seen in the visualisations shown in Figures 4 and 5.

### C. Insights on Feature Importance

Analysis of feature importance shows that the news sentiment plays a massive role in the predictions made by the model. This can be seen in Figures 4 and 5. Technical indicators also play a major role.

## VII. CONCLUSION AND FUTURE WORK

We developed a neural network to forecast stock price movements using financial news articles, stock price sequences, sentiment scores, and technical indicators. Although the initial results are promising, there are notable limitations that must be addressed to enhance the model's performance.

One major challenge we encountered was the lack of sufficient data for training the model. Yahoo Finance only provides one week's worth of minute-wise stock data at a time, and we have accumulated only two months' worth of this data. This limited dataset hinders the model's ability to generalise well and capture long-term trends.

Additionally, the news articles used for training have not been separated based on the specific company stocks that the model is targeting. This leads to the usage of data that may dilute the relevance of the information used for predictions, potentially resulting in poor correlations between the news sentiment and the stock price movements of the companies in question.

For future work, we aim to address these limitations by exploring alternative data sources that can provide a larger volume of historical minute-wise stock data to enhance the training dataset, implementing a mechanism to filter and classify news articles based on their relevance to specific stocks, ensuring that the model learns from relevant information only and investigating additional features and data points that could further improve the model's predictive capabilities. By addressing these issues, we hope to enhance the robustness and accuracy of our stock price prediction model in future iterations.

## REFERENCES

- [1] S. Bharathi and A. Geetha, "Sentiment Analysis for Effective Stock Market Prediction," *International Journal of Intelligent Engineering and Systems*, vol. 10, no. 3, pp. 146–154, 2017, doi: 10.22266/ijies2017.0630.16.
- [2] J. Maqbool, P. Aggarwal, R. Kaur, A. Mittal, and I. A. Ganaie, "Stock Prediction by Integrating Sentiment Scores of Financial News and MLP-Regressor: A Machine Learning Approach," *Procedia Computer Science*, vol. 218, pp. 1067–1078, 2023, doi: 10.1016/j.procs.2023.01.086.
- [3] C. Palomo and Stanford University, "Tweet Sentiment Analysis to Predict Stock Market," in *Stanford CS224N Custom Project*, [Thesis], Stanford University, n.d. Available: <https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1234/final-reports/final-report-170049613.pdf>.
- [4] Q. Xiao and B. Ihnaini, "Stock trend prediction using sentiment analysis," *PeerJ Computer Science*, vol. 9, p. e1293, 2023, doi: 10.7717/peerjcs.1293.
- [5] "Stock Price Prediction using Sentiment Analysis and Deep Learning for Indian Markets," [Journal-article], 2021.
- [6] A. Yadav, C. K. Jha, A. Sharan, and V. Vaish, "Sentiment analysis of financial news using unsupervised approach," *Procedia Computer Science*, vol. 167, pp. 589–598, 2020, doi: <https://doi.org/10.1016/j.procs.2020.03.325>.
- [7] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock closing price prediction using machine learning techniques," *Procedia Computer Science*, vol. 167, pp. 599–606, 2020, doi: <https://doi.org/10.1016/j.procs.2020.03.326>.
- [8] M. H. Kabir, A. Sobur, and M. R. Amin, "Stock price prediction using the machine learning," *Journal*, vol. 11, pp. f946–f950, Jul. 2023, doi: 10.1729/Journal.37948.
- [9] X. Chen, "Stock price prediction using machine learning strategies," *BCP Business & Management*, vol. 36, pp. 488–497, Jan. 2023, doi: 10.54691/bcpbm.v36i.3507.
- [10] M. Bansal, A. Goyal, and A. Choudhary, "Stock market prediction with high accuracy using machine learning techniques," *Procedia Computer Science*, vol. 215, pp. 247–265, 2022, doi: 10.1016/j.procs.2022.12.028.