

# LSTM-based Stock Price Prediction Model using News Sentiments

Yijin Yan <sup>1, a</sup>, Xin Nie <sup>1, b</sup>, Mingyang Wang <sup>1, c</sup>, Yuxin Chen <sup>2, d</sup>

<sup>1</sup> Harbin Institute of Technology (Weihai), China;

<sup>2</sup> Central South University of Forestry and Technology, China.

<sup>a</sup> 1091943170@qq.com, <sup>b</sup> JasonNie040123@outlook.com, <sup>c</sup> 2575854738@qq.com,

<sup>d</sup> 2282796442@qq.com

**Abstract.** Forecasting stock prices has been a prominent research topic in finance for many years. With the rapid advancement of internet technologies, financial news has become a crucial source of information for investors, and sentiment analysis of news articles has shown considerable potential in predicting future stock price movements. In this study, we developed a model that combines a Naive Bayes sentiment classifier to convert news text into sentiment indicators, and an LSTM neural network model to predict stock price movements. We obtained data from Oriental Fortune's financial commentary bar, covering the period from 28 July to 23 December 2022, resulting in a total of 318,373 data points. Then, we selected 1400 news items and manually labeled them to train a Naive Bayes classifier to quantify the news sentiment of each trading day. Our feature variables include sentiment indicator, closing price, price movements, P/E ratio, and P/B ratio of SSE 50 constituents. Finally, we applied the Long and Short-Term Memory (LSTM) model to predict stock price movements and compare their performance with the ARIMA time series method. Our results demonstrate that the LSTM model outperforms the traditional ARIMA model in predicting stock price movements. Furthermore, we compared the LSTM model with and without sentiment indicators and found that the inclusion of sentiment indicators significantly improves forecasting its performance. Overall, our proposed model offers a promising approach to predicting stock movements and provides insights into the effectiveness of incorporating sentiment analysis in financial forecasting.

**Keywords:** News sentiment; long short-term memory (LSTM) model; Naive Bayes sentiment classifier.

## 1. Introduction

The stock serves as an essential barometer of the economic market, and accurate stock price prediction has become a topic of great interest in both industry and academia. However, accurately predicting stock price movements is challenging due to the high noise and non-linearity in the stock market. Therefore, machine learning algorithms such as decision trees, genetic algorithms, support vector machines, logistic regression, and deep learning neural network models have been extensively employed to address these challenges. In particular, long short-term memory (LSTM) recurrent neural network models have shown exceptional performance in recent years, as they can effectively handle time series data with long intervals and delays in essential events [1]. Qing Yang [2] et al. conducted a comparative study of the prediction results of support vector regression (SVR), multilayer perceptron (MLP), autoregressive integrated moving average (ARIMA), and LSTM neural networks for 30 different stock indices worldwide. The study found that LSTM neural networks outperformed the other models in terms of prediction accuracy and stability. Similarly, Hui Liu [3] et al. used a convolutional autoencoder (CAE)-LSTM model based on unstructured morphological feature data of stock price images and technical data for short-term forecasting of SSE stocks, which implied excellent prediction results. In a separate study, Peng Zongyu [4] applied LSTM and Transformer data organization methods to predict the stock prices of the Bank of China, illustrating that this data organization method could improve the accuracy of bank stock prediction on validation data. The LSTM model is a promising tool for predicting stock prices, especially in financial forecasting and related fields. The application of big data methods, such as

machine learning algorithms and deep learning neural network models, has led to significant progress in the field of stock prediction. Future research is a necessity to explore new techniques to improve the accuracy and reliability of stock price predictions.

Over time, the predictive capability of structured data for the stock market has gradually declined, leading to increasing attention among domestic and international scholars toward financial texts. Zheng Tracy Ke [5] et al. applied the sentiment-enhanced sequential trend mining (SESTM) method to my text data from the Dow Jones News Network machine text feed and archive database, proving that news sentiment scores can predict the price of new information reaction and subsequently employed for portfolio selection. Faria, Lucas J. [6] et al. studied the relationship between Twitter news sentiment and Brazilian stock market behavior. Nur Azmina Mohamad Zamani [7] et al. analyzed the impact of Malaysian news sentiment on cryptocurrency prices and found that sentiment characteristics significantly contribute to superior price prediction results. In a study on the Chinese stock market, Jia-Yu Hu [8] et al. analyzed four tourism stocks using data from stock forums on Tom eastmoney and demonstrated that adding investor sentiment and news features to the XGBoost model resulted in superior prediction results. Huang Rupeng [9] suggested that the general trend of social sentiment reflected by microblog sentiment information could influence and predict the general trend changes in stock market prices. In addition, Lin Jianhao [10] et al applied a machine-learning trading strategy based on news sentiment for the CSI 300 constituents from 2013 to 2020 and achieved excellent results, indicating that the model's annualized Chinese returns after transaction costs far exceeded the market index returns over the same period. However, few scholars have combined sentiment indicators with other quantitative indicators for stock forecasting in the Chinese stock market. In a more informative portfolio study by Xu Weijun [11], three indicators, including sentiment indicators, stock closing prices, and trading volume, were employed as characteristic variables for stock forecasting.

In this paper, our major contribution is the proposal of an LSTM model, which incorporates a Naive Bayes classifier to represent financial news as sentiment indicators, combining both unstructured financial text data and structured stock data as inputs to predict stock movements. Our study demonstrates that the LSTM model, which integrates unstructured data via a machine learning technique, yields more precise stock price predictions than conventional time series models. The paper is organized as follows: Section 2 outlines the construction of news sentiment indicators using financial news data. Section 3 introduces two forecasting models, namely the ARIMA model and the proposed LSTM model. Section 4 presents the empirical research section, which includes data sources, parameter selection, and results analysis. Finally, the paper concludes by discussing the implications of our findings and offering directions for future research.

## 2. Construction of News Sentiment Indicators Based on Financial News Data

### 2.1 Naive Bayes Sentiment Classifier.

The Naive Bayes classifier is a probabilistic classifier, which means for any financial news article  $d$ , the classifier returns the maximum posterior probability  $\hat{c}$  of  $d$  in all classes  $c_t \in C\{t = 0,1\}$  (where  $c_1$  represents financial news with positive sentiment and  $c_0$  represents financial news with negative sentiment).

$$\hat{c} = \arg \max_{t \in \{0,1\}} p(c_t | d) \quad (1)$$

Bayesian conditional probability:

$$P(x | y) = \frac{P(y | x)P(x)}{P(y)} \quad (2)$$

It combines the above equations (1) (2) yields.

$$\hat{c} = \arg \max_{t \in \{0,1\}} \frac{P(d | c_t) P(c_t)}{P(d)} \quad (3)$$

For the same financial news  $d$ , it is necessary to have the same probability  $P(d)$ . Thus we simplify the above equation.

$$\hat{c} = \arg \max_{t \in \{0,1\}} P(d|c_t)P(c_t) \quad (4)$$

The Naive Bayes algorithm is based on two assumptions: Firstly, each word  $f_m$  in a sentence has the same influence on the classification, regardless of its position in the sentence. Secondly, the features are independent of each other, and a text with  $n$  subwords satisfies this assumption:

$$P(f_1, f_2, \dots, f_n | c_t) = P(f_1 | c_t) \cdot P(f_2 | c_t) \cdot \dots \cdot P(f_n | c_t) \quad (5)$$

The final equation of the Naive Bayes classifier:

$$\hat{c} = \arg \max_{t \in \{0,1\}} P(c_t) \prod_{m=1}^n P(f_m | c_t) \quad (6)$$

To prevent data overflow during training, we obtained the logarithmic formulation of the equation:

$$(6) \hat{c} = \arg \max_{t \in \{0,1\}} \left( \log P(c_t) + \sum_{m=1}^n \log(f_m | c_t) \right) \quad (7)$$

The TF – IDF technique is utilized to feature the text during classifier training. Furthermore, laplace smoothing is also employed to prevent a zero probability in the likelihood term of any class, which can result in a zero probability of that class. Here, TF represents the word frequency, and IDF stands for the inverse document frequency.  $k_m$  is the number of occurrences of the word  $f_m$  in a given text,  $K_m$  is the number of occurrences of the word  $f_m$  in all strand-posted training samples,  $G$  represents the total number of samples posted by the training strand, and  $g_m$  represents the total number of texts containing the word  $f_m$ , yielding

$$TF - IDF = \frac{k_m}{K_m} \cdot \left( \log \frac{G+1}{g_m+1} + 1 \right) \quad (8)$$

The Oriental Wealth Network is a highly influential financial and securities portal in China, attracting a large number of visitors every day with its extensive coverage of financial news across various domains. In this study, we obtained our training sample news by crawling the information stock postings on the Oriental Wealth Network. To ensure the quality of our data, we eliminated two types of text: First, news with obvious positive and negative sentiments which were assigned to multiple companies. Secondly, texts of individual stock postings are posted by non-official information agencies. After this screening process, we were left with 1400 financial news articles, which were then de-weighted. To determine the sentiment of each article, we manually annotated the news texts by hand and conducted two rounds of review to ensure the accuracy of our experimental results. This approach enabled us to obtain a high-quality dataset of financial news articles, which was critical to the success of our sentiment analysis experiments.

In this paper, we utilized the "HIT Discontinued Word List" to eliminate frequently occurring words that do not contribute to the actual meaning of the article, and trained the Naive Bayes classifier in Python. For each piece of news, if the sentiment probability  $\hat{c} > 0.5$ , then the news is classified into a positive class,  $F(d) = 1$ ; on the contrary,  $F(d) = 0$ . The experimental results in this paper were evaluated based on the accuracy rate, calculated using the following formula:

$$Accuracy = \frac{TP + TN}{G} \quad (9)$$

In this paper, we utilized TP to represent the number of news articles which are correctly classified as having positive emotions and TN to represent the number of articles which are correctly classified as having negative emotions.

Table 1. Predicted results

		Predicted classification		Accuracy
		positive	negative	
Actual classification	positive	672	99	89.07%
	negative	54	575	

We divided our dataset into a training set and a test set, with a ratio of 7:3. By adjusting the smoothing parameter Alpha of our model, we were able to achieve an accuracy rate of 90.20% on the test set when Alpha = 11.0. Table 1 shows the experimental results for the entire training set of stock postings.

### 2.2 Construction of Daily News Sentiment Indicators.

Considering the different influences of different news, this paper adopts the sentiment indicator representation method of Xu Weijun [11], who introduced stock posting views as a weighting factor. Since this paper focuses on stock price movements, the time slices are segmented by the trading day degree. The news sentiment generates the sentiment indicator of the (i)-th trading day from 9:30 a.m. of that day to 9:30 a.m. of the (i+1)-th trading day, and the news generated during time intervals such as holidays, or suspensions are deferred to the previous trading day. Below is the calculation formula of sentiment indicators on the (i)-th trading day.

$$S_i = \frac{\sum_{h=1}^T L_i(h) \times F_i(h)}{\sum_{h=1}^T L_i(h)} \quad (10)$$

where  $L_i(h)$  is the number of views of the (h)-th news of the ith trading day and  $F_i(h) \in \{0,1\}$  is explained as the sentiment labeling of the (h)-th news of the (i)-th trading day as judged by the Naive Bayes classifier.

## 3. Stock price movement prediction model

### 3.1 LSTM Model.

LSTM is a recurrent neural network employed to predict problems related to time series. The LSTM neural network incorporates memory units in each hidden layer unit and adjusts the degree of remembering and forgetting of previous and current information by controlling gates, combining short-term and long-term memory to enable the network to have long-term memory capabilities. The working process of the LSTM neural network is as follows [1]

The input information is filtered through the forgetting gate  $f_t$ . Where  $h_{t-1}$  represents the input information of the previous neural unit,  $x_t$  represents the input information of this neural unit, and  $W$  and  $V$  represent the weight matrices, while  $b$  represents the bias vector, and the following equation is presented as:

$$f_t = \sigma(W_f x_t + V_f h_{t-1} + b_f) \quad (11)$$

(2) The state of the input information  $i_t$  and memory information is updated through the input gate, where  $d$  is the activation function, and the input information is:

$$i_t = \sigma(W_c x_t + V_c h_{t-1} + b_c) \quad (12)$$

Create a new vector of candidate values  $c_t$  to add to the new state.

$$\hat{c}_t = \tanh(W_c x_t + V_c h_{t-1} + b_c) \quad (13)$$

(3) Update the cell state to update  $c_{t-1}$  to long-term memory  $c_t$

$$c_t = i_t \hat{c}_t + f_t c_{t-1} \quad (14)$$

(4) Determine the output. The output gate outputs the current information as

$$o_t = \sigma(W_o x_t + V_o h_{t-1} + b_o), h_t = o_t \times \tanh(c_t) \quad (15)$$

**3.2 ARIMA Model.**

The autoregressive integrated moving average (ARIMA) model is a time series model denoted as ARIMA(p, d, q). The parameter p denotes the autoregressive order, which represents the number of past values in the time series which are employed to predict the current value. The parameter q represents the moving average order, which denotes the number of past forecast errors applied to predict the current value. The parameter d is explained as the number of differences needed to make the time series stationary. When d = 0, the data does not require differentiation, and the model simplifies to an autoregressive moving average model denoted as ARMA(p, q). This model is used to describe a stationary time series where the current value is a linear combination of past values and past forecast errors. The formula for the ARMA(p, q) model is as follows:"

$$x_t = \varepsilon_t + \sum_{i=1}^p u_i x_{t-i} + \sum_{n=1}^q \theta_n \varepsilon_{t-n} \tag{16}$$

where  $u_i$  and  $\theta_i$  are the autoregressive and moving average coefficients respectively, and  $s_t$  is a white noise series with zero mean and constant variance. The d-order difference of the time series is given by:

$$\nabla_d y_t = (1 - B)^d y_t = x_t \tag{17}$$

where  $\nabla$  is the difference operator,  $\nabla_d y_t$  is a d-order difference sequence, and conforms to ARMA(p, q).

**4. Empirical Study**

**4.1 Data Sources.**

The sample stocks applied in this study were selected from the constituent stocks of the SSE 50 Investment Index, which is known for its high transaction activity, high liquidity, and representativeness. A total of 318,373 news articles, including news titles, text, posting time, views, and posters, were collected from the financial commentary stock bar under Oriental Fortune for a period of 100 trading days from 28 July to 23 December 2022. Sentiment indicators were obtained after performing Naive Bayes processing on the news data. According to a study by Pan Li [11], average market return, stock market capitalization, and P/E ratio can explain over 90% of the variation in A-share returns. In addition, Weijun Xu [7] utilized sentiment indicators, stock closing price, and volume as characteristic variables to predict future returns of stocks. Based on these previous findings, we select closing price, price movements, P/E ratio, P/N ratio, and sentiment indicator as the characteristic variables to construct the prediction model. The daily data for these variables are shown in the following table 2.

Table 2. Data table for selected trading days

Transaction Date	Closing Price	Up or down (%)	P/E ratio	P/N ratio	Sentiment indicator
2022-07-28	2827.1067	-0.0995	10.3	1.33	0.534812577
2022-07-29	2792.3287	-1.2302	10.2	1.32	0.462640913
2022-08-01	2781.6822	-0.3813	10.16	1.31	0.525675895
2022-08-02	2730.7553	-1.8308	9.99	1.29	0.423999447
		.....			
2022-12-21	2607.4236	0.3053	9.85	1.23	0.580348332
2022-12-22	2619.7569	0.473	9.9	1.24	0.568734161

**4.2 LSTM Model Construction.**

In the first step, we normalized the five variables mentioned above using deprogramming techniques. We then formed a matrix using the processed data from t – 1 trading days and entered

it into the LSTM model. The data from  $t$  trading days were implied for output error correction comparison, thus creating a batch of the dataset. To improve the accuracy of the predictions, we constructed a neural network with a layer of hidden layers ( $layer\_size$ ) of 1 and a hidden node number ( $hidden\_size$ ) of 64. To optimize the model, the parameters of the model are adjusted one by one. Finally, we set the number of training rounds (epochs) to 50, the dropout rate to 0.5, the optimizer to Adam, the batch size to 15, the loss function to MAE, and the activation function to  $relu$ .

$$relu(x) = \max(x, 0) \tag{18}$$

The training set consists of data from the first eighty trading days ranging from July 28 to November 25, while the test set contains data from the last twenty trading days for prediction purposes. The model is trained and the loss function is plotted against the number of training sessions. The left graph shows the prediction results using five factors: closing price, price movements, P/E ratio, P/N ratio, and sentiment indicator. On the other hand, the right graph shows the prediction results without considering the sentiment indicator.

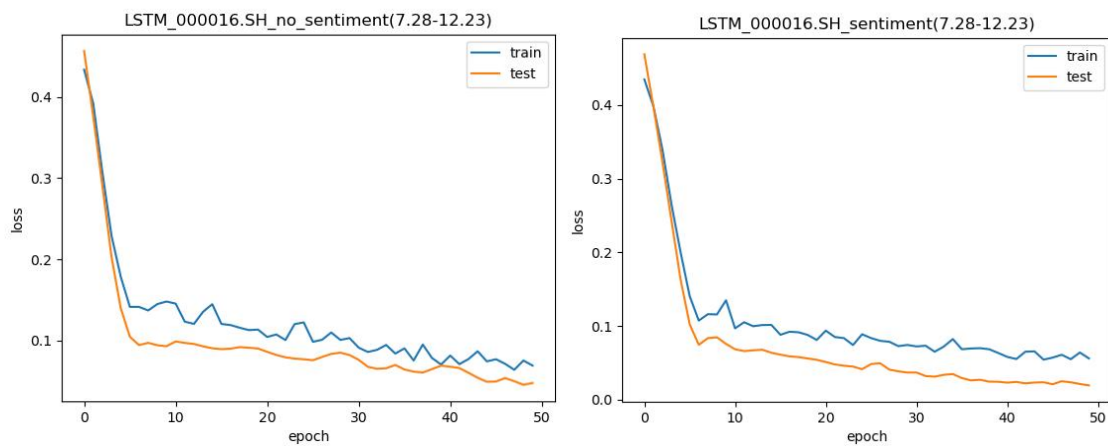


Fig.1 LSTM loss function plot with and without sentiment indicators

After multiple rounds of training, the results demonstrate that the fitting error decreases with the increasing number of training sessions, and the average absolute error between the training set and the test set can be kept below 0.1 at this parameter value.

### 4.3 ARIMA Model Construction.

To assess the smoothness of the data and to determine the appropriate number of differences, we perform a unit root test. The results revealed a significant  $p$  – value of 0, indicating that the data do not require any differencing ( $d = 0$ ). We then created autocorrelation coefficient (ACF) and partial autocorrelation coefficient (PACF) plots to analyze the data for different autoregressive orders ( $p$ ) and mean shift orders( $q$ ), respectively.

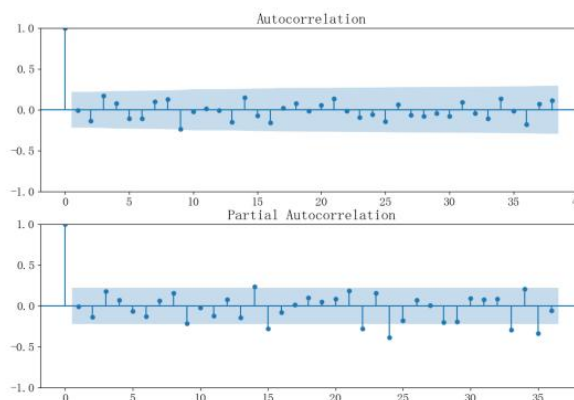


Fig.2 ACF and PACF plots with different differential counts

The correlation between the variables is inspected, as the results indicate a significant relationship when  $p = 9$  and  $q = 15$ . Therefore, an ARIMA(9,0,15) model is created to predict stock market trends.

**4.4 Analysis of Results.**

To ensure an objective evaluation of the model's forecasting ability and accuracy, we employed several statistical measures, including MSE, RMSE MAE, and R\_square to assess the accuracy of the model's predictions. The higher the values of MSE, RMSE, and MAE, the more accurate the model's predictions are. Additionally, the R\_square is used to evaluate the overall performance of the forecasting model. A higher R\_square value indicates better forecasting results and the formulas calculating these statistical measures are as follows.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|} \tag{19}$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{20}$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \tag{21}$$

$$R\_square = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \tag{22}$$

where  $y_i$  is the actual value,  $\hat{y}_i$  is the predicted value,  $n$  is the number of samples, and  $\bar{y}$  is the mean. The data from the first sixty trading days are selected as the training set, and the data from the last forty trading days are designated as the test set for comparison. The experimental results are presented in Table 3.

Table 3. Comparison of the prediction accuracy of different models when the training set: test set = 3:2

	ARIMA(9,0,15)	LSTM_sentiment	LSTM_no_sentiment
MSE	1.785267	0.788390	1.603944
RMSE	1.336139	0.887913	1.266469
MAE	0.999368	0.571139	0.893070
R_square	0.219258	0.642415	0.272510

We evaluated the predictive power of the LSTM and ARIMA models for stock prices, with and without the inclusion of a sentiment indicator. The results demonstrate that the goodness-of-fit of both models was poor in the absence of the sentiment factor, with the LSTM model yielding a goodness-of-fit (R\_square) of 0.272510 and the ARIMA model yielding a goodness-of-fit of 0.219258. However, after incorporating sentiment as a factor, the goodness-of-fit improved significantly to 0.642415, suggesting a strong predictive performance with the low noise treatment. To further validate these findings, the training set and test set are split, and the number of training samples is increased. Specifically, the initial eighty trading days were designated as the training set, whereas the final twenty trading days were assigned as the test set. The experimental results, as shown in Table 4, confirm the accuracy of the models when sentiment is included as a factor. Furthermore, Figure 3 shows the predicted values of each model plotted against the actual values.

Table 4. Comparison of the prediction accuracy of different models when the training set: test set = 4:1

	ARIMA(9,0,15)	LSTM_sentiment	LSTM_no_sentiment
MSE	1.226530	0.037130	0.291549
RMSE	1.107488	0.192691	0.539953
MAE	0.831606	0.160713	0.400793
R_square	0.292972	0.978597	0.831938

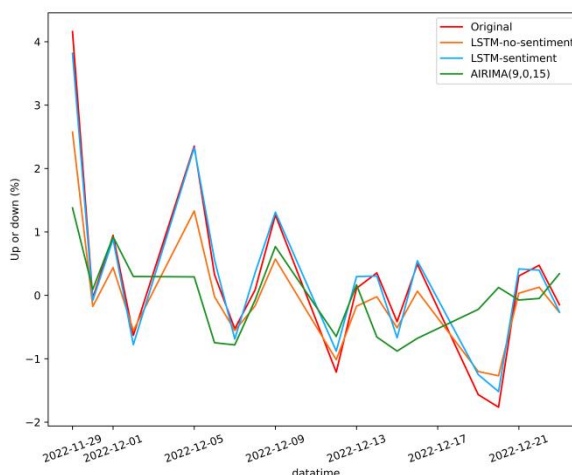


Fig.3 Comparison of prediction results of different models

By augmenting the training dataset, we have observed significant improvements in the predictive capabilities of both the long and short-term memory (LSTM) model and the ARIMA model, with the MSE, RMSE, and MAE of the ARIMA model decreasing by 31.30%, 17.11%, and 16.79%, respectively. The MSE, RMSE, and MAE of the LSTM model with the addition of the sentiment factor decrease by 95.29%, 78.30%, and 71.86%, respectively. The MSE, RMSE, and MAE of the LSTM model without the addition of the sentiment factor decrease by 81.83%, 57.37%, and 55.12%, respectively. In comparison, the study reveals that the noise levels of the long and short temporal memory (LSTM) model are substantially decreased and the goodness-of-fit measures exceed 0.8., which indicates that the LSTM is effective in processing text data and can achieve more accurate prediction results. The model proposed in this study achieved the most significant training effect, with a substantial improvement in model performance that is both robust and achieves excellent results, as evidenced by a goodness-of-fit superiority of approximately 0.9786 in the experiment.

## 5. Summary

In this paper, we study the method of stock forecasting by extracting the news sentiment from financial news, crawling the financial information from the financial commentary stock bar of the Oriental wealth network, constructing news sentiment indicators using the Naive Bayes method, and employing the ARIMA time series model and LSTM models to predict the next business day's stock price movement, respectively. The actual performance of the LSTM model based on news sentiment in the Chinese stock market is examined. The analysis of the experimental results leads to the following conclusions: Initially, a simplistic approach was utilized by us to represent news sentiment using a Naive Bayes classifier, based on a limited number of manually annotated samples, which resulted in superior classification accuracy. Additionally, reading volume was applied as a weighting factor to derive sentiment indicators. Secondly, We combine the sentiment indicator with the structured stock index closing price, price movements, P/E ratio, and P/N ratio, and obtain better prediction results than the LSTM model using the structured stock index alone, confirming that financial news has an important influence on stock movements in the Chinese market. Finally,



the news sentiment-based LSTM model proposed in this paper achieves much better prediction results than the traditional ARIMA time series model, and the accuracy of the prediction results improves significantly after adding training samples, which verifies the effectiveness and superiority of the model proposed in this paper.

While our study has yielded some valuable results, there is still much room for improvement and exploration. In future research, we aim to improve the performance of the model in several aspects. One limitation of our study is the limited sample size, as we only selected constituents of the SSE 50 Index. This sample size is relatively small compared to the total size of more than 4,000 A-share stocks. Consequently, the impact of financial news on stocks with small market capitalization and low attention may not be significant, which could result in lower overall prediction accuracy than expected.

Overall, we believe our future studies will lead to more accurate predictions and provide valuable insights for financial decision-makers.

## References

- [1] Peng Yan, Liu Yuhong, Zhang Rongfen. Modeling and Analysis of Stock Price Prediction Based on LSTM. *Computer Engineering and Application*, 2019,55 (11): 209-212
- [2] Yang Qing, Wang Chenwei. Research on global stock index prediction based on deep learning LSTM neural network. *Statistical Research*, 2019, 36 (03): 65-77
- [3] Liu H, Qi L, Sun M. Short-Term Stock Price Prediction Based on CAE-LSTM Method. *Wireless Communications and Mobile Computing*, 2022.
- [4] Peng, Zong-Yu, and Pei-Chang Guo. A data organization method for LSTM and transformer when predicting Chinese banking stock prices. *Discrete Dynamics in Nature and Society* 2022 (2022): 1-8.
- [5] Ke Z T, Kelly B T, Xiu D. Predicting returns with text data. *National Bureau of Economic Research*, 2019.
- [6] Faria L J, Silva K C, Fernandes D. Tweet and news sentiment indicators and the behavior of the Brazilian stock market. *IEEE Transactions on Power Systems*, 2022: 578-583.
- [7] Zamani N A M, Yan J L S, Yusof A M. Cryptocurrency Price Prediction using Bi-GRU Model with English and Malay News Sentiment Features. *IEEE Transactions on Power Systems*, 2022: 136-141.
- [8] Hu Y, Shao L, La L. Using Investor and News Sentiment in Tourism Stock Price Prediction based on XGBoost Model. *IEEE Transactions on Power Systems*, 2021: 20-24.
- [9] Huang Runpeng, Zuo Wenming, Bi Lingyan. Stock Market Prediction Based on Weibo Emotional Information. *Journal of Management Engineering*, 2015,29 (01): 47-52.
- [10] Lin Jianhao, Zhang Yifan, Chen Liangyuan, Deng Yimeng. Machine learning trading strategy based on news sentiment. *Journal of Econometrics*, 2022,2 (04): 881-908.
- [11] Xu Weijun, Huang Jinglong, Fu Zhineng, Zhang Weiguo. Research on the Black-Litterman portfolio model based on emotion mining of financial text - taking the post text of Oriental Fortune Stock Bar and China's A-share market as an example. *Journal of Operations Research*, 2022,26 (04): 1-14.
- [12] Pan Li, Xu Jianguo. Risk and characteristic factors of the A-share market. *Financial Research*, 2011 (10): 140-154